

UFBA

UNIVERSIDADE FEDERAL DA BAHIA
ESCOLA POLITÉCNICA
PROGRAMA DE PÓS GRADUAÇÃO EM
ENGENHARIA INDUSTRIAL - PEI

DOUTORADO EM ENGENHARIA INDUSTRIAL

EDUARDO CARDOSO MORAES

DESENVOLVIMENTO DE INTERFACES BASEADAS EM SERVIÇO PARA INTEGRAÇÃO DE SISTEMAS HETEROGÊNEOS NA MANUFATURA





SALVADOR 2017





UNIVERSIDADE FEDERAL DA BAHIA ESCOLA POLITÉCNICA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA INDUSTRIAL

EDUARDO CARDOSO MORAES

Desenvolvimento de interfaces baseadas em serviço para integração de sistemas heterogêneos na manufatura

Salvador

EDUARDO CARDOSO MORAES

Desenvolvimento de interfaces baseadas em serviço para integração de sistemas heterogêneos na manufatura

Trabalho apresentado ao Programa de Pósgraduação em Engenharia Industrial-PEI do da Universidade Federal da Bahia como requisito parcial para obtenção do grau de Doutor em Engenharia Industrial.

Orientadores: Prof. Dr. Herman Augusto Lepikson

Prof. Dr. Armando Walter Colombo

Salvador

2017

Modelo de ficha catalográfica fornecido pelo Sistema Universitário de Bibliotecas da UFBA para ser confeccionada pelo autor

Desenvolvimento de padrões de interfaces para integração de

MORAES, EDUARDO CARDOSO

```
sistemas heterogêneos na manufatura / EDUARDO CARDOSO MORAES. -
- Salvador, 2017.
247 f.: il

Orientador: Herman Augusto Lepikson.
Coorientador: Armando Walter Colombo.
Tese (Doutorado - Engenharia Industrial) -- Universidade
Federal da Bahia, PEI, 2017.

1. Manufatura Avançada. 2. Indústria 4.0. 3. SOA. 4.
Metodologia orientada a serviços. I. Lepikson, Herman Augusto.
II. Colombo, Armando Walter. III. Título.
```

DESENVOLVIMENTO DE INTERFACES BASEADAS EM SERVIÇO PARA INTEGRAÇÃO DE SISTEMAS HETEROGÊNEOS NA MANUFATURA

EDUARDO CARDOSO MORAES

Tese submetida ao corpo docente do programa de pós-graduação em Engenharia Industrial da Universidade Federal da Bahia como parte dos requisitos necessários para a obtenção do grau de doutor em Engenharia Industrial.

Examinada por:
Prof. Dr. Ângelo Amâncio Duarte
Doutor em Ciências da Computação, pela Universidad Autonoma de Barcelona, Espanha, 2008
Prof. Dr. Manoel Gomes de Mendonça Neto
Doutor em Ciências da Computação pela Universidade de Maryland Em College Park,
Estados Unidos da América, 1997
Prof. Dr. Sergio Soares Que De Constantina de Const
Doutor em Ciências da Computação, pela Universidade Federal de Pernambuco, Brasil 2005
Luizan Chat
1 1011 211 201201 001111111111
Doutor em Engenharia Eletrônica e Computação pelo Instituto Tecnológico de
Aeronáutica, Brasil, 2001
Prof. Dr. Márcio André Fernandes Martins Plaicis Andri Finnandes Plantin
Doutor em Engenharia Química pela Universidade de São Paulo, Brasil, 2014

Agradecimentos

Agradeço a Deus a oportunidade concluir esta etapa de vida que para mim é um sonho. A trajetória de um doutorado parece ser um caminho de muito esforço e solitário de um aluno em uma pesquisa de alto nível, porém isso não descreve o processo nem os auxílios necessários para que isso se torne realidade. O processo foi árduo, mas importante para me tornar um pesquisador melhor e uma pessoa melhor, com uma visão mais ampla da vida. E não chegaria aqui sem o apoio de muita gente.

Como Isaac Newton escreveu em sua famosa frase: "Se eu vi um pouco mais longe é porque me apoiei nos ombros de gigantes." Eu agradeço de coração a meus orientadores Prof. Herman Lepikson e o Prof. Walter Colombo por todo apoio e orientação durante o doutorado.

Agradeço ao Prof. Herman por todo o tempo me guiar durante esse processo, com uma visão holística e segura, pôde com sua experiência me indicar inicialmente uma oportunidade de pesquisa, depois por me auxiliar em minha visão limitada sobre a temática. É um líder natural onde aprendi e aprendo continuamente com seu vasto conhecimento, profissionalismo, gentileza, e humildade.

Sinto-me orgulhoso e grato pelo Prof. Colombo me receber na Alemanha e de fazer parte grupo de pesquisa dele. Prof. Colombo me deu acesso a conhecimento que é realmente o estado da arte, dando-me uma outra visão sobre a pesquisa de alto nível e integração entre os projetos da academia e indústria. Obrigado por excelentes direcionamentos e *feedback*s que fizeram a diferença na minha tese. Eu nunca vou esquecer suas palavras: NÃO mantenha o conhecimento só para você, compartilhe por um mundo melhor e sempre ajude os outros.

Agradeço a minha família, em especial Priscilla pela paciência e apoio e ao meu filho Guilherme, que mesmo sem saber, é uma fonte inesgotável de motivação para mim.

Agradeço o apoio do laboratório de pesquisa CTAI da UFBA, aos meus amigos de doutorado Ana Paula Tanajura e Eduardo Teles, pela ajuda durante as disciplinas e pelas longas conversas sobre a definição do foco da tese, essa troca de experiências foi muito rica.

Eu agradeço meus amigos internacionais que são pessoas que levarei comigo para o resto da vida em especial Jeffrey Wermman que se tornou um amigo, e é um profissional incrível que me deu todo o suporte necessário durante os casos de uso desta tese e na escrita de artigos científicos comigo.

Gostaria de citar e agradecer a Bruno Alvarez Gil, Sergey Konstatinov, Aitor Casado, Frederik Gosewehr, Matthias Wermman, Ahmad Abdalazim, Bilal Ahmad, Frank Stefan, Anika, Anke, Hauke Beninga, Thorsten Stromann, Nils Kliesing pelo apoio na pesquisa ou por conversas ou happy hours, minha vida pessoal foi muito melhor por causa de vocês.

Dankeschön.

Acknowledgments

I thank God for the opportunity to complete this stage of my life which for me is a dream. The trajectory of a PhD seems to be a path of many efforts of a lonely student in a high-level research, but it does not describe the process or aid necessary for it become true. The process was arduous but important for me to become a better researcher and a better person, in a broader perspective of life. I could not achieve this without the support of many people.

As Sir. Isaac Newton wrote in his famous phrase: "If I have seen a little further it is by standing on the shoulders of Giants". I thank from my heart to my advisors Prof. Herman Lepikson and Prof. Walter Colombo for all the support and guidance during the PhD.

I thank Prof. Herman for all the time spent to guide me through this process with a holistic and safe vision, could share with his experience initially show me a research opportunity, then by helping me in my limited view on the subject. It is a natural leader where I continuously learn with his vast knowledge, professionalism, kindness, and humble.

I am proud and thankful for Prof. Colombo for receiving me in Germany and make me part of his research group. Prof Colombo has a huge knowledge and illuminated vision, he gave me access to the state-of-the-art knowledge, giving me another view of high-level research and integration between the academy and industry projects. Thanks for excellent inputs and feedbacks that were critical in my work. I will never forget your words: Don't keep the knowledge only for you, share for a better world and help others.

I thank my family, especially Priscilla for their patience and support and my son Guilherme, even unknowingly, is an inexhaustible source of motivation for me.

I am grateful for the support of CTAI (Industrial Automation Lab) in

UFBA and especially my doctoral friends Ana Paula Tanajura and Eduardo Telles, for helping me during the disciplines and the long talks about the definition of the thesis, this exchange of experience was very rich.

I thank my international colleagues during my exchange in Germany, an especial experience that I will take with me for the rest of my life. I want to thank especially Jeffrey Wermman who became a friend, and is an incredible professional who gave me all the necessary support for developing use cases and wrote scientific papers with me.

I would like to thank Bruno Alvarez Gil, Sergey Konstatinov, Aitor Casado, Frederik Gosewehr, Matthias Wermman, Ahmad Abdalazim, Bilal Ahmad, Frank Stefan, Anika, Anke, Hauke Beninga, Thorsten Stromann, Nils Kliesing for the research support or for enjoyable conversations and sharing with me traditional German aspects. My personal life was much better because of you.

Dankeschön.



MORAES, Eduardo Cardoso. DESENVOLVIMENTO DE INTERFACES BASEADAS EM SERVIÇO PARA INTEGRAÇÃO DE SISTEMAS HETEROGÊNEOS NA MANUFATURA. 2017 (Tese de doutorado). Programa de Pós-graduação em Engenharia Industrial. Universidade Federal da Bahia, Salvador, 2017

RESUMO

Graças ao constante progresso na ciência, o mundo está em constante transformação, onde novas tecnologias são desenvolvidas diariamente. Isto tem um grande impacto sobre diferentes áreas, e uma das mais afetadas, e com importantes efeitos na competitividade, é a da indústria. O chão de fábrica em geral possui estruturas rígidas e hierárquicas que não conseguem atender a constante necessidade de inovação requerida pelas empresas. Empresas industriais modernas requerem processos e linhas de produção cada vez mais flexíveis para permanecerem competitivas. Este progresso tem ocorrido em tal magnitude que atualmente há quem proponha que estamos em via de uma quarta revolução industrial. Esta nova revolução está baseada na digitalização e no barateamento de hardware, que habilitam a introdução de modernos sistemas de computação e tecnologias de comunicação e controle, especialmente no chão de fábrica, de forma a aumentar a interoperabilidade entre os diferentes sistemas (hardware e software) existentes em uma fábrica. Esta tese propõe uma arquitetura modular e uma metodologia baseada em serviços via interfaces para subsidiar o conceito "Conecte e Produza" (Plug and Produce) com ênfase na interoperabilidade. Estudos de caso com aplicações de diferentes níveis de complexidade foram elaborados para validar esta tese. Verificou-se que as soluções baseadas em serviços reduzem a complexidade de integração dos elementos envolvidos na automação, com ganhos em flexibilidade e menor tempo de configuração para mudanças de processo/produto. Porém limitações foram detectadas e detalhadas ao longo da tese.

Palavras chaves: Manufatura avançada, sistemas ciberfísicos, orientação a serviços, sistemas distribuídos

MORAES, Eduardo Cardoso. DESENVOLVIMENTO DE**INTERFACES** EMSERVIÇO INTEGRAÇÃO DE BASEADAS PARA SISTEMAS HETEROGÊNEOS NA MANUFATURA. 2017 (Tese de doutorado). Programa de Pós-graduação em Engenharia Industrial. Universidade Federal da Bahia, Salvador, 2017

Abstract

Thanks to the steady progress in science, we are living in a quickly emerging world, where new trends and technologies are developed daily. This has an impact on many different areas, one of which is the whole field of industrial automation. Especially in the last few years, a lot of progress has been made in this area, leading to some people speaking about the 4th industrial revolution. When talking about this new revolution, also referred to as Industry 4.0, one usually is speaking about introducing modern information and communication technology into the whole factory to maximize the interoperability between all the different existing systems in a factory. This interoperability is the backbone to make a factory more flexible and intelligent, as the different sub-systems are now able to communicate and interact between each other, facilitating the interactions between business strategy and operations. These changes are important steps to reach most of the current needs of industrial facilities, such as the rising demand for highly customized products, improving the resource efficiency and higher throughput. This thesis proposes a modular architecture and a methodology based on services via interfaces to subsidize the concept Plug and Produce with emphasis on interoperability. Case studies with different levels of complexity were developed to validate this thesis. It was found that solutionbased services reduce the complexity of integration of the elements involved in automation, with gains in flexibility and less time to adapt the configuration of the process/product. However, limitations were detected and described through the thesis.

Key words: advanced manufacturing, cyber-physical systems, SOA, distributed manufacturing

Sumário

CAPÍTULO 1 – INTRODUÇÃO E CONTEXTUALIZAÇÃO	21
1.1 - Delimitações da pesquisa e Problematização 24	
1.2 - Hipótese que orienta a Pesquisa25	1
1.3 - Objetivos)
1.4 - Metodologia27	,
1.5 - Estrutura do Trabalho29	١
CAPÍTULO 2 - REVISÃO DO ESTADO DA ARTE	31
2.1 – Evolução da automação industrial	
2.2 – Manufatura avançada 34	
2.3 – SOA	l
2.4 – Modelos Empresariais 49	١
2.5 – Sistemas Ciberfísicos	!
2.6 - Computação nas Nuvens 54	·
2.7 - Internet das Coisas - IoT	i
CAPÍTULO 3 – MODELO PROPOSTO	61
3.1 – MODELO AMOS	
3.2 – Webservices	١
3.3 - REST	
3.4 - OPC UA 82	!
3.5 - DPWS85	1
3.6 - GSOAP 87	
Capítulo 4 – ARQUITETURA MODULAR - AMOS	89
4.1 – Conceitos preliminares	١
4.2 - Arquitetura modular baseada em componentes e serviços 91	

4.3 – Desenvolvimento de soluções orientadas a serviço 108
4.4 – Considerações
CAPÍTULO 5 - ESTUDOS DE CASO
5.1 - Estudo de caso 1
5.2 - Estudo de caso 2
5.3 - Estudo de Caso 3
5.4 - Estudo de Caso 4
CAPÍTULO 6 - CONCLUSÃO
6.1 - Contribuições científicas da Tese
6.2 - Limitações
6.3 - Disseminação do Conhecimento
6.4 – Trabalhos Futuros
REFERÊNCIAS
APÊNDICE A191
Modelo de planejamento de Migração191
Apêndice B – Estudo de Caso Leitor RFID196
Apêndice C – Estudo de Caso Cloud Manufacturing210
Glossário

Lista de Figuras

Figura 1 - Visão geral das áreas e contribuições da Tese20	б
Figura 2 - Algumas ferramentas e aspectos considerados28	8
Figura 3 - Linha do tempo das evoluções industriais33	3
Figura 4 – Nova de estrutura empresarial baseada em modelo de dados42	2
Figura 5 - Transformação da arquitetura ISA-95 em serviços40	б
Figura 6 - Níveis do modelo ISA-955	1
Figura 7 - Componentes de um serviço de computação nas nuvens5	5
Figura 8 - Níveis de evolução de sistemas inteligentes58	8
Figura 9 - Etapas de um sistema voltado para a IoT54	4
Figura 10 - Concepção de Serviço e Interface63	3
Figura 11 – Modelo de comunicação entre dispositivos e sistemas69	9
Figura 12 – Modelo de Integração AMOS7	1
Figura 13 – Camadas de Controle73	3
Figura 14 – Fluxo de comunicação via SOAP80	0
Figura 15 - Estrutura do protocolo DPWS80	б
Figura 16 – Estágios de maturidade no uso de serviços93	3
Figura 17 – Visão da arquitetura proposta94	4
Figura 18 - Estágio 1.0 - Elementos conectados via <i>Gateway</i> 96	б
Figura 19 – Fluxo de comunicação com o uso de gateway98	8
Figura 20 - Estágio 2.0	0

Figura 21 - Estágio 3.0104
Figura 22 - Exemplo de tela de controle com indicadores em tempo real 106
Figura 23 – Visão geral da metodologia proposta na tese109
Figura 24 – Estudos de caso realizados116
Figura 25 – Funcionalidades selecionadas para transformar em serviços . 118
Figura 26 – Estrutura de um pacote SOAP122
Figura 27 – Lista de bibliotecas necessárias Erro! Indicador não definido.
Figura 28 – Equipamentos do Kit RFID utilizado124
Figura 29 – Raspberry Modelo B+ com 512 Mb125
Figura 30 – Imagem do robô Robotino128
Figura 31 – Comunicação entre o Servidor Raspberry, o robô Robotino 131
Figura 32 – Interface Web desenvolvida para controle do Robotino134
Figura 33 – Estande de Feira CEBIT 2015135
Figura 34 –Fábrica digital do Laboratório138
Figura 35 – Esteira e palete para transporte de peças LEGO139
Figura 36– Planta da fábrica digital da Universidade de Emden/Leer 140
Figura 37– Robô Mitsubishi que atua na estação de trabalho 1 Erro! Indicador não definido.
Figura 38 –Inico S100 – dispositivo para interface XML/SOAP143
Figura 39- Exemplo da criação de um evento147
Figura 40– Principais ações controladas pelo sistema orquestrador150
Figura 41– Planta da fábrica151
Figura 42– Componentes da ferramenta CCE155

Figura 43 - Diagrama geral do estudo de caso	. 156
Figura 44 – Parte do sistema utilizado para a virtualização	. 159
Figura 45 – Objetos virtualizados para o estudo de caso	. 162
Figura 46 – Exemplo de objeto com suas características	. 163
Figura 47- Descrição da tela para definição de comportamentos	. 164
Figura 48 – Interface de mapeamento entre Funções e CLPs via CCE	. 165
Figura 49 - CLP Siemens S7-300 utilizado	. 166
Figura 50 – Aplicativo UA Modeler	. 167
Figura 51 – Tela do Aplicativo Cliente OPC desenvolvimento próprio	. 168
Figura 52 – Tela do Aplicativo OPC Client	. 169
Figura 53 – Comparação tempo de resposta HTTP e binário OPC.TCP	.171
Figura 54 – Comparação tamanho de pacote HTTP e OPC.TCP	.171
Figura 55 – Fluxo de comunicação entre as funções definidas	. 196
Figura 55 - Diagrama de Classe do estudo de caso	.210
Figura 56 – Programa desenvolvido no CLP com blocos de Função	.211
Figura 57 – Transição de estados do elemento e sua descrição	.211

Lista de Tabelas

Tabela 01-Tabela comparativa sobre soluções em OPC UA	172
Tabela 02 – Métodos e variáveis da interface REST	199

Lista de Abreviaturas e Siglas

ABB Asea Brown Boveri

API Application Programming Interface (Interface de programação da aplicação)

CAD Computer Aided Design (Design auxiliado por computador)

CCE Component Core Editor (Núcleo Editor de componente)

CEP Controle Estatístico de Processo

CLP Controlador Lógico Programável

CP Controlador Programável

CNC Computer Numerical Control (Controle numérico computadorizado)

CNI Confederação Nacional da Indústria

COM Component Object Mode (Modelo objeto de componente)

CoAP Constrained Application Protocol (Protocolo de aplicação restrito)

CPU Central Process Unit (Unidade Central de Processamento)

CRM Customer Relationship Management (Gerenciamento de relacionamento com consumidor)

DA Data Access (Acesso a dados)

DB Data Block (Bloco de dados)

DCS Distributed Control Systems (Sistemas de controle distribuídos)

DCOM *Distributed Component Object Model* (Modelo distribuídos de components e objetos)

DLL Dynamic-link library (Bilioteca de ligação dinâmica)

DPWS Devices Profile for Web Services (Dispositivos Perfil de Web Serviços)

DX Data eXchange (Troca de dados)

EAM Enterprise Assets Management (Gerenciamento de recursos empresariais)

ERP Enterprise Resource Planning (Planejamento de recursos empresariais)

FB Function Block (Blocos de função)

HDA Historical Data Access (Acesso a dados históricos)

IHM Interface Homem-máquina

HTTP Hyper Text Transfer Protocol (Protocolo de trasnferencia de Hipertexto)

IEC *The International Electrotechnical Commission* (Comissão internacional de eletrotécnica)

IETF Internet Engineering Task Force (Força de tarefa de engenharia de internet)

IPEA Instituto de Pesquisa Econômica Aplicada

ISA The International Society of Automation (Sociedade Internacional de automação)

JSON JavaScript Object Notation (Notação de Objetos JavaScript

M2B *Machine to Business* (Maquinas para negócios)

M2M Machine to Machine (Maquinas para máquinas)

TI Tecnologia da informação

TIC Tecnologia de Informação e Comunicação

TICC Tecnologia de Informação, Comunicação e Controle

MES Manufacturing Execution System (Sistemas de execução da manufatura)

OPC Acrônimo adotado para *Object Linking and Embedding for Process Control*) (encadeamento de objetos para controle de Processos)

OPC UA - OPC Unified Architecture (Arquitetura Unificada OPC)

PMI Project Management Institute Instituto de Planejamento de Projetos

PMS *Production Management Systems* (Sistemas de gerenciamento de Produção)

REST Representational State Transfer (Transferêcia de estados representacional)

RTUs Remote Terminal Units (Unidade de terminal Remoto)

SAD Sistemas de Aquisição de Dados

SCADA Supervisory Control and Data Acquisition (Sistemas supervisório de controle e aquisição de dados)

SCF Sistemas Ciberfísicos

SDCD Sistema Digital de Controle Distribuído

SDK Software Development Kits (Kit de desenvolvimento de Software)

SOA Service Oriented Architecture (Arquitetura Orientada a serviço)

SOAP Simple Object Access Protocol (Protocolo de acessos simploficado a objetos)

SQL Structured Query Language (Linguagem estruturada de consulta)

SRM Supplier Relationship Management (Sistema de relacionamento com fornecedor)

TCP Transmission Control Protocol (Protocolo de Controle de transmissão)

UDP User Data Protocol (Protocolo de dados do usuário)

UDDI (Universal Description, Discovery and Integration)

WS Web Service (Serviço Web)

W3C World Wide Web Consortium (Consórcio gestor da Internet)

WSDL Web Services Description Language (Linguagem descritiva de Serviços Web)

XML eXtensible Markup Language (Linguagem de metadados extensível)

CAPÍTULO 1 - INTRODUÇÃO E CONTEXTUALIZAÇÃO

Hoje em dia as empresas enfrentam um mercado cada vez mais dinâmico. Para serem competitivas e rentáveis elas são obrigadas a responder às mudanças do mercado o mais rápido possível. A globalização, a alta variabilidade de produtos, a produção em massa, a pressão pela redução de custos e redução do tempo de vida do produto criam a necessidade permanente de mudança nos objetivos de negócio. As empresas que não são capazes de se adequar a este regime provavelmente não vão sobreviver.

A capacidade de mudança gerencial e tecnológica torna-se vital à manufatura e afeta diferentes níveis da empresa até o chão de fábrica. As empresas, especialmente as indústrias de manufatura, enfrentam, assim, a necessidade de capitalizar rapidamente as oportunidades quando elas surgem. O principal desafio ao fazê-lo é o de realizar essa evolução em organizações já estabelecidas, com processos, modelo de gestão, infraestrutura e sistemas de tecnologia de informação e comunicação (TIC) estabelecidos para transformá-la em uma "empresa flexível" com um modelo de negócios dinâmico e que seja interligado em diferentes níveis de gestão, tanto intra como interempresarial.

Atualmente a personalização através da estrutura de fabricação tradicional é difícil, com alto custo para configuração do maquinário, grande tempo do maquinário parado para ajustes, tornando-se tecnicamente inviável. Além de incompatibilidade entre sistemas de diferentes fornecedores e equipamentos com sistemas legados.

As empresas industriais modernas precisam ser ágeis e apoiar processos de tomada de decisão de forma dinâmica em diversos níveis. Como resposta a este cenário, a informação é uma necessidade crítica e que deve estar disponível em tempo hábil e na forma correta em todos esses níveis, o que é uma tarefa muito desafiadora.

A introdução de modernos sistemas de computação e tecnologias de comunicação, especialmente no chão de fábrica, com dispositivos possuindo maior capacidade de processamento e comunicação tem iniciado uma revolução nos processos e modelos de negócios. Com isso, podem ampliar a interoperabilidade entre todos os diferentes sistemas (hardware e software) existentes em uma fábrica, facilitando interações entre diferentes operações e áreas, além de maximizar interações com elementos fora da empresa como governo, cadeia de fornecimento e clientes.

Essa interoperabilidade é a espinha dorsal para tornar a manufatura mais flexível e inteligente. A manufatura para ser flexível (flexibilidade da linha de produção, flexibilidade na variabilidade dos produtos, flexibilidade da planta, flexibilidade do maquinário, flexibilidade de controle, flexibilidade de volume de produção, flexibilidade na distribuição e logística), necessita agregar valor em seus serviços e processos, e para tal é imprescindível o uso sistemas modernos de computação e tecnologias de comunicação, sendo base para uma nova forma de produção e automação industrial. Esse movimento tem ganhado vários apelidos, entre eles o de manufatura avançada, manufatura digital ou indústria 4.0. Independente do nome, este movimento busca extrapolar a tradicional estrutura hierárquica da fábrica e integrar clientes, fornecedores como elementos ativos do processo ao longo de todo o ciclo de vida do produto, desde sua concepção e design até pós-venda e descarte dos produtos. Para tal faz-se necessária uma reengenharia de processos intra-empresarial e inter-empresarial que irá impactar no chão de fábrica, que agora necessitará ser colaborativo e atender diversas fontes e sistemas.

Uma abordagem promissora que serve como base para este novo método de produção e que tem crescido rapidamente como uma solução padrão para publicação, conexão entre sistemas e acesso à informação é SOA- Service-Oriented Architecture ou Arquitetura Orientada a Serviços. SOA define interfaces e protocolos que permitem aos desenvolvedores encapsular funções e ferramentas orientadas a serviços e utiliza padrões abertos utilizados pela Internet.

Soluções efetivas de automação industrial compreendem a integração harmoniosa e alinhamento entre gerência em alto nível com as demais, operacionais, e também uma constante análise e monitoramento do comportamento dos dispositivos de automação no nível do chão de fábrica. Para atingir esse objetivo faz-se necessária quebrar a tradicional estrutura hierárquica da automação industrial e promover a inclusão de novas arquiteturas e métodos de produção que se beneficiem do avanço dos recursos de tecnologias de comunicação, Controle e Automação, como computação nas nuvens, SOA, Sistemas Ciberfísicos (SCF), novos protocolos de redes, utilizando dispositivos com maiores capacidade de processamento e comunicação e, sempre que possível, plataformas abertas e sistemas distribuídos.

Se em determinada empresa de manufatura, no nível do controle da linha de produção, (fabricação, montagem, pintura, etc.) não pode realizar o processo com a agilidade esperada, o resultado global da empresa será incapacidade de alcançar um bom desempenho, pois, a agilidade de uma empresa é sempre limitada por seu elemento menos ágil. Vale lembrar que a produção é que agrega valor em uma indústria. As demais áreas só agregam custos.

Desta forma, o chão de fábrica passa a ter um papel fundamental no apoio à agilidade geral da empresa, e os dispositivos no chão de fábrica são a fronteira da gestão, onde os requisitos, diretrizes e os fluxos de trabalho do processo de níveis superiores se materializam em uma série de ações e serviços físicos.

Para demostrar a relevância deste tema, o maior encontro entre os maiores líderes mundiais que envolve governos, empresas, organizações internacionais, sociedade civil e academia ocorreu na reunião anual do Fórum econômico mundial, em Davos na Suíça, em janeiro de 2016 e teve como tema central "Mastering the Fourth Industrial Revolution" (ECONOMIC, 2016) que pode se traduzir como "entendo a quarta revolução industrial". No

capítulo 2 se discutirá com maiores detalhes sobre esta nova geração da automação industrial, chamada de Indústria 4.0 ou manufatura avançada.

1.1 - Delimitações da pesquisa e problematização

Para atender a estas novas demandas, as empresas de manufatura necessitam melhorar seus processos para se adequar à nova realidade posta, e devem buscar soluções para os seguintes problemas já conhecidos:

- Longo tempo para o projeto, comissionamento e configuração do sistema;
- Complexidade dos ajustes dos sistemas industriais para atender a novos requisitos;
- Estruturas inflexíveis ou implementações complexas, hierárquicas e centralizadas;
- Aumento da escalabilidade da produção envolvendo exponencial complexidade;
- Incompatibilidade entre sistemas de diferentes fornecedores e equipamentos com sistemas legados;
- Chão de fábrica com pouca integração com outras instâncias do sistema produtivo.

Como descrito, inúmeros problemas ainda são recorrentes e tendem a se agravar com o aumento da complexidade dos equipamentos, visto que tem aumentado o grau de diversidade de fabricantes de dispositivos de entrada/saída, de protocolos de rede, de sensores e atuadores, assim como quanto à presença de sistemas heterogêneos e incompatíveis de hardware e de software.

Esta tese foi desenvolvida na área de automação e controle dos processos, com foco na aplicação da abordagem orientada a serviços para integrar sistemas heterogêneos no chão de fábrica.

Entende-se e delimita-se sistemas heterogêneos como a gama de dispositivos com diferentes capacidades de processamento (computadores,

robôs, controladores, interfaces com o usuário, sistemas embarcados, aparelhos celulares, etc.) que possuem diferentes sistemas operacionais e arquiteturas (Linux, Windows, Mac, Intel, AMD, ARM ...) e que se comunicam através de diferentes protocolos de redes (Ethernet, ISANET, CAN, MODBUS, TCP/IP, CoAP, etc.) onde necessitam interagir/cooperar de forma distribuída para realizar determinado processamento para atender a demanda de diferentes perspectivas da informação para diferentes usuários.

1.2 - Hipótese que orienta a pesquisa

Tomando como base a motivação e a contextualização do problema acima demonstrado, é fundamental se entender como as aplicações orientadas a serviço podem trazer benefícios concretos a essa nova geração de automação industrial. Para tal, propõem-se a hipótese que norteia este trabalho:

É POSSÍVEL SE APLICAR O PARADIGMA ORIENTADO A
SERVIÇOS PARA QUE SISTEMAS HETEROGÊNEOS EM
MANUFATURA POSSAM SE COMUNICAR DE FORMA FLEXÍVEL E
INTEROPERÁVEL SEM PERDA DE DESEMPENHO.

Durante o desenvolvimento deste trabalho pretende-se responder a esta hipótese, e para tal será realizada uma investigação sobre os princípios do paradigma de Arquitetura orientada a serviços (SOA), associado em conjunto com outros novos paradigmas oriundos da computação que podem trazer avanços na automação industrial da manufatura e implementados em diferentes casos de uso, desde robôs até uma fábrica que aplica um modelo de produção flexível.

1.3 - Objetivos

O objetivo da pesquisa, e a temática em geral pode ser visualizada graficamente na figura abaixo.

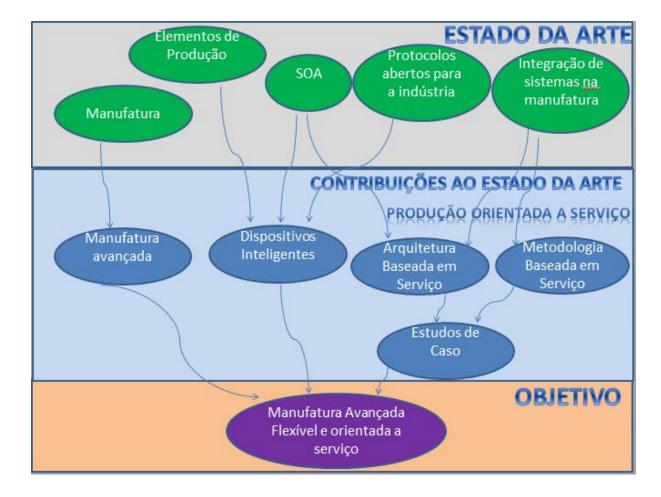


Figura 1 - Visão geral das áreas e contribuições da Tese

Fonte: Autor

Esta tese busca contribuir para o amadurecimento tecnológico e científico em direção a essa nova geração da manufatura industrial e para isso tem como objetivo principal:

heterogêneos no chão de fábrica da manufatura.

Para atingir este objetivo principal é esperado que a tese também atinja os objetivos específicos abaixo:

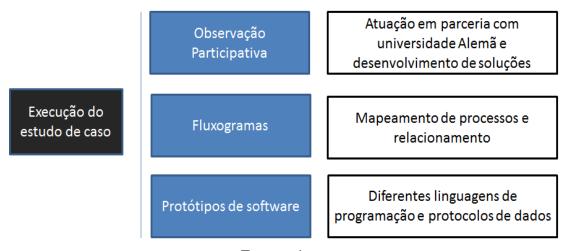
- Realizar uma avaliação entre novos padrões abertos de comunicação de dados entre dispositivos chão de fábrica e os níveis superiores da tradicional hierarquia da automação industrial;
- Propor uma Arquitetura de referência Modular e Orientada a Serviços (AMOS) para servir de subsídio para as indústrias desenvolverem soluções para sistemas heterogêneos;
- Propor uma metodologia para que as indústrias possam a partir seu atual estado tecnológico, buscar implantar novas tecnologias e com isso a implementação do novo modelo de produção;
- Testar e validar o modelo e a arquitetura propostos em estudos de caso.

1.4 - Metodologia

Para FONSECA (2002), a palavra metodologia vem do grego methodos, significa organização, e logos, estudo sistemático, pesquisa, investigação; ou seja, metodologia é o estudo da organização, dos caminhos a serem percorridos, para se realizar uma pesquisa ou um estudo, ou para se fazer ciência. Etimologicamente, significa o estudo dos caminhos, dos instrumentos utilizados para fazer uma pesquisa científica. Esta tese buscou melhorar as soluções já postas, iniciou-se pela necessidade de entender as limitações da estrutura hierárquica tradicional da automação industrial e buscou-se estudar as diferentes propostas para sua solução, caracterizandose como uma pesquisa exploratória. O objetivo de uma pesquisa exploratória é familiarizar-se com um assunto ainda pouco conhecido, pouco explorado. Ao final de uma pesquisa exploratória, deve-se avançar mais sobre aquele assunto, estando apto a construir ou validar hipóteses. E para isso, a pesquisa envolveu o levantamento bibliográfico e a análise e desenvolvimento de quatro estudos de caso que estimulam a compreensão do tema e a validação do conceito.

O problema geral emergiu de um contexto industrial, enquanto que a teoria proposta foi aplicada e os resultados foram utilizados para melhorar a objetivo problemas teoria, tendo como resolver práticos atuais, concomitantemente com a expansão do conhecimento científico. Para o levantamento do estado da arte uma pesquisa documental foi realizada. E por fim, a pesquisa tipo estudo de caso, que envolve o estudo profundo e exaustivo de um ou poucos objetos (GIL, 1995), foi necessária para o entendimento do contexto da manufatura, onde a metodologia proposta foi testada. A Figura 2 traz uma síntese dos aspectos considerados para a execução do estudo de caso: a observação participativa do autor, fluxogramas para registro das informações e os protótipos de softwares desenvolvidos.

Figura 2 - Algumas ferramentas e aspectos considerados para a execução do estudo de caso



Fonte: Autor

De acordo com (SJØBERG, ANDA et al., 2002) "Estudo de Caso é uma inquirição empírica que investiga um fenômeno contemporâneo dentro do

seu contexto de vida real, especialmente quando os limites entre fenômeno e contexto não são claramente evidentes". Assim, sua aplicação mais importante é em testar teorias e hipóteses.

Esta tese foi realizada através de uma parceria entre duas universidades: a Universidade Federal da Bahia – UFBA e a Universidade de ciências Aplicadas de Emden/Leer na Alemanha. A parceria com o centro de pesquisa alemão I2AR (Institute for Industrial Informatics, Automation and Robotics) coordenado pelo Prof. Armando Walter Colombo da Universidade de ciências Aplicadas de Emden/Leer e pesquisador da Schneider ensejou a oportunidade de se desenvolver trabalhos aplicados que redundaram nos estudos de caso que validam os conceitos desenvolvidos.

Esses estudos de caso, e os respectivos testes e validação, permitiram aprofundar a compreensão dos benefícios e limitações da inserção de SOA na área de automação industrial em seu sentido mais estrito, imposto pelos requisitos do chão de fábrica.

Definido o tipo de investigação científica a ser aplicada, uma pesquisa documental foi elaborada a partir da análise da literatura, essencialmente livros, artigos de periódicos, relatórios técnicos projetos pesquisa, algo fundamental para a elaboração do estado da arte. A partir de uma análise e observação participativa foram desenvolvidos estudos de caso com diferentes linguagens de programação (.Net, Java e C) e utilizado diferentes modelos de dados orientado a serviços (OPC UA, DPWS, GSOAP).

1.5 - Estrutura do Trabalho

Inicialmente, foi apresentada uma introdução com a proposta de trabalho, o escopo da pesquisa, o objetivo geral e os objetivos específicos, visando situar o leitor sobre o contexto e os limites da pesquisa. Os próximos capítulos se sucedem conforme abaixo.

O capítulo 2 tem como proposta levantar a revisão bibliográfica e

revisar as temáticas que serão utilizadas nesta Tese.

O capítulo 3 introduz o modelo AMOS e descreve propostas para desenvolvimento de soluções baseados em serviços.

O capitulo 4 apresenta a arquitetura AMOS, que é uma diretriz para o desenvolvimento de soluções orientadas a serviços para a manufatura e propõem uma metodologia para a aplicação da metodologia AMOS e um modelo de migração/implantação desse novo modelo de produção.

O capítulo 5 descreve estudos de casos realizados e a validação da arquitetura e metodologia proposta.

O capítulo 6 faz uma análise das etapas realizadas, conclusão, limitações e trabalhos futuros baseados na Tese.

CAPÍTULO 2 - REVISÃO DA LITERATURA

Este capítulo trata da revisão da literatura e discute tópicos relevantes que serão abordados na tese.

Como descrito no capítulo 1, é essencial promover a integração da manufatura reduzindo a distância existente entre as camadas de alta gestão que usam soluções de TIC. Por exemplo, sistemas ERP (Enterprise Resources Planning - Planejamento de Recursos da Corporação), e o controle de mais baixo nível, como sistemas que controlam o maquinário do chão de fábrica e buscam ter acesso/controle de informações em tempo real. De forma que a indústria se torne uma entidade ágil em que os seus elementos interajam de forma harmônica, deixando a comunicação mais fluida, não-hierárquica e sem necessidade de improvisações, ou a necessidade de exportações para aplicações terceiras, pois a granularidade de informação obtida por um único sistema não atende a gestão, muitas vezes não automatizadas e sem tempo real.

2.1 - Evolução da automação industrial

O aumento da competitividade (local, global) aumentou necessidade de redução de custos e aumento de flexibilidade no ciclo de vida do produto (flexibilidade da linha de produção, flexibilidade na variabilidade dos produtos, flexibilidade da planta, flexibilidade do maquinário, flexibilidade de controle, flexibilidade de volume de produção, flexibilidade na distribuição e logística). Para atender a demanda por melhor controle ocorreu a inclusão de máquinas/equipamentos controlados por computador, assim como o uso intensivo de computadores e meios de comunicação (redes) têm sido crescentes.

Inicialmente o controle era quase inexistente e era predominantemente manual através do operador humano, porém, com o avanço da tecnologia, esse controle passou a ser mais simplificado, incialmente com o uso de relés, e evolui principalmente com o aparecimento do Controlador Lógico Programável (CLP), que permitiu uma melhor programação e controle do processo e tornou a detecção de certas falhas mais simples e rápida.

Hoje em dia, a distribuição da função de controle é feita entre vários computadores de processos. Dessa forma é possível reduzir os custos através da utilização de subsistemas de E/S remotos, ligados a componentes de controle dos processos, já que os maiores custos se dão pela interligação entre transdutores, controladores e atuadores. Sistemas de controle distribuídos utilizam protocolos que deveriam ser padronizados para comunicação de dados, porém essa padronização é usualmente limitada. Em grandes plantas industriais, as funções de controle são distribuídas para diferentes sistemas SCADA e CLPs comunicando-se via rede com outras estações, RTUs (*Remote Terminal Units*) ou Unidade Terminal Remotas, CLPs através de protocolos industriais padronizados (HART, fieldbus, Profibus, Modbus, DeviceNet, CAN, etc).

O investimento no chão de fábrica é, em geral, muito alto e de longo prazo. Outro agravante são as estruturas rígidas e hierárquicas que não conseguem atender a constante inovação necessária para as empresas. Empresas industriais modernas requerem processos e linhas de produção altamente flexíveis e reconfiguráveis para serem competitivas em um mercado global instável. Especialmente nos últimos anos, um profundo progresso tem sido realizado nesta área, a ponto de a academia entender que hoje se está em via de uma quarta revolução industrial.

A figura 3 resume as revoluções industriais, nas quais os avanços tecnológicos têm impulsionado aumentos gigantescos na produtividade. Na primeira revolução, a máquina a vapor alimentou fábricas no século XIX. Na segunda revolução a energia elétrica, associada com metodologias mais eficientes de linhas de produção e a evolução no transporte (especialmente

estradas) impulsionou a produção em massa no início do século XX. A indústria tornou-se automatizada na década de 1970 com a introdução do computador no chão de fábrica e maior automação no processo fabril, considerada a terceira revolução industrial. Nas décadas que se seguiram os avanços foram incrementais, baseadas nas tecnologias emergentes que transformaram TI, comunicações e *e-commerce*. Agora, porém, verifica-se o advento de uma quarta onda de avanço tecnológico: a ascensão da nova revolução industrial digital que é alimentada por avanços tecnológicos e que se baseiam principalmente nos novos conceitos de digitalização dos processos e sistemas ciberfísicos.

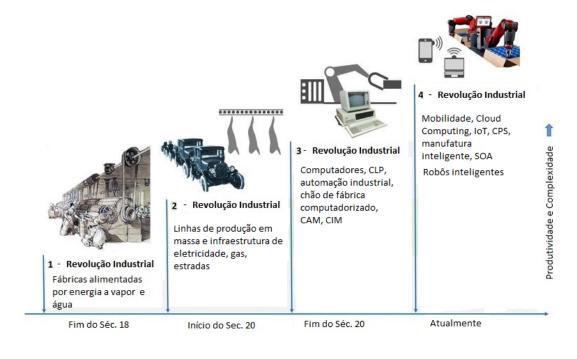


Figura 3 - Linha do tempo das evoluções industriais

Fonte: Elaborado pelo autor baseado em (ACATECH, 2014)

Os esforços para se posicionar adequadamente ante esta nova revolução estão sendo incentivados em diversos países (KAGERMANN, HELBIG et al., 2013), Brasil incluso (RIVERA, TEIXEIRA et al., 2014) e (MEMÓRIA, 2015) usando diferentes nomenclaturas, através de muitos grupos de trabalho e abertura de projetos de pesquisa e financiamento de

atividades de investigação que ligam a indústria e as universidades para promover inovação, mas inovação aplicada e com protótipos ao final. Por exemplo, na Alemanha, o governo está fortemente promovendo o "Industrie 4.0" (KAGERMANN, HELBIG et al., 2013) e formando o "Grupo de Trabalho da Indústria 4.0". Nos EUA, o programa de governo indústria inteligente é chamado advanced manufacturing (RUSSELL and FIELDING, 2014). Segundo dados do IPEA (CASSIOLATO, 2013), a China tem investido através em políticas de forma sistêmica em um projeto de desenvolvimento em que o progresso científico e tecnológico são elementos centrais, em especial na área da automação com novas tecnologias. No Brasil tem se usado a nomenclatura manufatura avançada, mas, como se viu no fórum mundial em Davos, a nomeclatura quarta revolução industrial tem sido usada em diversos países, especialmente na Europa, capitaneado pela Alemanha, que cunhou o termo indústria 4.0. Muitos projetos de pesquisa e inovação diferentes estão atualmente em curso, para construir novas tecnologias e padrões para chegar a realizar a quarta revolução industrial. Paralelamente, muitas empresas industriais, os usuários finais e integradores, estão mostrando um interesse crescente nestes resultados.

2.2 - Manufatura avançada

Como descrevemos na seção anterior o termo manufatura avançada ou Indústria 4.0 descreve uma fase evolutiva da terceira revolução industrial, é uma nova fase de organização e controle industrial que impacta em todo o ciclo de vida dos produtos.

Iniciou-se discutindo essa nova revolução tecnológica no tópico anterior e este tópico irá explicar com mais detalhes como se materializa este conceito e como esta tese busca contribuir cientificamente com o avanço da automação, utilizando a orientação baseada em serviços.

O Departamento de Defesa dos EUA divulgou um relatório interessante sobre o tema manufatura avançada (IDA, 2012). Este documento faz uma

discussão da manufatura avançada sob a ótica da tecnologia, partindo de uma análise sobre tendências gerais da indústria nessa perspectiva. Baseouse em entrevistas com quase uma centena de *experts* estadunidenses da academia, indústria e Governo. O relatório identificou cinco grandes tendências aplicáveis aos vários setores da indústria manufatureira (SHIPP, GUPTA et al., 2012):

- (a) O papel ubíquo da tecnologia da informação (TI).
- (b) O uso crescente da modelagem e da simulação computacional na manufatura.
 - (c) A complexa gestão das cadeias globais de fornecimento.
 - (d) A capacidade de mudança dos sistemas de produção.
 - (e) A adoção da manufatura sustentável.

O relatório do departamento de defesa norte americano (SHIPP, GUPTA et al., 2012) afirma que a manufatura avançada é um prolongamento do processo de automação industrial, levado agora a explorar as várias possibilidades abertas pelo uso das TICs.

O conceito manufatura avançada define uma nova geração de manufaturas ágeis, com produtos de maior valor agregado, que são dinamicamente reconfiguráveis para permitir auto-organização, evolução e adaptação ao longo do ciclo de vida do produto, para enfrentar os desafios de produção de forma contínua. Se adaptando mais rapidamente para a evolução das condições de mercado, redução de tamanhos dos lotes, além do curto prazo de entrega associado com a alta qualidade exigida. Este conceito promove maior integração desde sua concepção e design até a pós-venda, contato com o consumidor/cliente e o descarte. Estes avanços na tecnologia que formam a base da manufatura avançada já são utilizados na fabricação, mas com a manufatura avançada serão utilizadas com um contexto mais amplo e vão transformar a produção: células de produção deixarão de ser isoladas, otimizadas trabalharão juntas de forma totalmente integrada,

automatizada e melhorando o fluxo de produção. Com isso, maior eficiência e mudando a relação de produção tradicional entre fornecedores, produtores e clientes assim como entre homem e máquina (LEITÃO, COLOMBO et al., 2016), (AHMAD, KONG et al., 2013).

Descreve-se agora cada elemento da manufatura avançada. Estes elementos interagem entre si e se apoiam para buscar a evolução tecnológica Para esta descrição foram utilizadas prioritariamente duas fontes: primeiramente o relatório da comissão europeia sobre inovação: Transformação Digital da Indústria e das Empresas Europeias de 2015 (COMMISSION, 2015) e no relatório da Confederação Nacional da Indústria (CNI) de 2016: Desafios para Indústria 4.0 no Brasil (CNI, 2016).

Sistemas Ciberfísicos: Os Sistemas Ciberfísicos (SCF, em inglês Cyber Physical Systems) são elementos importantes na manufatura avançada. Os SCFs são compostos por entidades físicas controladas ou monitoradas por algoritmos baseados em computador, que evoluíram a partir de dispositivos mecatrônicos com capacidades aumentadas de comunicação e conectividade.

Segundo (BAHETI and GILL, 2011) o termo SCF refere-se "a uma nova geração de sistemas com capacidades computacionais e físicos integrados que podem interagir com os seres humanos através de muitas novas modalidades. A capacidade de interagir e expandir os recursos no mundo físico através de computação, comunicação e controle é um elemento essencial para desenvolvimentos tecnológicos futuros" (será descrito em maiores detalhes no tópico 2.5).

Simulação: Na fase de engenharia (pesquisa e desenvolvimento), simulações 3-D de produtos, materiais e processos de produção já são utilizadas, mas, no futuro, as simulações serão parte de um processo integrado, desde o design, com ferramentas que estarão integradas com o processo de produção e gerarão o código entendido pelo maquinário (LADDER, Function Blocks, IEC 61311), facilitando a prototipagem e reduzindo custos (COMMISSION, 2015). Com isso reduzir-se-á a lacuna

existente entre o design, prototipagem e a produção. Estas simulações gerarão dados em tempo real para espelhar o mundo físico em um modelo virtual, que pode incluir máquinas, produtos e seres humanos. Isso permite que os operadores possam testar e otimizar as configurações da máquina para o próximo produto na linha no mundo virtual antes da transição física, reduzindo dessa forma os tempos de configuração da máquina e aumentando a qualidade (SANTOS, BARBOSA et al., 2015). Além disso, sistemas utilizarão realidade aumentada que permitem uma variedade de serviços como treinamento, a seleção de peças em um armazém e o envio de instruções de reparação através de dispositivos móveis. Estes sistemas estão atualmente em sua infância, mas, no futuro, as empresas vão fazer uso muito mais amplo da realidade aumentada para fornecer aos trabalhadores informações em tempo real para melhorar a tomada de decisões e procedimentos de trabalho (NAGORNY, COLOMBO et al., 2014).

Manufatura aditiva: A usinagem é o processo mecânico que retira material de peças brutas, ou da matéria prima, a fim de chegar a uma forma final, enquanto no processo de manufatura aditiva é um processo de adicionar matéria prima para formar uma peça final. As empresas começaram a adotar manufatura aditiva, principalmente a impressão 3D, que é usada para produção de protótipos e de componentes individuais, além disso, introduz-se o conceito de manufatura híbrida, onde máquinas combinam os dois mundos da manufatura aditiva e usinagem. Com manufatura avançada, este método de fabricação será amplamente utilizado para a produção de pequenos lotes de produtos personalizados que oferecem vantagens de construção, tais como desenhos complexos e leves. Com a evolução da impressão 3D novos materiais estão sendo produzidos com maior resistência e com menor tempo de fabricação (RÜßMANN, LORENZ et al., 2015).

Por exemplo, as empresas aeroespaciais já estão usando a impressão 3D para aplicar a novos projetos que reduzem o peso das aeronaves, reduzindo as suas despesas de matérias-primas, como o titânio (SCHUMACHER, 2016).

Sistemas autônomos: Com o uso intensivo de sistemas ciberfísicos, manufaturas têm utilizado elementos industriais principalmente os robôs para lidar com serviços complexos, estão se tornando mais autônomos, flexíveis e cooperativos. Eventualmente, eles vão interagir entre si e trabalhar lado a lado com segurança com os seres humanos e aprender a partir destes. (RÜßMANN, LORENZ et al., 2015).

Por exemplo, Kuka, um fabricante de robôs europeu, oferece robôs colaborativos que interagem entre si (KNEPPER, LAYTON et al., 2013). Estes robôs são interligados para que eles possam trabalhar em conjunto e ajustar automaticamente as suas ações de acordo com o próximo produto na linha de produção. Sensores e unidades de controle permitem estreita colaboração com os seres humanos. Da mesma forma, outro fornecedor de robôs, a ABB, está lançando um robô de dois braços chamado Yumi que é projetado especificamente para montar produtos ao lado de seres humanos. Dois braços acolchoados e visão computacional permitem a interação segura e fazer reconhecimento do humano e da atividade realizada.

Computação nas nuvens/Big Data: As empresas já estão usando a computação nas nuvens para algumas aplicações corporativas e de gestão, com a manufatura avançada, mais empresas compartilharão informação e informação essa relacionadas à produção, exigindo maior compartilhamento e em tempo real. Ao mesmo tempo, o desempenho das tecnologias nas nuvens irá melhorar, alcançando tempos de reação de apenas alguns milissegundos. Como resultado, dados de máquina e funcionalidades serão cada vez mais mobilizados para a nuvem, permitindo que mais serviços baseados em dados para sistemas de produção. Mesmo os sistemas que monitoram os processos de controle podem disponibilizados via nuvem. Com a manufatura avançada e o uso de sistemas ciberfísicos haverá maior capacidade dos elementos produtivos processarem dados e transforma-los em informações valiosas (BRETTEL, FRIEDERICHSEN et al., 2014). Há oportunidades ilimitadas de usar esses dados. Pode-se ver imediatamente como uma fábrica está em execução e como o processo pode ser realizado de forma mais produtiva do que fazemos atualmente. Por exemplo, pode-se usar técnicas de análise e mineração de dados para prever a falha, autoconfiguração e capacidade de adaptação às mudanças e componentes individuais da linha de produção, tais como robôs ou transportadores podem ser modificado e/ou substituídos quando necessário. Como os padrões de comunicação de dados entre os componentes do desenvolvimento, linhas de produção tornar-se-ão menos dependente de componentes individuais dos fabricantes individuais (COLOMBO, BANGEMANN et al., 2014).

Inteligência computacional: Com a manufatura avançada e o uso de sistemas ciberfísicos haverá maior produção de dados e capacidade dos elementos produtivos processarem dados e transformá-los em informações valiosas, para isso são necessários algoritmos com o uso intensivo de inteligência artificial e aprendizagem de máquina. Poderá ser possível realizar previsões e análises preditivas com maior eficiência, tornando o processo mais seguro e com maior produtividade do que atualmente (BRETTEL, FRIEDERICHSEN et al., 2014). A inteligência computacional se baseia em algoritmos inteligentes (inteligência artificial e matemática). Algumas empresas já têm usado inteligência computacional em algumas aplicações corporativas, mas com a manufatura avançada, será utilizada em um contexto maior com colaboração entre os elementos envolvidos que compartilharão informação relacionadas à produção.

Virtualização e conectividade: Virtualização tem a ver com a digitalização e integração de processos e fluxos de trabalho via sistemas de toda cadeia de valor para a criação do produto, incluindo logística e pósvenda, com monitoramento em tempo real e promovendo uma convergência entre o mundo real e virtual (KAGERMANN, 2015). Isso gera um novo ecossistema de negócios com a implementação das demais tecnologias acima elencadas. Isso traz consigo a necessidade de um novo modelo de negócios e canais digitais para atender a demanda por produtos digitais, serviços digitais e interações entre clientes, fornecedores e demais *stakeholders* (vide glossário).

Para a materialização da manufatura avançada sensores, máquinas, peças e sistemas de TI serão conectados ao longo da cadeia de valor da empresa e também para além de uma única empresa. Segundo dados de relatórios da Gartner Group (GROUP, 2015), que lista as tendências tecnológicas e estratégicas, no ano de 2013, mais de 50% das conexões de Internet são provenientes de dispositivos inteligentes (chamados de "coisas").

A conectividade é base para tal e juntamente com o aumento da capacidade de processamento dos sitemas ciberfísicos a comunicação com maior complexidade será possivel, facilitando a interação e integração. Segundo o relatório acima citado em 2011, já existiam mais de 15 milhões desses dispositivos na Web, com mais de 50 bilhões de conexões intermitentes. Em 2020, teremos mais de 30 bilhões de dispositivos ligados, com mais de 200 bilhões com conexões intermitentes.

A integração é ubíqua. A manufatura avançada busca ter maior controle com mais informação do processo produtivo e para tal uma maior integração entre os elementos industriais faz-se necessária. Protocolos baseados no padrão da Internet permitirão comunicação de dados entre os insumos, maquinário, linhas de produção tornando-se menos dependente de componentes individuais de fabricantes específicos.

Duas tendências tecnológicas compõem esse elemento da manufatura avançada: a Internet das coisas (IoT – Internet of Things) e a Internet dos Serviços (IoS) (SORIANO, HEITZ et al., 2013). A Internet das coisas tem um grande potencial para alterar profundamente a maneira pela qual interage-se com o mundo físico e virtual, e pavimentar o caminho para um novo tipo de serviços que vai acompanhar onde quer que se esteja. No entanto, este potencial ainda não foi realizado devido parcialmente à complexidade atual enfrentado por dificuldades de integração. O nível de complexidade decorre da ausência de vários blocos para sua construção. O estado atual faz com que prestadores de serviços lidem com toda a pilha de hardware e software necessários para realizar tais serviços. O esforço necessário começa a partir de comunicar-se com um conjunto heterogêneo de coisas, através da criação

de serviços específicos e, possivelmente, a composição de vários desses serviços, em tempo de execução adequado e suporte de comunicação. A internet das coisas será detalhada no tópico 2.7. Serviços são onipresentes, as "coisas" conectadas só têm sentido se fornecerem algum serviço útil. Esse serviço precisa ser descrito de uma forma que as "coisas" e os humanos possam colaborar, para isso a semântica e sintaxe dos serviços deve ser bem definida e explicitada, em um mundo superconectado e integrado a abordagem orientada a serviço parece ser promissora (ACATECH, 2014).

Como descrito na figura 4 diversas tecnologias dão subsídio à construção da manufatura avançada, e esta tese busca aprofundar o conhecimento e contribuir com o tópico integração de sistemas no modelo da manufatura avançada, onde ocorre uma mudança na estrutura de integração tipicamente vertical (figura 5) para um modelo de gestão do ciclo de vida.

Esta tese busca **investigar uma melhor integração a partir da abordagem orientada a serviços.** Pois esta nova geração de automação industrial necessita de uma interação menos rígida e com baixo acoplamento (vide glossário) caminhando na direção de dispositivos e máquinas *conecte e produza*, ou seja, que podem, ao serem conectados na rede, detectar automaticamente os outros elementos, interagindo e provendo seus serviços físicos de forma virtual sem necessidade de configurações adicionais ou específicas. Esta inserção digital tem se expandido por toda a planta, com maior tratamento dos dados que anteriormente eram descartados, com a planta monitorada regularmente, e com mais detalhes. Veremos maiores detalhes sobre orientação a serviços no próximo tópico.

L4: Alta Gestão **Aplicações** Vertical Ciclo de L3: Produção Modelo de Horizontal L2: Controle dados Tipos de L1: Chão de Integração Chão de Fábrica Fábrica Processos Físicos LO:

Figura 4 - Nova de estrutura empresarial baseada em modelo de dados

Fonte: Elaborado pelo autor, baseado em (REBOREDO and KEINERT, 2013)

A manufatura avançada busca quebrar a tradicional estrutura hierárquica verticalizada (Figura 4) para uma estrutura com maior integração entre os componentes de produção até os mais altos níveis de gestão. Esta integração não é mais vertical ou horizontal, sendo colaborativa e feita ao longo do ciclo de vida entre diversos elementos de produção. Insumos, maquinários, computadores de processo agora tem como objetivo monitorar, controlar, rastrear todas as etapas de produção, com a gravação de dados em tempo real em cada fase de modo automatizado e que está disponível para a próxima fase de produção. Com isso, ganhasse maior rastreabilidade (uso de sistemas de aquisição de dados) e sistemas CEP (Controle Estatístico de Processo), já que o produto possui *tags* e utiliza-se leitores de códigos e sistemas RFID, que interagem com o processo e o automatizam sem a intervenção humana.

A produção se torna flexível e se adapta para a sequência produtiva, reservando o maquinário e ferramentas de forma autônoma, mudando as

linhas de produção em tempo real para otimizar a produção e reduzir tempo ocioso. Além disso, esses dados são compartilhados intra e inter empresas, onde sistemas de logística e distribuição (próprio e de fornecedores) são alimentados e sabem previamente as necessidades de insumos e os pedidos que devem ser liberados. Outro aspecto importante é a participação do cliente desde a concepção do produto e no pós-venda e descarte. Sistemas de relacionamento e aplicativos *mobile* estarão presentes e aproximarão o cliente da empresa, que iniciará uma retroalimentação com o *feedback* dos usuários em busca de maior satisfação e qualidade.

Essa integração acima descrita utiliza um modelo baseado em dados e aplicações, com elementos se comunicando baseados em serviços. No capítulo 3 serão descritos diferentes modelos de dados utilizados com essa finalidade e na sequência será abordado com maiores detalhes o modelos de dados orientados a serviços.

2.3 - SOA

Originária da área de TI, com foco em alinhamento do alto nível de gestão, Service-oriented Architecture (SOA) ou arquitetura orientada a serviço (Vide Glossário) é atualmente uma abordagem amplamente aceita tanto para negócios e integração de sistemas corporativos. SOA promove a descoberta, baixo acoplamento (vide Glossário), abstração (vide Glossário), autonomia e composição de serviços que se baseiam em padrões web abertos e que podem fornecer uma contribuição importante para o domínio da automação industrial (COLOMBO, 2012). SOA permite que os clientes possam acessar os serviços sem o conhecimento ou controle sobre a sua real implementação, pois abstrai a complexidade envolvida.

Davis (2009), define SOA como uma arquitetura que consiste em serviços de negócios, que são combinados para criar novas aplicações e implementar processos de negócios. De acordo com (ERL, 2005), os principais características de SOA são:

- Baixo Acoplamento: relacionamento que minimiza a dependência e requer apenas que os serviços mantenham um mínimo conhecimento mútuo.
- Contrato de prestação de serviços: acordo de comunicação comumente baseado em padrões abertos da web, conforme descrito na descrição do serviço (também conhecido como interface de serviço ou contrato).
- Autonomia: controle local sobre a lógica que um serviço encapsula.
- Abstração: as minúcias de lógica e recurso estão ocultas do resto do sistema.
- Reutilização: a lógica complexa pode ser dividida em diferentes serviços atômicos que podem ser posteriormente compostos e reutilizados em outras aplicações.
- Composição: os serviços podem ser coordenados e montados para formar serviços mais complexos.
- Sem estado: os serviços não reter informações específicas de uma determinada atividade.
- Descoberta: os serviços devem ser descritivos para serem encontrados e acessados através de mecanismos de descoberta.

Como os serviços são tipicamente fornecidos por diferentes aplicações que registram e divulgam seus serviços em rede, SOA pode ser vista como a integração da aplicação, porque SOA não é apenas um tipo específico de integração, mas também uma forma arquitetônica adequada para construir qualquer tipo de solução de integração.

A integração busca a interoperabilidade e, como descrito em IEEE 1990), refere-se à "capacidade de dois ou mais sistemas ou componentes de trocar informações e poder utilizar estas informações trocadas, apesar das diferenças na linguagem, interface, e plataforma de execução". A interoperabilidade depende ainda da semântica, semântica esta que depende da capacidade de interpretar o significado da informação trocada e precisamente o que se espera dos sistemas ou componentes envolvidos

(HEILER, 1995). Neste contexto, a informação trocada deve ser claramente definida, ou seja, a forma que é percebida pelo transmissor é a mesma que deve ser entendida pelo receptor.

A Arquitetura Orientada a Serviços (SOA) é um paradigma que cresceu rapidamente como uma solução padrão para publicação e acesso à informação em um mundo cada vez mais com uma onipresente Internet.

Esta nova abordagem que usa intensamente interfaces e protocolos padrões definidos, permite aos desenvolvedores encapsular funções e ferramentas como serviços em que os clientes podem acessar sem o conhecimento ou controle sobre sua aplicação (COLOMBO, BANGEMANN et al., 2014). SOA estabelece um modelo de arquitetura que visa melhorar a eficiência, a interoperabilidade, agilidade e produtividade posicionando serviços como o principal meio ao longo da solução lógica. Representa apoio para a realização dos objetivos estratégicos associados que se implementam através de serviços computacionais.

O interesse cada vez maior em SOA tem sido estimulado por uma tendência da indústria: A tecnologia *Web Services*. Embora os *Web services* não se traduzam necessariamente em SOA, e nem todas as aplicações SOA são baseadas em *Web Services*, estes últimos reconhecidamente têm ajudado a trazer SOA para um público mais amplo, enquanto conceitos e princípios SOA irão contribuir para mais iniciativas de sucesso para serviços aplicados à web.

Esta tese busca aplicar o conceito de SOA para o chão de fábrica com dispositivos com capacidade SOA ou adaptação para a migração para SOA, vide figura 5, com uma nova abordagem para a tradicional estrutura ISA-95.

NOVA GERAÇÃO de APLICAÇÕES

SCADA

CONTROLE

SENSORES/ATUADORES

PROCESSOS PRODUTIVOS

Figura 5 - Transformação da arquitetura ISA-95 em serviços

Fonte: Elaborado pelo autor baseado no Projeto de pesquisa FP7 IMC-AESOP

Com o uso de SOA informações de contexto provenientes de diversos sistemas heterogêneos podem ser obtidas de forma transparente ao usuário por meio de serviços. A arquitetura orientada a serviços possui características como abstração, troca de dados via mensagem, fraco acoplamento e autonomia, que proporcionam a evolução das aplicações por meio da integração de sistemas e que relacionam adequadamente com SOA (DELSING, ELIASSON et al., 2011). Cada elemento de diferentes níveis da estrutura ISA-95 não mais é um fornecedor de dados para a camada acima, com o conceito de serviços esses elementos passam a ser clientes ou servidores a depender da necessidade do processo e a interação se baseia em troca de mensagens.

Para que esta tese pudesse propor a arquitetura AMOS foi necessário levantar o estado da arte de SOA aplicada a manufatura, e importantes trabalhos foram realizados e que embasaram a presente tese.

F. Jammes and H. Smith (JAMMES and SMIT, 2005) foram os primeiros a discutir os desafios de SOA para a automação industrial, onde descreveram no projeto europeu SIRENA¹ que se iniciava, e que descreve a estrutura de comunicação orientada para o serviço proposto pelo Serviço de Infraestrutura em tempo real incorporados aplicações de rede do projeto SIRENA para redes de dispositivos industriais, e discute os beneficios esperados desta abordagem. Seu objetivo era criar um *framework* (Vide Glossário) que contemplasse a proposta de um novo protocolo para que os dispositivos se comunicassem baseados em XML, *Web Services*, *e definido o protocolo DPWS (Device Profiles for Web Services)*, que foi introduzido no Windows Vista pela Microsoft, o projeto foi encerrado, mas ainda não se obteve como resultados produtos comerciais que pudessem aplicar o novo protocolo.

A tecnologia foi portada para os sistemas operacionais ThreadX, Linux, Solaris, Windows CE e XP.

O resultado do projeto SIRENA foi um marco para dispositivos embarcados baseados no protocolo DPWS que foi a primeira implementação mundial DPWS com base em gSOAP (JAMMES and SMIT, 2005).

O projeto SOCRADES (CANNATA, GEROSA et al., 2008) foi uma continuação do projeto SIRENA, teve como principal objetivo a desenvolver um projeto, execução e plataforma de gerenciamento de sistemas de automação industrial de última geração, explorando a arquitetura orientada a serviços no nível do dispositivo.

Os projetos SIRENA, SODA, SOCRADES, IMC-AESOP foram uma sequência de projetos de pesquisa financiados pela União Europeia em um programa chamado (I*PROMS), com ênfase em criar uma rede de excelência NoE (*Network of Excellence*), que buscava formas e tecnologias inovadoras de

^{1 &}lt;u>SIRENA</u>- Service Infrastructure for Real-Time Embedded Networked Applications (SIRENA) projeto do conglomerado Europeu pertencente ao European Research and Development Program ITEA.

produção e sistemas com isso gerando novas tecnologia e uma agenda de pesquisa alinhada. Nargony (2014) realizaram uma pesquisa com os sistemas orientados a serviços (NAGORNY, COLOMBO et al., 2014), Colombo (2013) propuseram uma forma de colaboração na manufatura com vista neste novo modelo de produção (COLOMBO, KARNOUSKOS et al., 2013).

Gonçalo Cândido, em sua tese de doutorado, propôs uma arquitetura modelo para construção de dispositivos inteligentes com suporte ao ciclo de vida (CÂNDIDO, 2013). Arquitetura esta que serviu como base para o entendimento do ciclo de vida do produto e reflexões que serviram de base para esta tese.

Devido à importância entre a conexão das atividades do chão de fábrica e as da gestão de alto nível, é fundamental se avançar para criar serviços de alto nível mais sofisticados apoiando a tomada de decisão mais confiável. Nesta direção, Colombo (COLOMBO and HARRISON, 2008), (COLOMBO, SCHOOP et al., 2004), Stamatis, (KARNOUSKOS and COLOMBO, 2011) tem relevantes contribuições desenvolvendo protótipos e reflexões sobre tecnologias e metodologia voltadas a serviço. Esta sofisticação na tomada de decisão ajuda as empresas a aderir ou a criação de cadeias de abastecimento mais complexas, e o projeto de pesquisa IMC-AESOP (AESOP, 2014) buscou investigar protótipos em grande escala, buscando controlar, centenas ou milhares de dispositivos trabalhando orientados a serviço. Limitações e resultados deste projeto de pesquisa nortearam este trabalho.

Zeeb e outros (2007) propuseram o uso de DPWS dispositivos com perfil de Web Services (DPWS). Para tanto criaram protocolos subjacentes e um kit de ferramentas de implementação DPWS baseada em C e SOAP e discutem seu estado atual. Estes projetos desenvolveram ferramentas de desenvolvimento (APIs) para dispositivos que se comunicam via serviços que foram reutilizados nesta tese na elaboração de estudos de caso.

2.4 – Modelos Empresariais

A área da modelos empresarias nasceu com uma necessidade de se discutir soluções para gerenciar a complexidade cada vez maior em sistemas distribuídos (ZACHMAN, 1987). Começou a ganhar força no final dos anos 1980 como uma resposta aos problemas apresentados na gestão de empresas que rapidamente cresceram e se viram confrontados com a globalização, grande distribuição geográfica e uma grande gama de produtos e de serviços.

O modelo de referência empresarial preocupa-se principalmente com a arquitetura e infraestrutura de TI dentro da empresa, e as vias de informação dentro dela, com um olho para a gestão dos diferentes aspectos da produtividade de forma oportuna e eficiente. Um relatório da Microsoft (2007) afirma que o crescimento da revolução do computador no setor empresarial nos últimas décadas resultou em uma mistura de soluções pontuais para determinadas necessidades. Porém essa diversidade de sistemas torna cada vez mais dificil se alinhar com necessidades das empresas. O resultado final pode ser sucintamente descrito como "mais custos, menos valor", pois as empresas investem em diversas soluções de diferentes fornecedores, mas que tem dificuldade de integrar as suas soluções proprietárias para prover mais valor agregado ao cliente.

O tema de modelos empresariais é especialmente importante para as empresas de manufatura, pois afeta diretamente as vias de comunicação entre a gestão empresarial e as linhas de produção. Essas vias se tornam essencial na implementação de visão estratégica da empresa em face da mudança mundial e seus requisitos.

2.4.1 - ISA 95

A integração das regras de negócios para sistemas de manufatura é difícil e cara, transformar decisões de gestão em atividades alinhadas para diferentes níveis hierárquicos, em especial com o chão de fábrica é um desafio.

O ISA-95 é um modelo de referência para empresas industriais que fornece padrão e terminologia para descrever as interfaces entre os sistemas de negócios de uma empresa e suas operações de fabricação e sistemas de controle, por isso a importância de sua citação e detalhamento. Os modelos e terminologia apresentadas neste padrão buscam enfatizar boas práticas de integração de sistemas de controle com os sistemas da empresa durante todo o ciclo de vida dos sistemas.

O ISA-95 foi introduzido pela Sociedade Internacional de Automação, a fim de fornecer uma terminologia padrão para o campo da automação da produção, para assim facilitar a comunicação entre os fornecedores e fabricantes. Foi baseado no modelo PERA (WILLIAMS, 1998), e além disso, fornece informações de operação e modelos consistentes que definem como a informação deveria ser usado e transmitida no âmbito empresarial.

Pode ser usado para melhorar as capacidades de integração existentes de operações e sistemas de controle com sistemas corporativos de fabricação.

Especificamente, esta norma fornece uma terminologia padrão e um conjunto coerente de conceitos e modelos para a integração de sistemas de controle com os sistemas corporativos que irão melhorar a comunicação entre todas as partes envolvidas. É o modelo mais utilizado como referência para a indústria de manufatura discreta até hoje. É uma estrutura composta de 5 níveis, do nível 0 ao nível 4.

O objetivo do ISA 95 é reduzir os riscos, custos e erros associados à implementação da interface entre estes níveis (figura 6).

Administração da Empresa

Recompresa

Administração da Empresa

Gestão da Produção

O,1,2

Processo Produtivo Controle Discreto, Contínuo e Batelada

Figura 6- Níveis do modelo ISA-95

Fonte: Elaborado pelo autor

O modelo é dividido em 5 partes: Sistemas de controle, atributos do modelo, Atividades, Objetos e transações B2M (Business to manufacturing).

A figura 6 demonstra os níveis do ISA 95 agrupados por camadas da empresa. Os níveis 0,1,2 do modelo ISA95 descrevem objetos, modelos atividades e atributos referentes ao chão de fábrica, a parte 3 refere-se as funções e atividades no nível 3 (Gestão da Produção/camada MES). O nível 4 descreve operações em termos de intercâmbio de informações entre as aplicações que exercem atividades comerciais e de produção associada com mais altos níveis de gestão como o Aplicativo ERP.

O modelo ISA 95 é a maior e principal referência ainda hoje sobre estrutura de automação, e define a fronteira entre chão de fábrica, a logística de negócio e os sistemas/processos de fabricação entre os níveis.

O modelo ISA-95 é hierárquico e portanto não atende aos requisitos de uma novo conceito de manufatura, e para se avançar em direção a uma manufatura avançada onde os processos são rapidamente mudados necessita-se de uma nova concepção do processo fabril, menos rígido e menos complexo. Faz-se necessário entender este modelo de referência industrial para que possamos ampliá-lo e iniciar uma mudança conceitual na manufatura com a introdução de serviços em direção à manufatura avançada.

2.5 - Sistemas Ciberfísicos

Os Sistemas Ciberfísicos (SCF, em inglês Cyber Physical Systems) são elementos importantes na manufatura avançada. Os SCFs são compostos por entidades físicas controladas ou monitoradas por algoritmos baseados em computador, que evoluíram a partir de dispositivos mecatrônicos com capacidades aumentadas de comunicação e conectividade (vide figura 7).

Segundo (BAHETI and GILL, 2011) o termo SCF refere-se "a uma nova geração de sistemas com capacidades computacionais e físicos integrados que podem interagir com os seres humanos através de muitas novas modalidades. A capacidade de interagir e expandir os recursos no mundo físico através de computação, comunicação e controle é um elemento essencial para desenvolvimentos tecnológicos futuros".

Sistemas de produção serão melhor interconectados a diferentes processos internos e externos, bem como para as redes de fornecedores e clientes. Sistemas de produção serão construídos em torno dos robôs, máquinas, sistemas de armazenamento e canais de abastecimento. Esses elementos podem ser ligados entre si formando sistemas ciberfísicos (COLOMBO, KARNOUSKOS et al., 2013). SCFs são o resultado da fusão dos mundos físico e virtual, que é possibilitada pela integração das capacidades de computação crescentes com os processos físicos. Essa fusão permitiu computadores embarcados, máquinas e sistemas em rede para controlar e monitorar os processos físicos (KAGERMANN, HELBIG et al., 2013).

Esta tese busca, com o uso de sistemas ciberfísicos, servir-se de

elementos do chão de fábrica com maior capacidade de processamento e comunicação, não havendo mais uma barreira entre a comunicação entre os sensores, atuadores com sistemas de camadas superiores como o MES ou ERP e com isso dar subsídios para soluções alinhadas com os conceitos da manufatura avançada.

No cenário que se vislumbra, a automação possivelmente perderá seu caráter rígido e verticalizado e haverá maior colaboração entre os componentes. Esta tese busca utilizar serviços que serão expostos por interfaces padronizadas, utilizando *Webservices*, além da tradicional funcionalidade de controle.

SCFs podem ser controlados também em tempo real. As plantas resultantes e sistemas do futuro vão ter claramente definidas interfaces semelhantes que simplificam a escalabilidade do sistema (CANNATA, GEROSA et al., 2008).

Devido a essas novas oportunidades, SCFs irão impactar no processo de fabricação devido a (vide figura 7):

- Melhorar a simulação e virtualização dos processos físicos.
- Formação de sistemas inteligentes descentralizadas com tipo de autogestão e autodecisão aumentando a capacidade de tomada de decisões autônomas (menor interferência humana).

Análise em tempo real

Análise em tempo real

Busca interativa

CONSULTAS

Ex+y

Análise preditiva

Análise em lote

Integração

Figura 7 - Etapas de um sistema voltado para a IoT

Fonte: Elaborado pelo autor baseado em (GUBBI, BUYYA et al., 2013)

2.6 - Computação nas Nuvens

Segundo (FERNANDES, 2012) "computação em nuvem é a entrega da computação como um serviço ao invés de um produto, onde recursos compartilhados, software e informações são fornecidos, permitindo o acesso através de qualquer computador, *tablet* ou dispositivo conectado à Internet".

Na computação nas nuvens tudo é tratado como serviços, onde os recursos a serem utilizados não se encontram mais fisicamente próximos ou acessíveis de forma física, devem ser alocados de forma virtual. Essa é uma tendência que parece ser inevitável, a de que cada vez mais teremos os sistemas armazenados não localmente, mas nas nuvens e poderemos utilizar a qualquer momento e através de diversos meios.

Descreve-se os componentes desse paradigma, de acordo com os elementos mostrados na figura 8.

Software as a Service (SaaS)
Alvo: Usuário/Negócio

Platform as a Service (PaaS)
Alvo: Desenvolvedor

Infrastucture as a Service (IaaS)
Alvo: Administrador

Aplicações

Middleware em Tempo real

Poder cómputacional

Figura 8- Componentes de um serviço de computação nas nuvens e a visão de tudo como serviço

Fonte: Elaborado pelo autor baseado em (XU, 2012)

Software como serviço (Software as a Service - Saas): Nesta forma de serviço o software é oferecido geralmente via Web, assim, o usuário não precisa instalar programas ou adquirir licenças de uso para instalação ou mesmo comprar computadores ou servidores para executá-lo. Nessa modalidade, normalmente paga-se um valor periódico (assinatura) pelos recursos utilizados e período de tempo de uso. Exemplo disto: Google Docs, Photoshop.com, Free Pdf Converter, Gmail, Hotmail, Salary.com, Office Live.

Plataforma como Serviço (*Platform as a Service* - Paas): neste modelo, que fica entre o SaaS e IaaS, é possível a utilização de softwares de maneira mais simples, pois permite desenvolver suas próprias aplicações baseadas em alguma tecnologia (*framework* (vide glossário), linguagem etc). A PaaS

facilita a implantação de aplicações com menor custo complexidade na compra e gestão do hardware, software e recursos de provisionamento de infraestrutura. Exemplos: Microsoft Azure, Google App Engine e SalesForce.

Infraestrutura como serviço (Infrastructure as a Service - Iaas): neste modelo contrata-se a infraestrutura como serviço. Dessa forma, ao invés de se preocupar com compra de servidores, roteadores, racks e outras "caixas" de hardware, contrata-se servidores virtuais (e outros dispositivos de infraestrutura). Neste serviço há a tarifação por alguns fatores, como o número de servidores virtuais, quantidade de dados trafegados, dados armazenados e outros itens, dependendo de como e com quem (fornecedor IaaS) se trabalha. Exemplos: Amazon EC2 e IBM.

No caso da manufatura avançada existe um misto entre infraestrutura como serviço (Iaas), que será provido pelos SCF e de software como serviço (SaaS) que pode ser promovido pelo integrador, ou pelo próprio desenvolvedor do SCF, assim como por empresas terceiras de software.

No serviço de computação nas nuvens existe uma relação composta de duas partes: o provedor da solução e o utilizador, que pode ser uma pessoa, uma empresa ou qualquer outra organização.

A depender da aplicação pode-se optar por uma nuvem pública ou nuvem privada ou híbrida, que é uma mistura das duas.

Quando os dados podem trafegar pela Internet, e para serviços para pessoas físicas, pode-se entender esse contexto como um esquema de nuvem pública. No entanto, especialmente no que diz respeito ao segmento corporativo, é possível também o uso do que se conhece como nuvem privada.

A nuvem privada (*private cloud*) oferece praticamente os mesmos beneficios da nuvem pública. A diferença está, essencialmente, nos "bastidores": os equipamentos e sistemas utilizados para constituir a nuvem ficam dentro da infraestrutura da própria corporação.

Em outras palavras, a empresa faz uso de uma nuvem particular, construída e mantida dentro de seus domínios. Mas o conceito vai mais além: a nuvem privada também considera a cultura corporativa, de forma que políticas, objetivos e outros aspectos inerentes às atividades da companhia sejam respeitados.

A necessidade de segurança e privacidade é um dos motivos que levam uma organização a adotar uma nuvem privada. Em serviços de terceiros, cláusulas contratuais e sistemas de proteção são os recursos oferecidos para evitar acesso não autorizado ou compartilhamento indevido de dados. Mesmo assim, uma empresa pode ter dados críticos por demais para permitir que outra companhia responda pela proteção e disponibilização de suas informações. Ou, então, a proteção oferecida pode simplesmente não ser suficiente. Em situações como essas é que o uso de uma nuvem privada se mostra adequado.

A área de manufatura possui aspectos que são altamente sigilosos e críticos no que se refere à segurança, por isso deve-se optar por uma nuvem privada, pois oferece a vantagem de ser "moldada" com precisão às necessidades da companhia. Isso porque o acesso à nuvem pode ser melhor controlado, assim como a disponibilização de recursos pode ser direcionada de maneira mais eficiente, aspecto capaz de impactar positivamente a rotina corporativa.

Esta tese busca aplicar o conceito de serviços da manufatura na nuvem para que se possa gerir de forma transparente e independente da tecnologia e da posição geográfica os processos produtivos. Tem-se, para aplicar tal conceito, que se avançar nos itens de cibersegurança, menor tempo de resposta e vencer a resistência dos gestores em disponibilizar dados de produção fora do ambiente empresarial (local).

2.7 - Internet das Coisas - IoT

O conceito de internet das coisas (IoT) vem sendo construído baseado em outras tecnologias e tem mudado ao longo da última década com a evolução das tecnologias, capacidade de interação e interoperabilidade entre sistemas.

Conforme visto na figura 9, existe uma evolução em direção a um sistema interação entre dispositivos. Esta evolução passa por estágios de evolução também dos seus elementos constituintes.

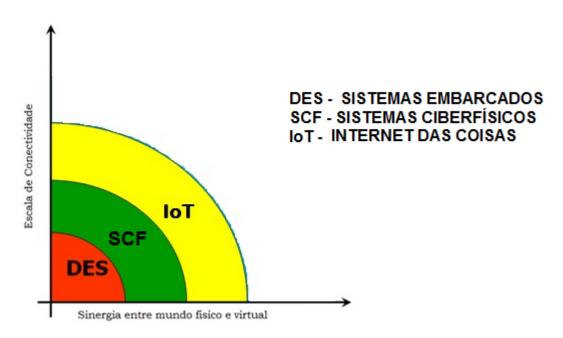


Figura 9- Níveis de evolução de sistemas inteligentes

Fonte: Elaborado pelo autor

A Internet das Coisas (IoT), segundo (GUBBI, BUYYA et al., 2013), refere-se a uma rede de objetos heterogêneos interligados que são exclusivamente endereçáveis e são baseados em protocolos de comunicação padronizados. Estes incluem sensores, atuadores, dispositivos inteligentes, computadores embarcados, dispositivos móveis, etc. A Internet das coisas

pode ser vista como uma tecnologia que permite sistemas descentralizados e distribuídos cooperarem e se conectarem entre si, ou seja objetos com grau de autonomia física ou virtual, com capacidade de detecção/atuação, processamento, armazenamento, comunicação em rede e decisão.

Neste contexto existirão objetos com menor capacidade de processamento e autonomia de bateria e outros objetos inteligentes podem atuar como agentes com algum grau de autonomia, cooperar com outros agentes, e trocar informações com usuários humanos e outros dispositivos de computação dentro de SCFs interconectados e interligados em rede e prover resultados. Conforme exposto na figura 9, a evolução da Internet das coisas busca que elementos que se conectam possam ser capazes de captar dados de seu contexto, realizar predições seja baseado em interações com humanos (via consultas), outros sistemas e dispositivos.

Essas "coisas" poderão atuar no ambiente. Para que isso se materialize, uma série de desafios de integração, análise de um volume imenso de dados, tratamento e inferência são necessários. Existem autores que conceituam esse desenvolvimento como a *Internet of Everything* (Internet de Tudo), para maiores informações vide (CHAMBERS, 2014) ou (EVANS, 2012)

O principal objetivo é fazer com que dados do dispositivo tenham sentido (ser relevante) com o mínimo ou sem intervenção humana. A radical evolução da Internet atual em uma rede de objetos interconectados colhe informações do meio ambiente (sensor) e interage com o mundo físico (atuação/comando/controle), usa padrões em conformidade com a Internet para prestar serviços para a transferência de informações, análise, aplicativos e comunicações.

Esta tese busca trabalhar com a evolução dos sistemas mecatrônicos, então temos a seguinte evolução: Sistemas mecatrônicos -> SCFs -> IoT (SCFs se comunicando, ou seja, SCFs + redes de comunicação + processamento/ inferência) -> Manufatura Avançada com elementos IoT com o propósito de produção ampliada.

2.8 - Comentários

Neste capítulo abordamos os principais temas que servirão como pedras fundamentais na proposta de Tese, é inegável que há uma crescente demanda para aprimorar o gerenciamento dos mais diversos níveis de gestão e reduzir a distância dos elementos da automação industrial, ampliando suas capacidades de monitoramento, e resposta a eventos.

Um dos gargalos dessa ampliação se tornar mais efetiva é o problema da integração, e a indústria e a academia têm investido cada vez mais esforços na abordagem orientada a serviços para maior integração entre hardware e software. SOA é atualmente uma grande aposta da indústria e academia como uma abordagem que pode ampliar essa integração. Esta tese busca contribuir para o amadurecimento tecnológico e científico com ênfase orientada a serviços em direção a essa nova geração da manufatura industrial.

No próximo capítulo, será descrito o modelo proposto. Ele reutiliza e expande trabalhos anteriores, principalmente o de Gonçalo Cândido (CÂNDIDO, 2013), que tinha como foco propor um modelo para o ciclo de vida dos elementos do chão de fábrica, e o projeto europeu IoT-A (FREMANTLE, 2014), que propôs um modelo de arquitetura com aspectos conceituais para subsidiar soluções com ênfase em Internet das coisas. A tese reutiliza avanços de trabalhos de (CÂNDIDO, JAMMES et al., 2009) que desenvolveram uma estrutura para gerenciamento de serviços para dispositivos baseados no protocolo DPWS que é baseado em SOA, (JAMMES and SMIT, 2005) foram os primeiros a propor o uso de serviços aplicados a automação industrial, descrevendo um modelo de interação entre dispositivos não mais por ciclos, mas por eventos, consideralvemente o tráfego em rede e processamento. Serão também apresentadas tecnologias existentes para o desenvolvimento baseado em serviços para a automação industrial, bem como seus desafios para a integração.

CAPÍTULO 3 - MODELO PROPOSTO E TECNOLOGIAS BASEADAS EM SERVIÇOS

Métodos e metodologias tradicionais não estão atendendo mais os requisitos do usuário (COLOMBO, BANGEMANN et al., 2014), a necessidade de uma constante inovação, que deve ser enfrentada e gerida com as mais novas e melhores práticas de engenharia e de TI. As empresas têm o desafio de adaptar o seu planejamento e sistemas de fabricação para produzir de uma forma mais integrada, flexível, reconfigurável e com melhor colaboração (MORAES, LEPIKSON et al., 2015).

A ampla oferta de tecnologias no escopo da manufatura avançada dificulta a identificação, por parte dos usuários, das formas mais eficientes para atender suas necessidades. Além disso, complexos modos de funcionamento exigem elevado grau de conhecimento para que os usuários consigam empregar as tecnologias elencadas de forma eficiente. Desta forma existe dificuldade de se implementar as soluções que buscam materializar a manufatura avançada. O modelo proposto busca avançar com uma maior interoperabilidade e integração entre diferentes níveis da produção com vista a desenvolver uma solução distribuída e baseado em serviços.

Este capítulo apresentará o modelo base para a construção da arquitetura orientada a serviços AMOS (Arquitetura Modular Orientada a Serviço) e em seguida abordará as ferramentas e tecnologias usadas para o desenvolvimento de sistemas de produção baseados em serviços.

3.1 - MODELO AMOS

Descreve-se no modelo AMOS, numa visão macro, os elementos que dele participam, como é feito o controle e monitoramento na manufatura avançada, as interações, e dos principais serviços contidos. Existe uma interligação entre o modelo proposto e a arquitetura proposta. O modelo

busca contextualizar e estruturar o conhecimento. Já a arquitetura busca subsidiar o desenvolvedor na criação das interfaces para o provimento dos serviços em um cenário da manufatura avançada e iniciando em um ambiente onde não se trabalha com orientação a serviços.

Para um melhor entendimento do modelo, o conhecimento relacionado está estruturado da seguinte forma:

- Inicialmente será definido o conceito do que é serviço e interface no escopo desta tese;
- Em seguida descreve o conceito "Conecte e Produza", e como ele se materializa em termos práticos;
- Depois descreve-se um cenário real, com exemplo em uma manufatura de automóveis onde o conceito de manufatura avançada se aplica, tendo o produto como um recipiente da informação e agente no processo;
- Segue a figura 11, com o modelo de comunicação e integração entre dispositivos e sistemas em um ambiente orientado a serviços, mostrando que sistemas ciberfísicos são parte hardware e parte software, e esta tese vem atuar na interface que interliga essas duas partes;
- Tendo compreendido os conceitos até aqui expostos, detalha-se o modelo AMOS, descrito na figura 12.
- Em seguida descreve-se modelos de dados orientados a serviços que permitem comprovar que a proposta não é voltada para uma tecnologia específica, quatro modelos são detalhados e foram utilizados nesta tese nos estudos de caso.

3.1.1 - DEFINIÇÃO DE INTERFACE E SERVIÇO

Sistemas ciberfísicos serão utilizados no modelo proposto para ofertar funcionalidades via serviços, com interfaces uniformes de dados e de rede. A integração de elementos mais complexos, que necessitam combinar e trocar

dados, é feita por mecanismos de orquestramento e composição de serviços que são disponibilizados nas nuvens.

Tendo em vista a manufatura avançada, e com o objetivo aprofundar o conhecimento em relação ao tópico sistemas de integração utilizando a abordagem orientada a serviços via interfaces faz-se necessária uma definição do entendimento desses conceitos e a figura 10 auxilia esse entendimento.

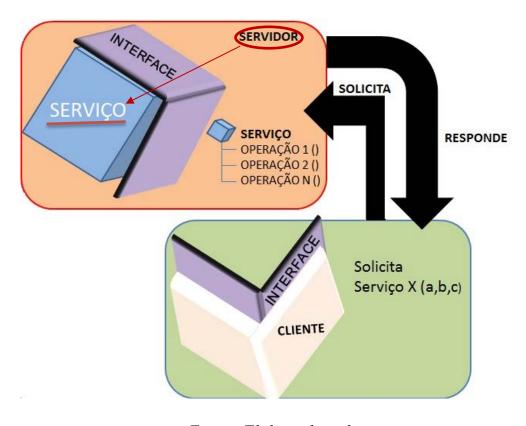


Figura 10 -Concepção de Serviço e Interface

Fonte: Elaborado pelo autor

O chão de fábrica sendo composto por robôs, sensores, bombas, inversores de frequência, válvulas, etc., serão módulos de sistemas ciberfísicos capazes de expor suas funcionalidades através de Webservices e protocolo de rede baseado em Ethernet utilizando modelo de dados orientado a serviços como OPC UA ou DPWS. Os elementos de produção podem dispor

e também consumir serviços. Com isso pode-se ter, em determinado momento, um elemento atuando como servidor e, em outro contexto, ser cliente dependendo da relação e do processo a ser realizado.

Para que isso ocorra deve haver uma colaboração entre os elementos ciberfísicos para que os serviços possam atingir seu objetivo, muitas vezes a solicitação envolve mais de um elemento de produção. A aplicação de SOA no chão de fábrica implica na composição de versão digital dos componentes, havendo uma tendência de que sejam agrupados e modularizados.

Tendo como base os problemas elencados no capítulo 1 (longo tempo para o projeto, comissionamento e configuração do sistema; Complexidade de ajustes dos sistemas industriais para atender a novos requisitos; Chão de fábrica com pouca integração com outras instâncias do sistema produtivo; Incompatibilidade entre sistemas de diferentes fornecedores e equipamentos com sistemas legados;) busca-se aplicar o conceito de Conecte e Produza (*Plug & Produce*) para que, se beneficiando das vantagens das interfaces que uniformizam a comunicação e os sistemas ciberfísicos, possa-se reduzir a complexidade e melhorar o desempenho da produção da manufatura. Para que isso ocorra é importante simplificar a configuração de um novo dispositivo, para tanto propõem-se os seguintes passos:

- Ao tentar se conectar na rede pra iniciar a produção, o sistema ciberfísico envia uma mensagem de broadcast (vide glossário) solicitando entrada;
- A mensagem do dispositivo deve ser detectada por uma interface de rede e um elemento (ou mais) de produção será responsável pelo gerenciamento de novos Sistemas Ciberfísicos na rede;
- O elemento gerenciador deve inicialmente responder ao elemento entrante confirmando a recepção da mensagem, e solicitando mais dados sobre o dispositivo;
- O dispositivo entrante deve responder com mais dados, como ID, modelo, série, versão, etc.
- O elemento gerenciador deve ter em seu banco de dados informações

sobre os dados enviados pelo dispositivo entrante para com isso detectar suas funções no processo. O elemento gerenciador deve retornar ao dispositivo entrante os arquivos de configuração, o IP a ser usado e as instruções de controle correto do processo;

- O dispositivo entrante deve receber e informar ao elemento gerenciador se foi possível atualizar os seus dados de configuração a partir da mensagem recebida e informar que está preparado para iniciar a produção;
- Ao receber a mensagem de confirmação o elemento gerenciador registra os dados do dispositivo, e envia uma mensagem liberando o mesmo a iniciar a produção.

Com as etapas acima descritas pode-se com um novo dispositivo se registrar na rede e ser autorizado a produzir, recebendo as instruções de produção, será agregado para compor determinado serviço de acordo com suas características, com isso tem-se a reconfiguração (setup de maquinário) em tempo de execução no chão de fábrica. Exemplos incluem lidar com sistemas ciberfísicos autônomos, estruturas heterogêneas que necessitam evoluir com a demanda e onde são adicionadas, removidas ou modificadas funções e objetivos ao longo do ciclo de vida.

Para que isso aconteça, uma série de abstrações e concepções no desenvolvimento da solução são necessárias. Com o modelo e arquitetura de referência proposta, considerações e módulos necessários para a materialização dessas interfaces na manufatura avançada serão detalhados.

A disseminação de SOA no chão de fábrica é facilitada pela instalação de protocolos orientados a serviços em interfaces dos equipamentos. Com isso os dispositivos têm maior facilidade de integração e maior oferta de composição de serviços.

Os tipos de serviços promovidos serão classificados, de acordo com (COLOMBO, BANGEMANN et al., 2014), em:

• Serviços atômicos: Serviços elementares que podem ser promovidos por

um único recurso.

- Serviços Compostos: Serviços que necessitam de interação de mais de um elemento de produção e são ofertados como um serviço único ao cliente.
- Serviços de Configuração: Serviços de atualização, inserção de novo elemento de produção, *download* de novas regras de produção, e que não retornam dados ao cliente, ficando a cargo dos gestores da produção.

Definido o serviço, é importante salientar o conceito de interface utilizado nesta tese. Interfaces são camadas que norteiam o registro, descoberta, provimento e o gerenciamento dos serviços. As interfaces abstraem a complexidade da rede e do modelo de dados, e permitem que o cliente e o servidor possam se comunicar satisfatoriamente, e que ocorra a troca de dados. A troca de dados pode ser síncrona ou assíncrona. Diferentemente do modelo atual da automação industrial, ao se implementar serviços nos dispositivos se introduz o conceito de eventos, onde não há tempo de ciclo pré-determinado, ficando a cargo da necessidade do cliente. As interfaces auxiliam a composição de serviços e reuso de componentes já desenvolvidos.

3.1.2 - CENÁRIO DE APLICAÇÃO

Um cenário prático é a fabricação de vários automóveis com várias linhas de produção, onde robôs podem atuar em mais de uma linha de produção, onde maquinário e insumos "conversam" sobre as operações industriais e o processo de fabricação, colaborando entre si para buscar qual disponibilidade de receber as matérias-primas. Imagine que recebem o motor de um fornecedor, o vidro de outro, suspensão de outro, mas durante o processo de montagem cada peça possui uma tag RFID com todas as informações sobre onde, quando, quais testes foram feitos pelo fornecedor e os resultados armazenados no produto que chega à montadora. Esses dados de diferentes fontes, inclusive de diferentes unidades industriais ou mesmo empresas diferentes serão trocados, defeitos encontrados e registrados (controle de qualidade) e alertados os responsáveis (interno ou de fornecedores), que devem garantir o produto dentro dos parâmetros

contratados. Outros testes serão feitos e adequações necessárias para o produto final. Estas informações estarão sendo levadas junto a peça durante todo o processo e retroalimentada automaticamente por outras máquinas que poderão ler e trocar informações entre si.

Além disso, a cada momento o sistema, de forma autônoma, analisa qual a linha de produção está desocupada e realoca para os recursos que otimizam a produção. Com esse processo muitos dados são gerados, dados estes que devem ser sumarizados por técnicas de mineração de dados, algoritmos inteligentes, armazenados em *datawarehouses*, e disponibilizados para a gerência com o uso de BI e *big data*, sendo refletidos em indicadores para a tomada de decisão. Após a fabricação há a integração entre sistemas da cadeia produtiva que permitem uma otimização logística e ademais agregam valor ao processo, pois tais informações otimizam outras áreas como: design, projeto, pós-venda e descarte.

Importante salientar que o modelo e a arquitetura propostos nesta tese limitam-se ao chão de fábrica, e tem como ênfase na integração via serviços na produção, auxiliando a "conversa" entre os elementos de produção, o estabelecimento de serviços, porém não contempla o gerenciamento da cadeia produtiva, logística e camadas superiores que envolvem sistemas corporativos.

Uma situação como esta descrita permitiria um controle em tempo real com maior quantidade de informações, onde os dispositivos e sistemas colaboram para o objetivo do negócio, e propicia uma tomada de decisão da gestão mais assertiva, buscando os beneficios advindos da facilidade de integração, publicação, recuperação de informação.

3.1.3- ESTRUTURA DE COMUNICAÇÃO ORIENTADA A SERVIÇO ENTRE SISTEMAS CIBERFÍSICOS

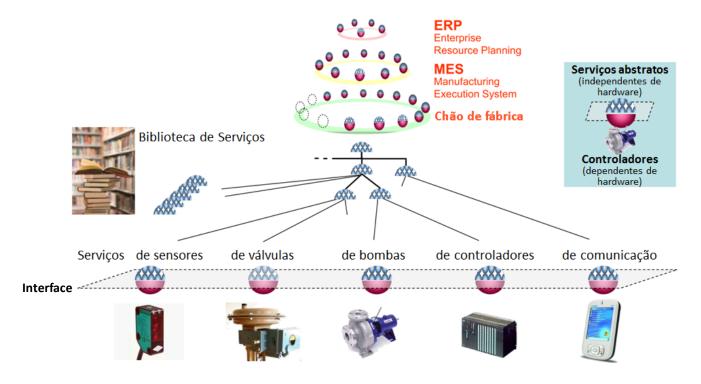
A figura 12 demonstra o modelo de comunicação entre sistemas e dispositivos do chão de fábrica que são sistemas ciberfísicos, compostos por parte software e parte hardware, havendo a exposição e o encapsulamento de funcionalidades do hardware via serviços, através requisições de software sob demanda e se comunicando baseado em modelo de dados orientados a serviços (será descrito com maiores detalhes exemplos de modelos de dados no decorrer do capítulo). O hardware será agregado com uma nova camada de software que permitirá que ele se comunique via serviços, serviços estes que podem ser alterados de acordo com o planejamento e necessidade da gestão, os serviços agregados são abstratos e podem ser independente de hardware.

Esta tese vem atuar entre a camada de hardware e a camada de software, promovendo interfaces na busca de romper a barreira hoje existente com a comunicação hierárquica e verticalizada da automação industrial tradicional, transformando um dispositivo sem capacidades SOA para que trabalhe com interoperabilidade, autonomia e de forma aberta, que são princípios do conceito de serviço.

Isso tem impacto em como a manufatura atua e tem um impacto importante sobre treinamento de pessoal, infraestrutura, modelo de negócios, em busca de concretizar conceitos da manufatura avançada.

A tese propõe um modelo e uma diretriz para o desenvolvimento de interfaces orientadas a serviços. Como descrito na figura 11, uma interface flexível é o que vai possibilitar a ruptura da estrutura com alta coesão e hierárquica, permitindo que diferentes serviços(que são implementados em software) possam ser alocados a determinado hardware, onde o hardware pode se comunicar com diversos outros sistemas ciberfísicos ou sistemas empresariais.

Figura 11 – Modelo de comunicação entre dispositivos e sistemas em um ambiente orientado a serviços e interface entre hardware e software



Fonte: Elaborado pelo autor, adaptado de (Lepikson, 2016)

No cenário proposto, a integração não é mais vertical ou horizontal, mas sim colaborativa e feita ao longo do ciclo de vida entre diversos elementos de produção, insumos, maquinários, computadores de processo que têm como objetivo monitorar, controlar, rastrear todas as etapas de produção, com a gravação de dados em tempo real em cada fase de modo automatizado.

Todos os elementos elencados oferecem suas funcionalidades através de serviços, podendo atuar como cliente ou servidor a depender do processo e do contexto, serviços esses que são descritos no modelo de dados, que interligam o chão de fábrica com os sistemas corporativos através de interfaces comuns. Com isso, diferentes padrões de interação entre um consumidor e servidor de serviço podem acontecer:

• a mensagem de sentido único, quando o remetente não se preocupa com a mensagem e a operação não tem um valor de retorno;

- a solicitação-resposta, que formam pares de RPC (chamadas remotas de procedimento) seja diretamente (solicitar-resposta) ou inversamente;
- as notificações, que permitem que o cliente seja informado quando algo acontece no lado do serviço;

Descreve-se com maiores detalhes desse modelo de dados ao final deste capítulo. Para maiores detalhes sobre essa integração vide capítulo 2.2.

3.1.4 - DETALHAMENTO DO MODELO AMOS

De forma a condensar e utilizar todo o conhecimento descrito neste capítulo, em busca da implantação da manufatura avançada e orientada a serviços, propõem-se um modelo de integração (figura 12) que descreve os principais componentes e suas funcionalidades. O modelo descreve com uma visão macro os elementos que participam, como é feito o controle e monitoramento na manufatura avançada, as interações, e dos principais serviços. Existe uma interligação entre o modelo e a arquitetura proposta, onde os componentes do modelo são utilizados na arquitetura. Na arquitetura, a ser discutida no próximo capítulo, descreve-se os componentes da nuvem de acordo com a experiência, grau de maturidade, do maquinário da manufatura, de uma forma modular e incremental.

O modelo foi concebido com o objetivo para preparar o desenvolvedor a implementar a arquitetura, de forma a mostrar como será a coexistência entre sistemas legados e SCFs. No modelo os elementos do chão de fábrica podem se comunicar via serviços com as camadas superiores da gestão e com uma visão de computação nas nuvens. Há uma tendência forte na virtualização dos processos, e com processos e elementos virtualizados é simplificada a transição para a computação nas nuvens. Hoje em dia com a nova geração da automação tem-se a inserção de novos sistemas de informação e comunicação, associada a SCFs inteligentes que estão mudando as empresas e, especialmente, o setor industrial (LEITÃO, MARIK et al., 2013).

SCF App Base de Sistemas Móvel dados Corporativos Externa MONITORAMENTO/CONTROLE EM TEMPO REAI Orquestramento MES Predição Colaboração GERENCIAMENTO Inteligência Auto-Computacional Base capacidades de Repositório dados Segurança de Serviços Interna dos dados Análise de Contexto SERVIÇOS EM REDE Servicos SCADA dinâmicos Dispositivos SCF SCF SCF Legados

Figura 12 - Modelo de Integração AMOS

Fonte: Elaborado pelo Autor

No modelo proposto baseado em serviços, os sistemas legados (mais antigos) que não puderem trabalhar diretamente via serviços continuarão sendo monitorados por sistemas SCADA e devem ser inseridos no modelo através de um *gateway*. Este detalhamento será aprofundado no capítulo seguinte durante a descrição da arquitetura, juntamente com a explicação sobre os serviços na nuvem que ligam os serviços orquestrados em rede com o ambiente de gerenciamento externo.

Os serviços serão disponibilizados através de componentes, distribuídos via rede. Descreve-se conceitos do modelo proposto em seguida.

a) Sistemas de Controle:

A forma de controle atual se manterá e será adicionada camadas de controle que devem combinar sistemas baseados em tempo com sistemas baseados em eventos para controle de feedback. Sistemas críticos necessitam de tempo de resposta em milissegundos, e diversas possíveis falhas em equipamentos eletrônicos, ou queda de energia poderiam comprometer a planta e necessitam de pronta resposta e atuação no processo. A área de controle em sistemas ciberfísicos ainda carece de amadurecimento, sendo uma área aberta de pesquisa, onde gargalos de tempo de resposta, melhoria de performance e sua inserção em loops críticos A figura 13 descreve em maiores detalhes as não possuem respostas. camadas de controle. As camadas externas são as que podem ser acionadas primeiramente. Em uma manufatura avançada, existem dispositivos que atuam com capacidade de autoorganização, automanutenção, diagnóstico remoto e interações autônomas, sendo uma camada mais interna desenvolvida via software. Aliada ao controle tradicional, direcionamente vindo de camadas superiores de sistemas corporativos serão possíveis e coexistirão.

Interação Homem-Máquina Interação entre sistemas

SOFTWARE

Hardware eletrônico

Sensores e atuadores

Figura 13 - Camadas de Controle

Fonte: Elaborado pelo autor, baseado em (MONOSTORI, 2014)

Físico / Mecânico

O Monitoramento possui as seguintes características:

- Em um ambiente de manufatura avançada que é naturalmente distribuído, onde a estrutura atual de monitoramento é utilizada e ampliada com os alertas e os algoritmos provenientes da Inteligência computacional, e inferências preditivas.
- Este monitoramento deve aplicar-se a hierarquias envolvendo dinâmica assíncrona em diferentes escalas de tempo (de meses a microssegundos) e escopo geográfico (desde elementos no mesmo barramento até em escala planetária).

O componente **Orquestramento** possui as seguintes características:

 Será responsável pela coordenação e gestão (semi-) automatizada de complexos sistemas de produção. Entende-se semi-automatizado pois, algumas ações são realizadas de forma automática de acordo com o fluxo do processo e outras ações necessitam intervenção manual. As escolhas dependerão do tipo de processo, custos envolvidos e segurança dos operadores.

- Além disso, será responsável por permitir uma visão baseada em tarefas em termos dos processos de fabricação para promover a (re)configuração do sistema, atuando como "maestro" da manufatura, ordenando atividades, promovendo serviços compostos e informando dados para que novos elementos ao se conectarem possam interagir e contribuir com o processo fabril. A (re)configuração do sistema são os ajustes necessários, setups de maquinário, alocação de matéria prima, de ferramentas para que a linha de produção não pare. Busca-se com isso minimizar o tempo de setup e reduzindo o seu tempo ocioso e aumentar a produtividade. Além de buscar facilitar a interferência humana e minimizar retrabalhos e/ou ajustes manuais.
- Garante o baixo acoplamento (vide glossário) via troca de mensagens entre os componentes para promover a composição simples de serviços para permitir uma (re)configuração mais fácil. O orquestramento está preocupado com a execução orientada para o fluxo de trabalho e a sequência de processos unitários. Na prática, o sequenciamento e sincronização da execução dos serviços, é o que encapsula os objetivos da empresa. Um mecanismo de orquestramento implementa a lógica necessária para orquestrar serviços unitários da aplicação.
 - O componente **Autocapacidades** reúne características que são importantes na manufatura avançada, esta lista características não é fechada, podendo em cada contexto ser adaptada:
 - Autodetecção: capacidade do SCF ser descoberto sem maiores configurações para materializar o conceito conecte e produza acima descrito.
- Autoconfiguração: Cada dispositivo tem arquivos de configuração e controle a serem executados também disponibilizados no repositório.
 Assim que é ligado o SCF inicia o processo de autoconfiguração,

buscando esses arquivos no repositório seguindo as regras ali definidas.

- Transparência: A transparência envolve a capacidade de o sistema reorganizar a realização de determinada atividade quando um dispositivo apresenta falha. O orquestrador tentar a alocação de determinada tarefa a determinado SCF e não recebendo resposta, o orquestrador enviar uma solicitação de serviço para um outro SCF com o serviço semelhante, se receber a resposta de que ele está em funcionamento, livre e ativo, então é alocado sem a interferência do usuário.
- Autoreparo: Esta é a capacidade de o SCF realizar testes e detectar alguma não conformidade. Inicialmente uma reinicialização será realizada e em seguida deverá ser realizado o download dos arquivos de configuração que estão salvos no repositório, atualizando os serviços e configurações do dispositivo. Caso não se resolva, um alarme deve ser disparado para o orquestrador e para a um setor responsável (manutenção).

Componente **Repositório de Serviços**: Este componente pode ser visto como as páginas amarelas dos serviços. Ele realiza uma busca no repositório de serviços e retorna as informações disponíveis pela interface.

Ex: Nome do dispositivo, status do dispositivo (online ou offline), localização geográfica, etc. Um repositório de serviço pode ser tanto de base local ou disponíveis on-line como um portal de negócios web. O repositório on-line pode ser atualizado com os recursos apresentados pelos desenvolvedores de terceiros e disponibilizados on-line para a comunidade que quer empregálos em seus próprios sistemas. Esses recursos podem incluir funcionalidades específicas para diversos domínios, recursos avançados de configuração ou acesso à informação, em

particular implementações de equipamentos ou gateways de comunicação. Neste componente os dispositivos se subscrevem para poderem ser localizados e fornecer ou usar serviços. A subscrição é a primeira etapa que um elemento deve fazer ao buscar interagir com o processo fabril, seria como o anúncio nas páginas amarelas. Nesta etapa inicial o elemento solicita o cadastramento, e se autorizado, inicia-se uma troca de mensagens de acordo com o modelo de dados escolhido, a depender da complexidade do processo e do número de serviços que o dispositivo irá fornecer. Após a subscrição, o elemento estará credenciado a oferecer e solicitar serviços.

Componente **Serviços sensíveis ao contexto**: Com uma maior quantidade de informações em tempo real, a aplicação poderá se adaptar, detectar e realizar ações automaticamente, sem a necessidade de interação direta com o usuário. Para o acesso a estas informações, uma interface deve ser utilizada, implementando conceitos de sensibilidade ao contexto aplicado à automação industrial. Para que isso se materialize dois aspectos são importantes:

Aspecto 1: Serviços de contexto aplicado a diferentes visões de usuários, por exemplo, as diferentes perspectivas e visões necessárias de um mesmo sistema quando observadas por um operador, pela manutenção, pelo controle da produção, pela gerência da produção, pelo setor de vendas, entre outros. Um mesmo dado obtido de uma máquina vai servir a diferentes contextos. Por exemplo, o dado "OS no 3895 65% executada" orienta o operador sobre próximos passos a seguir, programação da produção sobre a próxima OS, o sistema CAM sobre os próximos programas CNC e CLP a executar, o gerente de vendas sobre a situação de entrega do pedido ao cliente. O componente deve prover informações na granularidade específica e adequada para casa usuário, criando a visão adequada ao contexto.

Outro aspecto importante é a adequação da plataforma de software e hardware do usuário, detectar a quantidade de dados, processamento suportado pelo dispositivo, e adequar ao melhor protocolo para seu reconhecimento, comunicação e controle, além de gerenciar variáveis em seu contexto. Imagine ambientes onde o acesso deve ocorrer por meio de cabo USB ou fibra ótica, cabo par-trançado, redes sem fio, como Wi-fi, telefonia móvel, aparelhos com diferentes sistemas operacionais e notebooks com os sistemas operacionais Windows e Linux. É claro que para cada dispositivo ou meio de comunicação teremos um tipo de resposta do serviço, e diferentes possibilidades de processamento de dados. O desenvolvedor deve estar atento e adequar-se tornando essas particularidades e complexidades transparentes ao usuário.

Componente **Predição**: O componente predição é associado ao componente inteligência computacional e pode realizar prognósticos e analisar não somente a base de dados internos, mas utiliza dados referentes a desempenhos de dispositivos semelhantes buscando similaridades e funcionamentos abaixo dos padrões estabelecidos, gerando alerta para a gestão. Além disso, busca inferir comportamentos futuros, buscando-se obter o ponto ótimo para realizar alguma ação preventiva e para tal informará ao responsável técnico ou gestor responsável. Busca-se com esse elemento implantar uma análise preditiva mais eficaz na detecção de falhas, fazendo com que as manutenções por quebra ou as paradas não planejadas sejam reduzidas, com mais informação pode-se planejar e executar melhores ações.

Componente **Inteligência Computacional**: um componente tem forte ligação com o componente predição, e será responsável por ler, analisar e inferir principais dados e informações do processo fabril, utilizando algoritmos matemáticos e com o apoio de inteligência artificial ele mostrará

notificações automaticamente sempre que existirem padrões definidos, ou poderá inserir algoritmos de aprendizagem para que com auxílio humano possa acontecer uma análise de melhor qualidade e assertiva. Este componente tem como missão analisar a base de dados interna, e realizará inferências buscando promover maior autonomia e com maior capacidade de raciocínio, se beneficiando da maior capacidade de processamento dos dispositivos. Pode utilizar algoritmos de Big Data e de mineração de dados.

Componente **Segurança da informação**: Em um sistema de manufatura que pode ser acessado em tempo real, este quesito tem alta relevância, e deve ser garantido para um funcionamento confiável. A estratégia de segurança pode estar associada ao modelo de dados, pois alguns já possuem criptografia dos dados (ex. OPC UA), ou pelo sistema de controle com regras de autenticidade, disponibilidade e integralidade. Este componente pode ser vigilante e possuir identificação proativa de invasões e ataques cibernéticos, com plano de ação previsto para um um sistema integrado com diferentes fontes heterogêneas.

Componente **Colaboração**: Este componente descreve regras para que os elementos trabalhem em conjunto buscando o objetivo comum, armazena-se dados no banco de dados interno, alimenta-se os componentes de inteligência computacional e o componente predição. Por exemplo em uma situação de falha o módulo colaboração é acionado e pode conter regras de substituir serviços por outros, paralisar operação e/ou acionar elementos ou pessoas externas.

Esses componentes contribuem para que o conceito de manufatura avançada aconteça, abstraindo a complexidade do sistema fabril, modificando a interação entre os dispositivos, facilitando adaptações e mudanças de setup e permitindo uma menor interferência humana. Para tal uma base de dados interna será utilizada e será alimentada por todos os componentes na nuvem com dados: por exemplo dados históricos, tags, endereços IP, últimas revisões sobre dispositivos críticos para a operação. A base externa poderá conter dados sobre falhas, procedimentos de

manutenção, paradas, documentação, dados dos responsáveis, dados sobre configuração, ou outras informações consideradas relevantes.

A próxima seção lista alguns modelos de dados considerados mais relevantes e que foram avaliados ou testados ao longo deste trabalho para materializar a troca de mensagens através do conceito de serviços.

3.2 - Webservices

O modelo proposto é baseado na interação através de serviços, e cabe esclarecer o conceito de *WebService* ou em português Serviço na rede: tratase de um conjunto de protocolos e padrões que servem para trocar dados entre aplicações de forma distribuída. Essa tecnologia é utilizada para integrar sistemas, sendo empregada principalmente em ambientes heterogêneos (vide Glossário). *Webservices* foram utilizados para desenvolver estudos de caso, conforme descrito no capítulo 5.

Podemos desenvolver aplicações conforme o modelo proposto, com softwares ou componentes capazes de interagir, com outros softwares recebendo ou enviando, independente da linguagem de programação, do sistema operacional ou hardware utilizado. Existem atualmente dois padrões principais para desenvolvimento de webservices: SOAP e REST.

SOAP (Simple Object Access Protocol, em português, Protocolo Simples de Acesso a Objetos) é um protocolo para troca de informações estruturadas em uma plataforma descentralizada e distribuída. Ele se baseia na Linguagem de Marcação Extensível (XML) para seu formato de mensagem, e normalmente baseia-se em outros protocolos da Camada de aplicação, conhecida como Chamada de Procedimento Remoto (RPC) e Protocolo de Transferência de Hipertexto (HTTP), para negociação e transmissão de mensagens, sendo regulado pelo World Wide Web Consortium (W3C).

SOAP pode formar a camada base de uma pilha de protocolos de *web* services, fornecendo um *framework* de mensagens básico sob o qual os

serviços web podem ser construídos. Este protocolo baseado em XML consiste de três partes (vide figura14):

- um envelope, que define o que está na mensagem e como processá-la;
- um conjunto de regras codificadas para expressar instâncias dos tipos de dados definidos na aplicação;
- uma convenção para representar chamadas de procedimentos e respostas.



Figura 14 - Fluxo de comunicação via SOAP

Fonte: Elaborado pelo autor baseado em (NASCIMENTO, 2012)

Como descrito na figura 15, para fazermos uma requisição ou consulta via SOAP é necessário que primeiramente haja serviços cadastrados no

PROVEDOR DE *WEBSERVICES*, que é um componente que corresponde a um servidor de aplicações.

Esses serviços são descritos e podem ser consultados pelo cliente que interagem na sequência descrita na figura. Descreve-se os componentes da figura abaixo:

- WSDL (Descreve o WebService) É um arquivo XML, que descreve detalhadamente um webservice, especificando suas operações e fomatos de entrada e saída de cada operação.
- UDDI (Registra, Publica e Descobre Serviços do WS) É um mecanismo que armazena arquivos WSDL e fornece meios para que os webservices sejam registrados e publicados, permitindo que os serviços sejam pesquisados e localizados.
- CLIENTE (Software consumidor de Serviços) É um software que consome as operações do *webservice* e efetua solicitações e recebe resultados do webservice via XML.

3.3 - REST

REST (Representational State Transfer, transferência de estados representacional) é outro um protocolo de comunicação, baseado no protocolo de HTTP. Porém ele não impõe restrições ao formato da mensagem, apenas no comportamento dos componentes envolvidos. REST é mais fácil desenvolver, mais leve em comparação ao SOAP devido à inexistência de camada de mensagens, tem uma pequena curva de aprendizado, é mais independente de ferramentas e suporta diferentes formatos de dados além de XML.

REST não é um protocolo exclusivamente para sistemas de automação industrial como outros que veremos logo abaixo, sendo oriundo das aplicações de TI. REST foi utilizado para o desenvolvimento do estudo de caso 02, a serapresentado no capítulo 5, onde foi necessário controlar um robô via um dispositivo móvel de forma remota.

Flexibilidade é a maior vantagem do protocolo REST, pois um desenvolvedor pode optar pelo formato mais adequado para as mensagens do sistema de acordo com sua necessidade específica. Os formatos mais utilizados são JSON, XML e texto puro, mas em qualquer formato pode ser usado.

Isso nos leva a outra vantagem: quase sempre Webservices que usam REST são mais leves (estrutura dos pacotes de dados com menor tamanho) e, portanto, mais rápidos. Utiliza os próprios códigos e comandos do protocolo HTTP (PUT, GET, POST). Esses comandos são associados a recursos. Um recurso é qualquer coisa que pode ser identificado por um URI (Identificador Universal de Recurso).

O problema com o REST é que ele não mantém o *status* da transação, sendo necessário para cada mensagem realizar transações independentes, não aceitando blocos de dados. Não se utiliza o WSDL para padronizar métodos e interfaces. O corpo de dados fica totalmente a cargo do desenvolvedor, não tendo padronização e com isso problemas de interoperabilidade são mais comuns (PIAZZA, 2007).

Os modelos de dados a seguir são os mais citados na academia e indústria como base para implementar transferência de dados orientada a serviço.

3.4 - OPC UA

A primeira especificação OPC foi desenvolvida em 1996 por uma forçatarefa entre diversos fabricantes industriais na área de automação industrial para ser uma interface para *drivers* de comunicação, permitindo uma leitura unificada e acesso e escrita a dados existentes nos dispositivos de automação.

OPC significava inicialmente OLE for Process Control (Object Linking and Embedding for Process Control, (FORCE, 1998), mas em 2011 a OPC

Foundation mudou seu significado. OPC agora significa *Open Platform Communications*.

A tecnologia foi baseada em tecnologias Microsoft como DCOM, COM e OLE e funcionava somente com sistemas operacionais desenvolvidos pela Microsoft, o que era uma limitação para usuários de plataformas abertas. O modelo de dados OPC UA foi utilizado no estudo de caso 04, que envolve um controlador Siemens S300, em uma planta educacional que deverá ser monitorada em uma grande distância geográfica, em outro país.

As principais aplicações na área de automação industrial envolviam sistemas SCADA e em Interfaces entre Homem e máquina (IHM) a partir de vários tipos de dispositivos de automação e hardware de diferentes fornecedores, usando uma interface unificada. Posteriormente foram desenvolvidos padrões como o OPC *Alarm* & OPC *Historical Data Access* (OPC DA), que foram projetados para acessar e tratar eventos e os dados históricos fornecidos pelos sistemas SCADA e IHM.

OPC obteve adoção bem-sucedida em um grande número de produtos, e até hoje é usado como uma interface unificada independente de fornecedores de hardware e software, para sistemas de automação em vários níveis da pirâmide de automação. Há muitas mais outras áreas onde os fabricantes gostariam de utilizar este tipo de padrão, mas não podiam usá-lo porque o OPC dependia da tecnologia COM (proprietária da Microsoft) ou por causa dos limites de acesso remoto com o uso de DCOM (MAHNKE, 2010).

Para resolver a limitação acima exposta, a OPC Foundation criou o OPC XML-DA como a primeira abordagem para manter as características de sucesso de OPC com a possibilidade de utilizar uma infraestrutura de comunicação neutra entre fornecedores de diferentes plataformas.

Uma série de razões levaram a não adoção plena da versão XML-DA que estava em desenvolvimento usando serviços da *Web*. Uma das razões foi a especificação XML-DA ter fraco desempenho com *Webservices* em comparação com a tecnologia COM, outra razão importante foi a necessidade

de uso de várias pilhas de Serviços XML Web, o que causou problemas de interoperabilidade.

Com isso o OPC UA (*Unified Architecture*) surgiu então com o objetivo do substituir todas as especificações existentes baseados na tecnologia COM e ainda poder trabalhar com serviços sem perder quaisquer funcionalidades ou desempenho e possui características descritas abaixo:

Comunicação entre sistemas distribuídos:

- Confiabilidade pela robustez e tolerância a falhas;
- Plataforma independente;
- Escalabilidade;
- Interoperabilidade;
- Implementação de Segurança e criptografia;

Característica do Modelo de dados:

- Orientado a objeto;
- Possui tratamento a dados complexos e métodos;
- Modelo comum a todos dispositivos;
- Garante a comunicação com sistema legados via OPC tradicional;

OPC UA é dividido em 13 especificações, sendo padronizado oficialmente na norma IEC 62541, para maiores detalhes consultar (MAHNKE, 2010)..

O mais importante para a concepção e desenvolvimento um sistema de automação com OPC UA são as especificações: 3, 4, 5 e 6 da norma:

- Parte 3 Endereço Espaço Modelo. Esta parte define os blocos de construção e elementos para a criação do espaço de endereço do servidor OPC UA.
- Parte 4 Serviços. Esta especificação representa possíveis interações entre aplicações cliente-servidor OPC UA. Também fornece expectativas de comportamento do servidor e cliente e tipos de dados comuns que são usados por parâmetros de serviço.
- Parte 5 Modelo de Informação. Esta especificação descreve a forma como o espaço de endereço de OPC UA, nos, e referências são utilizadas.
- Parte 6 Mapeamento de Serviço. Esta especificação fornece uma descrição de como a informação e os dados são transferidos entre servidores e clientes OPC UA.

O OPC-UA não substitui padrões OPC-DA existente, mas amplia sua funcionalidade com uma camada de interoperabilidade universal. O OPC UA foi criado para poder interagir e manter uma camada de interoperabilidade com o OPC Classic.

Além disso, OPC UA deve atender a todos os requisitos para as interfaces do sistema independente de plataforma, deve ter recursos de modelagem ricos e extensíveis para ser capaz de descrever sistemas complexos.

3.5 - **DPWS**

DPWS (Device Protocol Web Services) é um modelo de dados que define os requisitos mínimos que permitem mensagens usando Webservices de forma protegida onde seja implementado a descoberta, descrição e oferta em recursos dispositivos com limitada capacidades de processamento. Esse padrão foi integrado pela Microsoft desde o sistema operacional Windows Vista. DPWS é construído sobre o núcleo em padrões de Webservices (WS-

descoberta, WS-Addressing, WS-Eventos, etc.), e tem duas versões DPWS 1.1 e DPWS 2006 (vide figura 15). Esse modelo de dados orientado a serviços foi utilizado para o desenvolvimento do estudo de caso 03, que é uma planta de manufatura flexível na qual vários conceitos de automação diferentes são implementados.

Protocolo de Aplicação Específico

WS- Descoberta WS- Eventos WS-MetadataExchange/WS-Transfer

WS- Politicas , WS- Endereçamento

SOAP-over-UDP, SOAP, WSDL, XML Schema

HTTP
TCP

IPv4 / IPv6 / IP Multicast

Figura 15- Estrutura do protocolo DPWS

Fonte: Elaborado pelo autor adaptado de (CÂNDIDO, JAMMES et al., 2009))

A tecnologia DPWS define:

- Os serviços de hospedagem: Diretamente relacionada com o dispositivo, o que é importante para o processo de descoberta.
- Os serviços hospedados: Serviços funcionais que dependem do dispositivo que hospeda para a descoberta.
- Discovery Services: permite que dispositivos de rede para fazer propaganda de si mesmos e descobrir outros dispositivos.
- Publicação / Assinatura serviços *eventing*: permitir a assinatura de mensagens assíncronas transmitidos por outros serviços.

O objetivo do protocolo DPWS é oferecer:

- Um protocolo de descoberta dinâmica leve para localizar serviços Web que compõe com outras especificações de serviços Web.
- A ligação de SOAP para UDP (*User Datagram Protocol*), incluindo padrões de mensagem, requisitos de endereçamento, e considerações de segurança.
- Um perfil de protocolos de serviços da Web que consistem em um conjunto mínimo de restrições de implementação para permitir mensagens seguras serviço Web, descoberta, descrição e eventing em de recursos limitados.

3.6 - **GSOAP**

gSOAP é resultado da tese de doutorado de Robert A. van Engelen, que atualmente é professor da Universidade de Flórida (VAN ENGELEN and GALLIVAN, 2002). O modelo de dados SOAP foi utilizado no estudo de caso 01, onde havia um conjunto de leitores e tags RFID que não posuíam a capacidade de se comunicar via serviços. Para interconecta-los foi desenvolvido um *gateway* com gSOAP em um Raspberry que permitiu a comunicação entre os elementos via mensagens e de forma assíncrona.

O projeto gSOAP inclui tradutores de código-fonte para facilitar o desenvolvimento de Webservices e o desenvolvimento de aplicativos habilitados para XML pela geração automática de código. As ferramentas utilizam algoritmos avançados e técnicas de modelagem para o estabelecimento de um mapeamento entre o esquema XML e programação de sistemas do tipo língua para facilitar as ligações de dados completos e som. O kit de ferramentas gSOAP é usado pelas cem entre as 500 maiores empresas mundiais, incluindo as 15 melhores empresas de tecnologia.

gSOAP é uma Web toolkit aberta de desenvolvimento rápido orientado a serviços que facilita o desenvolvimento de serviços Web em C / C ++, oferecendo tanto XML para C / C ++, bem como C / C ++ para ligações de linguagem XML. Ele suporta muitas plataformas.

gSOAP automatiza totalmente o processo de serialização XML. Ao utilizar essas ligações, a interoperabilidade SOAP / XML é conseguida com uma API simples aliviando o usuário do fardo de se preocupar com detalhes sobre WSDL e SOAP, assim permitindo se concentrar na lógica essencial da aplicação e, portanto, de ser libertado de se preocupar sobre os internos da funcionalidade da comunicação via *Webservices*.

O kit de ferramentas gSOAP roda na maioria dos sistemas como Linux, BSD Unix, HPUX, Solaris, Irix, AIX, Mac OS X, Cygwin, e Windows. Modificações customizadas foram incorporadas usando a compilação condicional #ifdef para apoiar construções especificamente em sistemas embarcados.

Nesta tese foram desenvolvidos estudos de caso com as diferentes tecnologias apresentadas neste capítulo, provando que o modelo proposto é independente de plataforma e do modelo de dados escolhido, buscou-se analisar nível de complexidade de desenvolvimento, estágio de maturidade, linguagem de programação envolvida, documentação disponível e promovendo subsídios para futuros trabalhos na área.

No próximo capítulo uma arquitetura modular baseada no modelo AMOS é proposto, e nela se descreverá diferentes estágios da empresa com o conceito de serviços e uma metodologia referência descrevendo sobre etapas de como o modelo poderá ser implementado.

Capítulo 4 - ARQUITETURA MODULAR PARA DESENVOLVIMENTO DE SOLUÇÕES ORIENTADAS A SERVIÇOS - AMOS

O capítulo anterior descreveu o modelo de comunicação e o modelo de integração, onde buscou-se ter uma visão macro dos elementos que participam, de como é feito o controle e monitoramento e dos principais serviços. A arquitetura busca nortear o desenvolvedor de soluções orientadas a serviços aplicada a manufatura com a forma e os componentes já descritos no modelo, mas que serão detalhados e especificados em que estágio são utilizados.

Este capítulo detalha a AMOS (Arquitetura Modular Orientada a Serviço) que é uma arquitetura orientada a serviços com uma abordagem modularizada para o desenvolvimento de interfaces (vide glossário) no chão de fábrica buscando materializar conceitos da manufatura avançada. Buscou-se desenvolver serviços e avançar no conceito Conecte e Produza ("Plug & Produce"), com isso progredir em soluções para os problemas elencados no capítulo 1 desta tese, aumentando a flexibilidade produtiva facilitando a (re)configuração e integrando tecnologias de forma eficiente.

Como descrito no capítulo anterior, tem-se recentemente uma diversidade de modelos de dados que permite desenvolver sistemas orientados a serviços, porém o modelo de dados sozinho não garante ganhos no desempenho. Com o uso do modelo de dados orientado a serviços existem ainda diversas demandas a serem solucionadas, como o orquestramento dos serviços, a segurança, o processamento dos eventos, a descoberta e registro de serviços e dispositivos, etc., para se realizar o conceito de conecte e produza. A arquitetura AMOS busca prover princípios e uma diretriz para nortear o desenvolvimento de soluções orientadas a serviço com uma visão incremental de acordo com a complexidade do processo e maturidade da manufatura, com serviços contendo componentes que podem ser

implementados de acordo com a familiaridade da indústria com as tecnologias envolvidas e de acordo com as características específicas que podem ser utilizadas na solução no caso concreto.

Esta tese propõe a AMOS para a manufatura trabalhar orientada a serviços e que tem como um dos alicerces a produção nas nuvens, pois o chão de fábrica é o elo que materializa o planejamento gerencial de todos os níveis e deverá se adaptar aos sistemas computacionais distribuídos e muitas vezes nas nuvens, e essa mudança tem inegavelmente reflexos em outras áreas, inclusive na produção. Esta tendência oriunda da área de TI associada à orientação a serviços oferece uma solução atraente e natural, propiciando maior facilidade de integração, contribuindo para a produção em mundo globalizado de forma distribuída geograficamente, com sistemas diversos utilizando a mesma base de dados para a tomada de decisões, podendo com maior facilidade terceirizar parte ou toda a produção para locais remotos com menores custos.

4.1 - Conceitos preliminares

A arquitetura AMOS propõe que sistemas ciberfísicos sejam capazes de se autoconfigurarem, autodescreverem e ofertarem serviços com menor intervenção para outros elementos de produção de baixo nível no chão de fábrica. Além disso propõe uma metodologia que será descrita em seguida.

A tese propõe dois modelos que foram descritos no capítulo 3, uma arquitetura e uma metodologia de referência que será descrita ao longo deste capítulo (vide figura 17), que auxilia o desenvolvimento de soluções baseados em serviço via interfaces, baseando-se nas melhores práticas e em uma arquitetura referência modular, abstrata, para que venha a ser no caso concreto implementada.

Devido à diversidade de possíveis cenários de aplicações e processos, esta tese propõe uma arquitetura e metodologia abstrata apta a nortear o desenvolvimento de futuras aplicações para a manufatura avançada.

4.2 - ARQUITETURA MODULAR BASEADA EM COMPONENTES E EM SERVIÇOS

A partir desta seção detalha-se como está estruturada a arquitetura orientada a serviços e uma metodologia de implantação, que serve ao objetivo proposto na tese.

Conforme (RATHWELL, 2001) o projeto da planta industrial e de sistemas informação deve começar com um Planejamento da ARQUITETURA, se beneficios de longo prazo e crescimento são objetivos a serem alcançados.

Com a proposta de uma arquitetura referência modular, a depender da complexidade e do processo, a manufatura pode iniciar por pequenas etapas, que permitam uma migração mais segura, iniciando com funcionalidades simples no chão de fábrica, simples troca de dados via *webservices* e incrementalmente evoluir até complexos sistemas com uso de serviços compostos colaborativos. A arquitetura proposta é planejada para ser gradualmente aplicada em 3 (três) estágios (figura 16), inicia-se a partir de um estado onde a manufatura não trabalha com sistemas orientados a serviços, e aborda sobre como modelar e introduzir serviços para que se mude o paradigma vertical e hierárquico.

Descreve-se os estágios propostos:

- Estágio 1.0 Neste estágio é previsto para as soluções em estágio inicial que não trabalham com serviços, e nele descreve-se como iniciar o desenvolvimento orientado a serviços. Na arquitetura referencia-se ao módulo IntraProcesso.
- Estágio 2.0- No estágio 2.0 os dispositivos já possuem capacidade de processamento e comunicação via serviços, mas de forma limitada e com serviços simples. Possuindo capacidade de com estruturas de controle e computadores de processos em camadas superiores. Não há colaboração entre os pares. Esse estágio referencia-se ao módulo Interprocesso da arquitetura.
- Estágio 3.0- O estágio 3.0 é um estágio de aplicações mais maduras e

empresas que já realizaram projetos prévios utilizando serviços, onde aplica-se algoritmos de analíticos com uma capacidade de predição e inferência nos dados estruturados e não-estruturados, tomando-se ações antes que problemas críticos ocorram. Neste estágio se planeja e executam serviços complexos e onde os sistemas ciberfísicos colaboram entre si para realizar o(s) objetivo(s) do processo. Na arquitetura referencia-se ao módulo computação baseada nas nuvens.

A figura 16 mostra os estágios de maturidade da manufatura relativo ao uso do conceito de serviços na automação industrial, sua capacidade de interação entre elementos e aplicação dos conceitos de manufatura avançada. Esta capacidade de prover serviços está associada à arquitetura AMOS e seus componentes que serão descritos ao longo do capítulo.

Propõe-se uma arquitetura e uma metodologia de implantação, com princípios a serem seguidos e componentes que materializam o conceito da manufatura avançada.

Descreve-se a arquitetura em maiores detalhes na figura 17.

3.0
2.0
IntraProcesso
IntraProcesso
Maturidade no uso de serviços

Figura 16 - Estágios de maturidade no uso de serviços

Fonte: Elaborado pelo autor

O foco da arquitetura AMOS (figura 17) é mostrar a nova interação entre o chão de fábrica, e os sistemas corporativos com as características diferentes em cada instância, os diferentes serviços promovidos por cada instância. Com o uso de serviços, acaba-se com a tradicional estrutura verticalizada da automação industrial e busca-se ampliar a capacidade de decisão da gestão baseada em inferência nos dados do chão de fábrica combinados com outras fontes e sistemas. Com o uso de serviços é simplificada a adequação e composição de serviços para atender novas demandas e requisitos de processo ou realizar integrações.

A arquitetura AMOS proposta busca o nortear o desenvolvimento de interfaces para o desenvolvimento de soluções baseadas em serviços aplicadas a manufatura e seguirá os estágios propostos na figura 16.

Interface essa que deve ser adequada à necessidade de usuário ou

dispositivo, utilizando a detecção do contexto aplicado a diferentes visões de usuários. Por exemplo, as diferentes perspectivas e visões e integrações necessárias de um mesmo sistema quando observadas por um operador, pela manutenção, pelo controle da produção, pela gerência da produção, pelo setor de vendas, entre outros. Um mesmo dado obtido de uma máquina vai servir a diferentes contextos.

Outro aspecto importante da interface é a adequação da plataforma de software e hardware do usuário, detectar a quantidade de dados, processamento suportado pelo dispositivo, e adequar ao melhor protocolo para seu reconhecimento e armazenamento, conhecendo suas limitações. Ex: Dispositivos móveis.

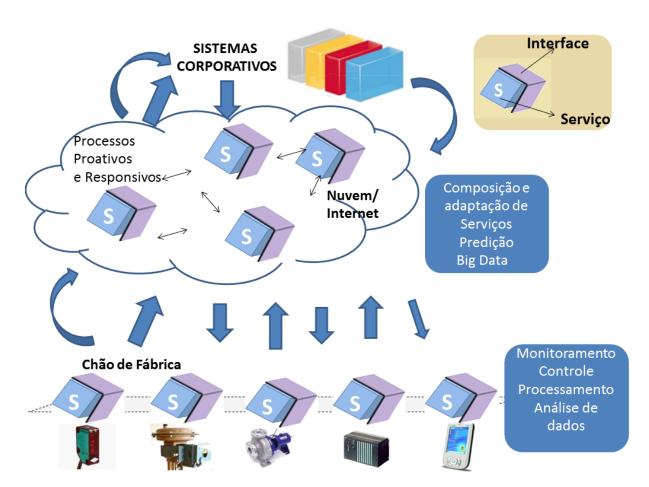


Figura 17- Visão da arquitetura proposta para desenvolvimento baseada em serviços

Fonte: Elaborado pelo autor

Descreve-se detalhadamente a arquitetura AMOS e propõem-se para cada estágio um módulo com conceitos e componentes de acordo com o modelo do capítulo 3 (figura 12).

4.2.1 - MÓDULO INTRA-PROCESSO

O módulo intra-processo é adequado ao estágio 1.0, mais baixo nível de implementação de serviços, e se preocupa com o estabelecimento da comunicação entre os dispositivos utilizando os preceitos de serviços.

Nesse módulo descreve-se as dentro das funcionalidades ou tarefas e informações que serão providas. Para que o paradigma orientado a serviços ocorra faz-se necessária uma mudança na forma de se programar os sistemas, pois em geral os integradores e desenvolvedores de soluções tem acesso a ler memória dos dispositivos diretamente, não se importando que com isso a solução se torna rígida e inflexível caso ocorram futuras mudanças na regra de negócio ou nos sistemas de hardware e software. Faz-se necessário, portanto, reduzir esse alto acoplamento (vide glossário) entre os sistemas, e deve-se estabelecer a comunicação via troca de mensagens, ao invés da forma acima descrita. Neste cenário, a integração é feita conforme o estágio 1.0 descrito na figura 18.

Estágio 1.0 (vide figura 18)- Neste primeiro estágio, utiliza-se o gateway como link entre dispositivos com menor poder de processamento ou comunicação, estruturas de controle e computadores de processos em camadas superiores. Esses dispositivos podem ser simples sensores, ou dispositivos mecatrônicos com limitação de bateria, processamento, comunicação, ou então sistemas legados onde seja inviável tecnologicamente seu upgrade.

Sistemas Corporativos

Nuvem

Gateway

Gateway

SCF

SCF

SCF

SCF

SCF

SCF

Figura 18 - Estágio 1.0 - Elementos conectados via Gateway

Fonte: Elaborado pelo autor

O módulo intraprocesso cria uma interface que abstrai a complexidade da implementação e expõe em um formato padronizado (de acordo com o modelo de dados escolhido) de forma que clientes possam ter acesso as funções e dados.

O primeiro passo a se analisar no módulo intraprocesso é decidir quais informações do dispositivo, recursos e quais as tarefas que ele é capaz de oferecer ao sistema integração.

Com isso uma tabela com nome do recurso, tarefas, nome das variáveis e tipos dos dados e se é de entrada ou saída (E/S) deverá ser criada conforme exemplo abaixo.

Dispositivo: AtuadorX01

Função : MoverPeça	Função: ConsultaStatus
Nome da Variável: Tipo : E/S	Nome da Variável: Tipo : E/S
PosiçãoinicialX: Real : E	PosiçãoatualX: Real : S
PosiçãoinicialY: Real : E	PosiçãoatualY: Real : S
PosiçãoinicialZ: Real : E	PosiçãoatualZ: Real : S
PosiçãoFinalX: Real : E	Status: String: S
PosiçãoFinallY: Real : E	FimdaExecução:Boolean : S
PosiçãoFinalZ: Real : E	
Status: String : S	
FimdaExecução:Boolean : S	

Como descrito na tabela acima, para realizar a funcionalidade MoverPeça no dispositivo AtuadorX01 precisamos informar a posição inicial nos 3 eixos, (X,Y,Z). O serviço disponibilizará uma variável de saída com o fim da execução do tipo Booleano informando se foi executada com sucesso (*True*) ou se ocorreram problemas (*False*).

Após mapear as funcionalidades, dados e tipo de dados que são relevantes, a segunda etapa é analisar se este dispositivo tem capacidade de se comunicar utilizando protocolo de dados abertos orientados a serviços.

Esta abordagem segue uma visão onde um dispositivo físico além compreender componentes eletrônicos e peças mecânicas, muitas vezes pode também incluir um *firmware*, que seja já de fabrica orientado a serviços, ou permita ser modificado, incluindo recursos e serviços incorporados em

tempo de execução no topo de uma interface de comunicação baseada em IP.

Caso dispositivo não possa ter *firmware* orientado a serviços, com isso deve-se projetar um *gateway*, que será responsável por executar as tarefas de tradução necessárias e se comunicar com o dispositivo legado utilizando padrões de comunicação anteriores. E a partir destes dados exporá os dados em algum modelo orientado a serviços. Como exemplo, sempre que um serviço é acionado no mundo real, irá gerar um evento no *gateway* que traduz a mensagem nos protocolos legados e, após processar sua saída, o traduz novamente em padrão orientado a serviços, antes de enviá-lo de volta para o solicitante, vide figura 19, para em seguida se realizar as etapas acima descritas.

Protocolos
baseados em
Serviços

Gateway

Dados dos
sistemas
legados

CLP

Figura 19- Fluxo de comunicação com o uso de gateway

Fonte: Autor

A figura 19 apresenta um exemplo de um CLP que não oferece suas funcionalidades como serviços, e necessita de um *gateway*, que receba os dados de saída do CLP, os processe, e adeque a um protocolo orientado a serviços, e que este exponha os serviços do CLP a outros elementos de produção da rede. Com o módulo intra-processo implementado, será então possível para qualquer outro dispositivo ou sistema compatível, interagir e trocar informações com os serviços do sistema, o exemplo utiliza um CLP, mas poderia utilizar qualquer outro dispositivo legado.

É importante salientar as configurações que devem ser realizadas pelo

gateway. Caso, o dispositivo possua um *firmware* orientado a serviços, ou após o *gateway* ser implementado, será disponibilizando diretamente ao usuário alguma interface que permita se descrever a regra de negócio e a exposição de dados via protocolos IP e *webservices*.

Então, algumas configurações devem ser realizadas:

- Descreve-se a configuração própria (do dispositivo), geralmente o seu IP e porta, e do *gateway*.
- Em seguida as regras de negócio e processo a depender do dispositivo utilizado, mas geralmente utilizando a linguagem textual estruturada ou XML.
- Definição do modelo de dados implementado (OPC UA, XML, gSoap, etc)
 que se encarrega do mapeamento dos tipos de dados das variáveis
 (Inteiro, char, Boleano, etc) para um modelo padronizado XML), pela
 validação dos pacotes de dados, pelo parsing (vide glossário), transporte e
 entrega dos dados e geração de arquivos auxiliares.

4.2.2 - MÓDULO INTERPROCESSO

O módulo comunicação Inter-Processo refere-se estágio 2.0 (vide figura 20). Neste estágio os dispositivos já possuem capacidade de processamento e comunicação via serviços, podendo interagir diretamente com estruturas de controle e computadores de processos em camadas superiores, busca-se oferecer serviços mais elaborados que no estágio 1.0.

Corporativos

Repositório
de Serviços
Serviços
dinâmicos

Chão de Fábrica

SCF

SCF

SCF

SCF

SCF

SCF

Figura 20 - Estágio 2.0 - Elementos se comunicam com maior capacidade via serviços

Fonte: Elaborado pelo autor

O módulo Interprocesso é o nível onde se busca monitorar, controlar e atuar no processo usando os princípios SOA para mediar à comunicação entre sistemas corporativos e sistemas mecatrônicos. Com a base construída no módulo intraprocesso, esse módulo será a definição de como se realizará o orquestramento, quais serviços serão disponibilizados, como se dará o acesso, o monitoramento, a definição do fluxo do processo e eventos controlados.

Em um sistema de produção orientado a serviço, todos os elementos podem expor suas habilidades na forma de serviços. E neste estágio os elementos irão realizar interações que podem incluir a descoberta, troca de dados e metadados(vide glossário) que são o instrumento para transformar dados brutos em conhecimento, ou até mesmo recursos. Além da

necessidade de implantar facilmente os dispositivos e serviços, há necessidade para automatizar o processo de identificação, configuração e gestão destes recursos de forma ágil, para implementar o conceito Conecte e Produza na área de manufatura.

O módulo InterProcesso, no estágio 2.0, possui componentes que foram mostrados na figura 12, juntamente com a descrição do modelo (capítulo 3) e serão aplicados a seguir.

- Serviços dinâmicos: Para atender a necessidade de flexibilidade no chão de fábrica, mudanças nos processos ou produtos, alteração nos requisitos, os elementos de produção devem ter serviços que podem ser ativados ou desativados em tempo de execução. Com isso pode-se desenvolver a capacidade de configuração ou *setup* dinâmico e gerenciamento do fluxo via orquestramento. Esses serviços dinâmicos podem ser providos seguindo estes princípios:
 - 1. O orquestrador recebe da gestão a lista de tarefas a serem executas.
 - O orquestrador tem uma macrovisão do processo e do que precisa ser feito, e deve selecionar de acordo com a lista de tarefas quais SCF podem (estão em funcionamento, livres e ativos) realizar a próxima tarefa.
 - 3. O orquestrador pode em tempo de execução solicitar/cancelar serviços de outros SCF que inicialmente não eram previstos.
 - 4. Antes de alocar uma tarefa a determinado SCF, ele envia uma mensagem de confirmação previamente, e se a resposta for positiva que o SCF está livre, ativo e em funcionamento ele aloca ao SCF.
 - Cada SCF descreve seus serviços (normalmente utilizando WDSL) e disponibiliza em um repositório. O repositório pode ser acessado por qualquer outro SCF ou sistema, pois utiliza padrão

WEB e é referenciado através de uma URI.

- 6. Cada dispositivo tem arquivos de configuração e controle a serem executados também disponibilizados no repositório. A cada tarefa executada o SCF retorna ao orquestrador que foi executada com sucesso ou não.
- 7. A sequência de tarefas a serem executadas são acompanhadas pelo orquestrador que notifica/atualiza o usuário e aos sistemas de acompanhamento da produção.
- 8. Neste nível InterProcesso os elementos de produção ainda não trabalham colaborativamente, e a sequência de atividades passa obrigatoriamente pelo orquestrador. Ainda não utilizam conceitos de *Big Data* em busca de padrões e análise de dados mais complexo para realizar um gerenciamento proativo do processo.
- Autocapacidades: No módulo Interprocesso algumas capacidades são disponibilizadas e descritas com maiores detalhes.
 - 1. Autodetecção: capacidade do SCF ser descoberto sem maiores configurações. Isso é realizado semelhante a um computador que se conecta em uma rede cabeada Ethernet via DHCP, onde somente ao conectarmos o cabo no computador, uma mensagem broadcast é enviada, com isso o dispositivo é detectado pelo servidor DHCP, e há uma troca de mensagens, e recebe-se um IP e um arquivo WSDL válido, e começa-se a interagir e acessar a rede.
 - 2. Autoconfiguração: Para materializar o conceito conecte e produza, após o processo de autodetecção, o dispositivo já cadastrado na rede solicita arquivos de configuração ao servidor. Esses arquivos normalmente são descritos em padrão XML/WSDL e o dispositivo deve ser capaz de receber os arquivos, armazená-los no diretório correto e adaptar seu

comportamento com base nos serviços e configurações recebidas. Modificações do processo, requisitos devem ser refletidas em ações nos dispositivos e estes devem estar aptos a mudanças em tempo real.

- 3. Autoreparo: Esta é a capacidade de o SCF realizar testes e detectar alguma não conformidade ou falha. Inicialmente uma reinicialização será realizada e em seguida deverá ser realizado o download dos arquivos de configuração que estão salvos no atualizando os serviços e configurações repositório, dispositivo. Caso não se resolva, um alarme deve ser disparado para o orquestrador e para o responsável técnico do setor. Se o conhecido pode-se estabelecer erro procedimentos automatizados a serem realizados, caso não informações sobre o processo, maquinário devem ser informadas de acordo com as regras da manufatura.
- Repositório de Serviços: No módulo InterProcesso este componente facilitará a descoberta e registro de quais serviços são oferecidos por quais dispositivos em tempo real. Podendo ser visto como as páginas amarelas dos serviços. Ele se materializa por via de um banco de dados interno e de algoritmos de checagem aleatória se os dispositivos se encontram ativos. Após a subscrição, o dispositivo estará credenciado a oferecer e solicitar serviços.

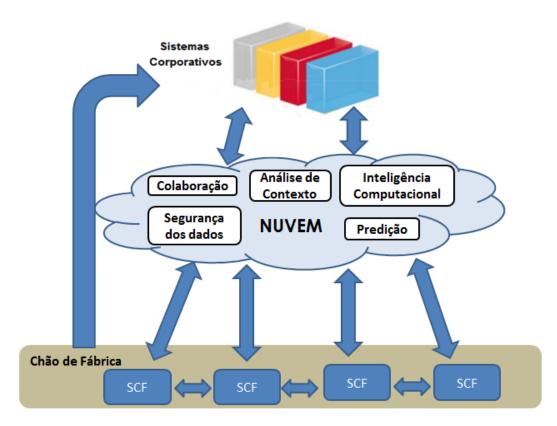
4.2.3 - MÓDULO COMUNICAÇÃO BASEADA EM NUVEM

Este módulo é referente ao estágio 3.0, sendo um avanço do estágio InterProcesso para a computação baseada nas nuvens onde os elementos de produção, além de trabalharem orientados a serviços, podem trabalhar colaborativamente, com o uso de predição e *Big Data* buscando prover serviços mais complexos e um melhor resultado. O objetivo é tornar os processos proativos, que busquem entender não somente o que aconteceu e por que aconteceu, mas o que pode ocorrer e seus impactos. A manufatura

tem aumentado o volume de dados gerados, porém sem o devido tratamento, havendo uma necessidade de informações de alta qualidade em tempo real para tornar as decisões mais assertivas.

Nesse estágio 3.0, uma arquitetura é proposta e componentes do modelo proposto são aplicados (figura 21), que é detalhado em seguida.

Figura 21- Estágio 3.0: Elementos estabelecem serviços complexos e colaboram entre si



Fonte: Elaborado pelo autor

O módulo de computação baseada em nuvem promove uma nova lógica de negócios orientada a serviços, maior interoperabilidade e extração de dados oriundos de diferentes fontes com o uso de inteligência computacional, predição e colaboração, provendo ou recebendo informação dos dispositivos disponíveis em tempo real.

O módulo de computação baseda em nuvens provê princípios e componentes, onde não é obrigatório o uso de todos, a depender do contexto e processo da manufatura. A arquitetura proposta é incremental e modular, e a empresa, baseada em suas necessidades e contextos, deve priorizar o que é mais importante e pode trazer maiores impactos positivos na organização, balenceada com o investimento, tempo e recursos diponíveis.

Para atender a um sistema complexo, o estágio 3.0 deve se nortear com componentes conforme descrito abaixo:

Serviços sensíveis ao contexto: Com uma maior quantidade de informações em tempo real, o componente deve prover informações em tempo real de diferentes fontes na granularidade desejada ao usuário adequado. O uso de indicadores, dashboards interativos para o acompanhamento da produção em tempo real municiar os gestores como está o andamento do processo e ao longo de um período desejado pela manufatura (exemplo de acompanhamento da produção na figura 22), possibilitando uma melhor visualização e mais intuitivaO componente terá a responsabilidade de ajustar a visão dos dados adequadamente ao usuário e para o meio de acesso escolhido. Por exemplo, o acesso e informações disponibilizadas sobre as ordens de serviço para os gestores de vendas via dispositivos móveis deve ser diferente da utilizada pelo operador de máquina CNC que necessita adequar-se aos pedidos seguintes. Para que isso se materialize deve-se a interface de acordo com as particularidades complexidades, capacidades de processamento do dispositivo e nível de informação/acesso adequado, e de forma transparente ao usuário.

Produção / Análise Detalhada

Primeira quinzena de janeiro de 2017

Resumo

Resumo

ACUMULADO

DIÁRIO

FALHAS

Acompanhamento Acumulado

12500
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
10000
1

Figura 22- Exemplo de tela de controle com indicadores em tempo real

Elaborado pelo autor baseado em Ratis Sistemas

- Inteligência Computacional: um componente que tem forte ligação com a predição, e é responsável por capturar, realizar os tratamentos nos dados e integrar fontes. Analisar e inferir principais dados e informações do processo fabril, utilizando algoritmos matemáticos, Big Data, mineração de dados e com o apoio de inteligência artificial ele mostrará notificações automaticamente sempre que existirem padrões definidos, poderá inserir algoritmos ou de aprendizagem (aprendizagem de máquina) para que com auxílio humano possa acontecer uma análise de melhor qualidade e assertiva. Este componente tem como missão analisar a base de dados interna, e realizará inferências buscando promover maior autonomia e com maior capacidade de raciocínio, se beneficiando da maior capacidade de processamento dos dispositivos.
- Predição: No módulo computação baseada em nuvens a predição é

associada à inteligência computacional podendo realizar prognósticos e analisar não somente a base de dados internos, mas utiliza dados referentes a desempenhos de dispositivos semelhantes buscando similaridades e funcionamentos abaixo dos padrões estabelecidos, gerando alerta а gestão. Além disso, busca inferir para comportamentos futuros, buscando-se obter o ponto ótimo para realizar alguma ação preventiva e para tal informará ao responsável técnico ou gestor responsável. Busca-se com esse elemento implantar uma análise preditiva mais eficaz fazendo com que as manutenções por quebra ou as paradas não planejadas sejam reduzidas.

- Segurança dos dados: Em um sistema de manufatura que pode ser acessado em tempo real, este quesito tem alta relevância, e deve ser garantido para um funcionamento confiável. A estratégia de segurança pode estar associada ao modelo de dados, pois alguns já possuem criptografia dos dados (ex. OPC UA), ou pelo sistema de controle com regras de autenticidade, disponibilidade e integralidade. Este componente pode ser vigilante e possuir identificação proativa de invasões e ataques cibernéticos, com plano de ação previsto para um sistema integrado com diferentes fontes heterogêneas. Outro quesito é a segurança física, deve ser controlado o acesso a pessoas e dispositivos com USB que podem estar infectados e com isso trazer malwares para a empresa.
- Colaboração: Neste nível, os sistemas ciberfísicos têm uma capacidade maior de interação, e capacidades como autootimização e autoorganização. Este componente descreve regras para que os elementos trabalhem em conjunto buscando o objetivo comum, armazena-se dados no banco de dados interno, alimenta-se os componentes de inteligência computacional e o componente predição. Por exemplo, em uma situação de falha o módulo colaboração é acionado e pode conter regras de substituir serviços por outros, paralisar operação e/ou acionar elementos ou pessoas externas.

Os componentes descritos acima permitem a evolução da manufatura e permitem promover serviços colaborativos, com uso mais avançado da dos avanços tecnológicos e tecnologias oriundas de TI. Nos últimos anos tem aumentado o número de sistemas distribuídos com menor interferência humana, e que tem se tornado mais automatizados, autônomos e autoconfiguráveis. A arquitetura proposta nesta tese descreve serviços entre os atores de cada camada. Esses serviços se materializam através de interfaces padronizadas, usando modelo de dados orientados a serviços, para com isso mais facilmente se adaptarem a diferentes fontes de dados.

4.3 – Desenvolvimento de soluções orientadas a serviço com a arquitetura AMOS

Nesta seção busca-se nortear etapas para a materialização e subsídios para implantar a arquitetura AMOS. Na abordagem adotada, a implantação deve ser feita a partir de um ambiente de teste, em unidades pilotos para que a manufatura adquira maturidade nos estágios propostos pela arquitetura.

A partir da arquitetura, uma série de passos são propostos (figura 23). Segue o detalhamento sobre cada etapa, e com os quais foram o subsídio para os quatro estudos de caso que validaram a abordagem proposta.

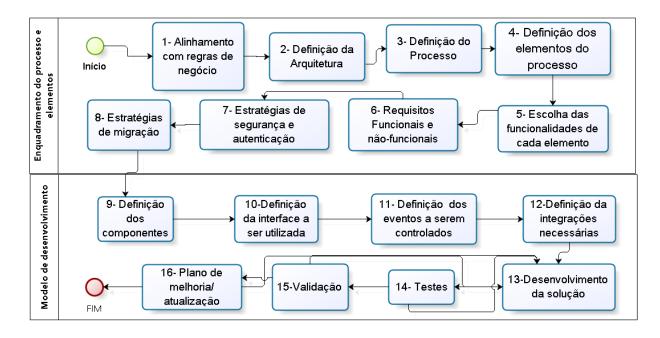


Figura 23- Visão geral da metodologia proposta na tese

Fonte: Elaborado pelo autor

Descreve-se agora as etapas da metodologia com maiores detalhes:

- 1 Alinhamento com as regras de negócio: Nesta etapa estabelece-se quais as vantagens esperadas e como será feita a integração, qual a área, as interações e com quais elementos (software ou hardware) os serviços devem prover dados e receber dados.
- 2- Definição da arquitetura: esta tese propõem uma arquitetura referência para embasar desenvolvimento de soluções orientado a serviços, que será melhor detalhado no próximo tópico.
- 3- Definição do(s) processo(s) a ser(em) automatizados ou melhorados com o novos paradigmas e tecnologias estudadas ao longo deste trabalho. A empresa deve selecionar um processo de acordo com a sua própria maturidade em trabalhar com orientação a serviços ou se for sua primeira iniciativa de se trabalhar com serviços, implementar a troca de mensagens entre os elementos de produção conforme o módulo Intraprocesso. Definir a

área, a equipe e seu envolvimento. Para tanto um modelo de planejamento de migração orientado a serviços foi proposto. Vide apêndice A.

- 4- Definição dos elementos do processo que farão parte, qual maquinário e sistemas envolvidos.
- 5- Escolha das funcionalidades de cada elemento: Dentro da estrutura de automação dos elementos previamente selecionados, escolhe-se as funcionalidades a serem ofertadas. Como em um ambiente de produção diversos elementos de produção trabalham de forma sinérgica e cada elemento pode oferecer um ou mais funcionalidades, por exemplo: Robôs podem oferecer funcionalidades mover a direita, mover a esquerda, pegar objeto, largar objeto, suspender objeto, entre outras funcionalidades. Devese escolher quais funcionalidades serão selecionadas e que serão oferecidas na forma de serviço.
- 6- Requisitos funcionais e não-funcionais: Requisitos são as métricas e características que devem ser atendidas pelo sistema. Requisitos de acordo com (SOMMERVILLE and SAWYER, 1997), são funções, objetivos, propriedades, restrições que o sistema deve possuir para satisfazer contratos, padrões ou especificações de acordo com o(s) usuário(s). De forma mais geral um requisito é uma condição necessária para satisfazer um objetivo. Requisitos funcionais descrevem explicitamente as funcionalidades e serviços do sistema. Documentam como o sistema deve reagir a entradas específicas, como deve se comportar em determinadas situações, ou que o sistema não deve fazer. Exemplo: O sistema deve oferecer ao usuário as linhas de produção possíveis e disponíveis no momento, telas apropriadas para o usuário ler a vazão e pressão em tempo real. Cada produto deve ser associado a um identificador único (ID), o qual o usuário poderá rastrear sua fabricação e características inerentes. Requisitos não-funcionais definem propriedades e restrições do sistema. Dizem respeito a aspectos como segurança, desempenho, usabilidade, espaço em disco.

Podem ser do sistema todo ou de partes do sistema. Requisitos nãofuncionais podem ser mais críticos que requisitos funcionais. Se não satisfaz, o sistema é perde sua razão de ser. Nesta etapa deve ser criada uma tabela com os requisitos funcionais e não funcionais, sua descrição, de forma a documentar quais requisitos devem se atendidos e classificando-os por prioridade se essencial, importante ou desejável. Para maiores detalhes sobre requisitos funcionais e não-funcionais vide (TURINE and MASIERO, 1996).

7- Estratégia de segurança e autenticação: Nesta etapa utiliza-se o requisito não funcional de segurança da etapa anterior ou define-se nesse momento. Diversas estratégias podem ser utilizadas de acordo com o nível de criticidade e se a rede é local ou tem acesso a redes WAN. Deve-se ter em mente estratégias de seguranças físicas que visam controlar o acesso de pessoas e o uso de dispositivos de armazenamento móvel como USB que tem sido causa de infecções em empresas. Além disso, aspectos relativos a segurança de rede referem-se a uma estratégia e provisões de uma organização para garantir a segurança de seus recursos e de todo o tráfego de rede. A segurança de rede se manifestar na implementação de uma política de segurança, de hardware e software, analisando os riscos de segurança. Para se defender e implementar políticas de autenticidade e boas práticas em segurança de rede são recomendadas, prescrevendo para que a rede seja protegida em camadas. Essas camadas aplicam uma variedade de controles de segurança para examinar bem as ameaças que tentam entrar na rede, como: Controle de acesso, autenticação, identificação, detecção de malware, criptografia, firewall, filtragem de URL (Localizador Padrão de Recursos), inibindo que dados fujam para uma rede externa, entre outros. Deve discutir e materializar quais estratégias serão adotadas em e analisar os requisitos e tradeoffs (vantagens e desvantagens) de segurança, elaborando um plano de segurança de acordo com a necessidade.

8- Estratégias de migração: De acordo com o processo, a capacidade do dispositivo em se comunicar com protocolos orientados a serviços, haverá a necessidade dos sistemas ciberfísicos embarcarem em sua memória a parte cliente/servidor destes protocolos ou de se criar um *gateway* que será responsável por criar uma interface com elementos de menor capacidade de

processamento ou capacidade de comunicação recebendo e traduzindo os dados de dispositivos (motores, sensores, válvulas, etc.) que em geral utilizam protocolos proprietários em protocolos abertos e orientado a serviço. A migração deve prever os riscos associados. Cada caso concreto deve realizar essa análise de risco e por se tratar de um assunto complexo, foge do escopo da tese este aprofundamento. O planejamento busca subsidiar e fornecer informação suficiente para ajudar a tomada de decisão final, de implantar nova abordagem para melhorar o desempenho de um processo. Com base neste planejamento estratégico, todas as métricas definidas, informação recolhida, levantamento de necessidades, deve proporcionar a empresa informação que permita responder a questões (no apêndice, modelo de formulário):

- Deve o projeto de migração ser abandonado, reduzido ou adiado?
- O modelo orientado a serviços é apropriado para o negócio?
- A análise, foi suficiente? Ou deve ser retomada e ser aprofundada?
- 9- Definição dos componentes: A arquitetura proposta é dividida em 3 módulos: Intra-Processo, Inter-Processo e computação baseada em nuvens, e cada módulo possui componentes que podem ser utilizados a depender do caso concreto e requisitos.
- 10 Definição da interface utilizada: Esta etapa define qual o protocolo rede e dados orientado a serviços a ser utilizados, como será a busca, descoberta, gerenciamento de serviços.
- 11- Definição dos eventos a serem controlados: Na orientação baseada em serviços, diferentemente da automação tradicional que trabalha em ciclos síncronos, essencialmente trabalha por eventos assíncronos, de acordo com requisição entre cliente para um servidor. Estes eventos ter diferentes origens como uma intervenção do usuário via um comando, ou a ocorrência de um alarme, ou a finalização de um serviço anterior que dispara a execução do próximo.

- 12- Definição das integrações necessárias: Nesta etapa deve-se analisar quais elementos do chão de fábrica podem estabelecer comunicação via protocolos orientados a serviços e quais necessitam de um *gateway*, e como será o processo de integração, e o desenvolvimento necessário para realizar a integração horizontal e vertical.
- 13 Desenvolvimento da solução: Esta etapa envolve decidir qual a linguagem de programação, qual a metodologia de software adequada para o desenvolvimento, qual a equipe de desenvolvimento necessária, se será da própria empresa, ou uma empresa contratada.
- 14- Testes: Uma estratégia de testes deve ser adotada, de acordo com a criticidade do processo, elaboração um banco de testes, elaboração de protótipos. Teste de funcionalidade: aqui o objetivo é avaliar a interface do programa, os links, campos de preenchimento, botões etc. Teste de desempenho: esse teste visa avaliar a performance de um programa, se ele atende aos requisitos mínimos preestabelecidos, como o tempo que demora para dar respostas a ações realizadas e se ele rende como o esperado. Teste unitário: esse visa avaliar pequenas unidades que compõem um software, responsáveis por funções diferentes dentro dele. Podem ser avaliados códigos, sub-rotinas entre outros. O foco aqui é descobrir se todas essas funcionando adequadamente Ajustes necessários partes estão desenvolvimento são esperados e devem ser realizados pelo desenvolvedor ou pela equipe responsável.
- 15 Validação do usuário: Todo processo deve ser acompanhado e validado pelo usuário durante o desenvolvimento. *Feedback* do usuário são essenciais e devem retroalimentar o desenvolvimento.
- 16- Plano de melhoria e atualização: A manufatura avançada busca continuamente utilizar os melhores métodos, tecnologias e para tal deve-se ter um plano de melhoria e atualização que vise avaliar se a mudança de paradigmas atendeu aos requisitos, totalmente, parcialmente ou não atendeu. Se a integração trouxe as vantagens esperadas, se o desempenho

está adequado. Além disso, também de que forma pode ser melhorado o processo, quais serão os próximos processos a serem integrados.

4.4 - Considerações

A arquitetura proposta se materializa via princípios e componentes o paradigma da manufatura avançada, promovendo um modelo de serviços para que se possa avançar ao ponto de termos dispositivos conecte e produza no chão de fábrica e para interligar componentes de software e hardware heterogêneos. Utilizando a AMOS os dispositivos tem capacidade de se conectar, registrar seus serviços, e poder colaborar com o processo fabril de forma simplificada.

A proposta introduz conceitos de predição e *big data* para tornar os processos proativos buscando inferir o que pode ocorrer, ao invés de reativos, reagindo somente após algo acontecer. Introduz também conceitos de sensibilidade ao conceito e indicadores para monitorar e acompanhar em tempo real o desempenho do processo de forma mais intuitiva.

Provê uma metodologia que direciona etapas para facilitar a implementação da arquitetura AMOS proposta.

O intuito é promover uma gestão integrada desde o nível do dispositivo do chão de fábrica para se tornar uma única entidade ágil, em que os seus elementos interajam de uma forma transparente e mais interoperável e, com isso, aumentem a produtividade/resultados do negócio.

CAPÍTULO 5 - ESTUDOS DE CASO

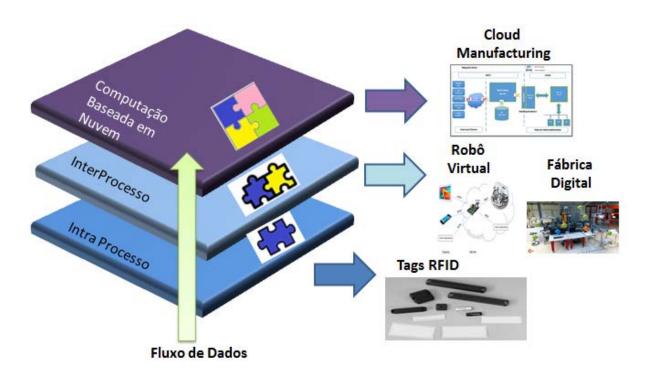
Este capítulo descreve os estudos de caso realizados para comprovar a eficácia e a aplicabilidade do modelo, arquitetura e metodologia proposta nesta tese aplicada a soluções orientadas a serviço. Para realizar tais experimentos, quatro estudos de caso foram pensados com diferentes níveis de complexidade e diferentes tecnologias para demostrar que a orientação a serviço não se prende a nenhuma tecnologia específica, mas é sim uma forma de se planejar e executar processos. Estes estudos de caso vêm validar a metodologia e a arquitetura proposta.

Todos os quatro estudos de caso visam validar a arquitetura e metodologia AMOS foram desenvolvidos na Alemanha durante doutorado sanduíche em parceria com a Universidade Alemã de Ciências Aplicadas Emden-Leer, no grupo de Pesquisa I2AR, sob orientação do Professor Armando Walter Colombo.

A tecnologia de desenvolvimento orientada a serviços aplicada a manufatura ainda está em fase de desenvolvimento, com isso uma das limitações da tese foi a dos estudos de caso terem sido desenvolvidos em ambiente acadêmico, não sendo possível testar em empresas reais, que se dispusessem a implementar o conceito aqui proposto. Empresas da área de automação estão investindo quantias significativas em pesquisas nessa direção e nos próximos anos produtos comerciais certamente estarão disponíveis, com maior maturidade tecnológica e que permitam aplicar os conceitos desenvolvidos nesta tese.

Os quatro estudos de caso (figura 24) foram ordenados em nível de complexidade do menor para o maior para se buscar melhor entendimento e replicação para pesquisas posteriores.

Figura 24- Estudos de caso realizados e seu nível correspondente na arquitetura proposta



Elaborado pelo autor

Para o módulo Intraprocesso um estudo de caso envolvendo a detecção, leitura e escrita de *tags* RFID via serviços foi realizado, para o módulo Interprocesso descreve-se dois (02) estudos de caso: um para controle via dispositivos móveis de um robô orientado a serviços, e outro na fábrica digital, que será descrito no decorrer do capítulo. Por último, um estudo de caso que busca desenvolver, integrar e controlar uma manufatura nas nuvens.

5.1 - Estudo de caso 1- Módulo IntraProcesso: Desenvolvimento de serviços baseados em tags RFID

Este estudo de caso exemplifica como a metodologia AMOS foi aplicada no estágio 1.0 e como as rotinas foram reescritas utilizando protocolos abertos. Ele tem como objetivo aplicar a abordagem para transformar um dispositivo típico de chão de fábrica, no caso, um leitor de *tag*s RFID, para disponibilizar suas funcionalidades via interfaces orientada a serviços

A primeira etapa do método prevê o alinhamento da regra de negócio:

- Neste caso, busca-se somente conectar o dispositivo para trabalhar com serviços.
- Isso será realizado via protocolo IP e com o modelo de dados GSOAP. O processo escolhido foi o de identificação de produtos, utilizando-se um Kit de Desenvolvimento RFID M6E da empresa americana *ThingMagic*.

A próxima etapa é a de **definição da Arquitetura** este estudo de caso se adequa ao estágio 1.0, pois não trabalhava ainda com orientação a serviços. Em seguida a **definição do processo**, porém como este estudo de caso é um processo isolado não se aplica.

Definição dos elementos do processo e das funcionalidades: selecionouse informações sobre o *status* da antena, leitura e escrita do valor do Código Eletrônico do Produto (EPC) registrado em memória, leitura dos dados dos usuários gravados na *tag*, uma função para ler todos os dados disponíveis (série de informações sobre as *tags* no cliente, IP e porta do servidor, números de antenas funcionando e qualidade do sinal) e definição da frequência de leitura (ciclo de leitura) vide figura 25.

RFID-System LER **EPC** LER RSSI Escrever <<include>> FPC Filtrar LER **EPC** Tag Escrever Usuário Ler Antena Parame trizar Info Leitor

Figura 25- Funcionalidades selecionadas para transformar em serviços

Fonte: Autor

Funcionalidades que foram selecionadas:

- -Ler antena Informações sobre o status da antena
- info Leitor Mostra dados como numero de serie, data de fabricação, frequência em hertz
- -Escreve EPC Escreve dados na Tag RFID
- -Ler RSSI Informa como esta o sinal, em decibéis, verifica-se a qualidade do sinal
- -Ler userdata Informa dados dos usuários gravados na tag
- -Escrever userdata Informa dados dos usuários gravados na tag
- -ler tudo Informa as informações disponíveis sobre as *tags* no cliente, IP e porta do servidor, números de antenas funcionando e qualidade do sinal
- -Parametrizar Define a frequência com que as *tags* são lidas, ciclo de leitura, canal e frequência em Hertz.

Admin

Requisitos funcionais e não funcionais:

Foram levantados os requisitos funcionais e não funcionais.

Requisitos Funcionais:

[RF001] Ler Antena

Descrição: Este requisito funcional permite que o usuário leia as informações da antena, para verificar se estão em funcionamento normal. Auxiliando a detectar o erro comum de erro de leitura devido à falta de comunicação com a antena.

Prioridade: (X) Essencial () Importante () Desejável

Entradas e pré-condições: não possui

Saídas e pós-condição: é enviada uma mensagem mostrada em forma textual com a situação atual da antena e dados sobre o sinal em decibéis além da frequência da antena.

[RF002] Parametrizar

Descrição: Este requisito permite que o administrador configure dados sobre, a frequência com que as *tags* são lidas, o ciclo de leitura, canal e frequência em Hertz.

Prioridade: (X) Essencial () Importante () Desejável

Entradas e pré-condições: recebe como entrada o endereço do Leitor RFID, qual a porta de comunicação utilizada.

Saídas e pós-condição: é informado ao usuário se foi possível ou não gravar as informações de acordo com os parâmetros.

Requisitos Não-Funcionais

[RNF001] Usabilidade

Descrição: A interface com o usuário é de vital importância para o sucesso do sistema. Uma interface simples e autodescritiva é importante.

O sistema terá uma interface simples com todas as opções disponíveis sobre as funcionalidades na página inicial e sobre como sair do sistema.

Prioridade: () Essencial (X) Importante () Desejável

[RNF002] Desempenho

Descrição: O sistema deve verificar e controlar seu funcionamento evitando o travamento, deve informar ao usuário que caso não obtenha resposta em 8 segundos considera-se sem comunicação, isso sem travar o aplicativo.

Prioridade: (X) Essencial () Importante () Desejável

Estratégias de segurança e autenticação: Não foi necessário, neste estudo de caso, se estabelecer estratégia de segurança, pois a rede era isolada das demais, e com pouca diversidade de perfis de usuários.

Estratégias de migração: como era IntraProcesso, o desenvolvimento ocorreu isoladamente.

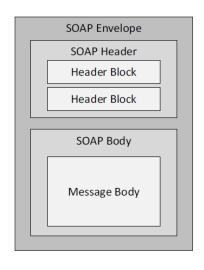
Definição dos componentes: Por ser um estudo de mais baixo nível, não existem componentes.

Definição da interface a ser utilizada: Escolheu-se o modelo de dados GSOAP, pois este modelo seria mais facilmente desenvolvido com a linguagem C usada para a criação de rotinas e convertê-la para um modelo baseado em XML (figura 26). A escolha do modelo de dados foi prioritariamente definida pois poderá futuramente ser integrada ao estudo de caso da fábrica digital e porque a linguagem C era dominada pela equipe desenvolvedora.

O processo de conexão com o servidor definido, de conversão e desconversão dos dados é interno ao protocolo e é descrito em maiores detalhes em: http://www.cs.fsu.edu/~engelen/soap.html

O sistema no kit M6E traz uma SDK para C, C# e Java e pode ser utilizado para o desenvolvimento ao longo de uma interface USB e rede Ethernet. Este kit foi escolhido por sua ampla biblioteca de funções o que auxilia em se desenvolver rapidamente e ele é independente de plataforma, trabalhando com diferentes plataformas como Windows, Linux e MacOS. O ponto negativo é a documentação, não muito extensa, mas o sistema é aberto e possui exemplos de programas com o código-fonte correspondente auxiliando que nosso projeto fosse realizado.

Figura 26- Estrutura de um pacote SOAP



Fonte: Autor

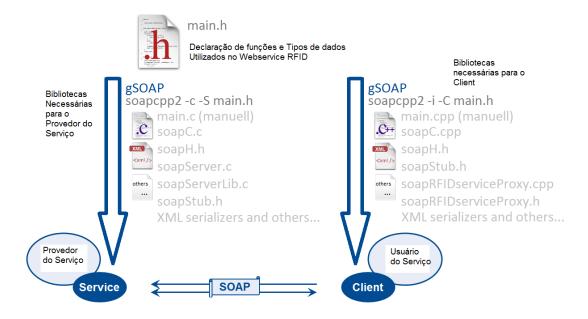
Definição dos eventos controlados: Neste exemplo os eventos controlados foram comandados por entradas de dados pelo usuário via teclado.

Definição das integrações necessárias: Segue o fluxo de informação e em seguida uma descrição do contexto e equipamentos utilizados. Configurações que foram feitas:

- Instalar o Linux Raspbian, que é uma adaptação de Linux para o Raspberry pi, e preparar o ambiente para o desenvolvimento, pois a solução proposta foi de o microcontrolador Raspberry Pi ser o Servidor e responder as demandas dos clientes, isso possibilita uma solução com mobilidade que poderia ser facilmente deslocada para outra planta, com fácil e simples a configuração, pois a configuração complexa fica instalada no servidor, e não se perde uma máquina dedicada somente em prover estes serviços, já que o Raspberry está realizando esse papel(vide figura 30).
- Instalar os pacotes de bibliotecas para o desenvolvimento orientado a serviços, e compilar na plataforma servidor e cliente (figura 27).
- Definir e estabelecer o endereço e a porta do servidor
 - o soap_bind(&soap,"192.168.157.240",1000,100);
- Implementar os serviços via protocolos abertos, neste caso via gSOAP

o O protocolo gSoaP implementa a codificação SOAP em um envelope com cabeçalho (Header) e corpo do texto (Body).

Figura 27- Lista de bibliotecas necessárias para a configuração da comunicação entre Servidor e Cliente via protocolo gSOAP



Fonte: Elaborado pelo autor

Descrição dos componentes utilizados: O Kit de desenvolvimento vem com um leitor RFID (figura 28a) com entrada para quatro antenascom conexão via USB e serial. O kit inclui diversas etiquetas RFID (figura 28c) e trabalha com freqüências ajustáveis, e uma fonte de alimentação para os padrões de EUA, UE, Reino Unido, e Austrália.

Ele tem a conexão para quatro antenas externas com conector RTNC quatro GPIOs ("General Purpose Input/Output"), que são portas programáveis de entrada e saída, bem como uma USB e uma interface RS232 e pode capturar dados de até 400 *tags* por segundo com uma taxa de até 921,6 kbps dos dados transmitidos

4 antenas externas, modelo ThingMagic ANT-12 WB-2043, permitem atingir o alcance máximo. Elas são grandes, aproximadamente 30 cm x 30 cm, ligadas por cabos com 10m de comprimento (figura 28b).

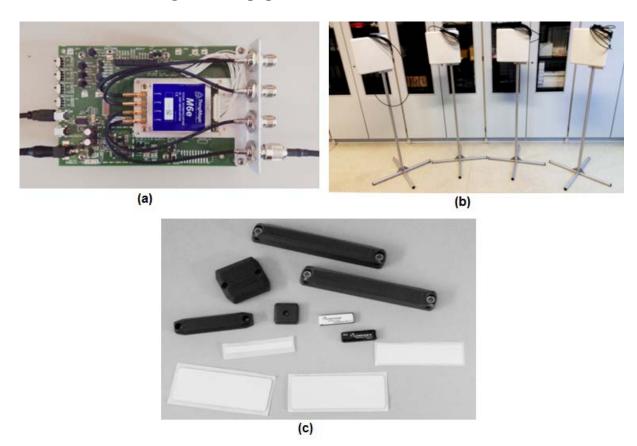


Figura 28- Equipamentos do Kit RFID utilizado

Fonte: Elaborado pelo autor

Desenvolvimento da solução: O módulo servidor foi desenvolvido em hardware Raspberry (figura 28), que oferecia os serviços do dispositivo leitor de RFID, e os publica utilizando o protocolo gSOAP, A linguagem de programação escolhida foi C/C++. Foi criado um módulo cliente que faz as solicitações dos serviços de forma assíncrona e por eventos, utilizando somente os argumentos de entrada de número de IP, chamada da função e seus parâmetros.

Com isso o servidor se conecta ao leitor RFID -> o leitor RFID as antenas RFID-> monitora as *Tags* RFID. Com os dados dos *tags* RFID -> retorna ao leitor RFID-> que devolve as informação ao servidor Raspberry->enviar ao cliente em outro computador. O modelo de dados gSOAP abstrai a complexidade de se adaptar ao formato XML toda requisição de solicitação ou envio de dados, o protocolo possui compilador próprio e *parser* (vide glossário) para validação da estrutura WSDL e XML. Utiliza chamadas remotas a procedures entre cliente e servidor (RPC, vide figura 35).

Testes: Uma série de testes foram realizados, os testes envolvem a gravação de dados em programas fornecidos pelos fabricantes dos equipamentos, e foi desenvolvido um programa com a linguagem C, mas com o modelo de dados gSOAP para que se verificassem se os dados inseridos via programa dos fabricantes seriam lidos de forma correta. Foi utilizado um servidor RFID que gerenciava as antenas RFID e foram testados 3 clientes: 2 notebooks, um utilizando o sistema operacional Linux e outro o sistema operacional linux e em um equipamento Raspberry (figura 29). Todos os clientes receberam as mensagens com os dados corretos.

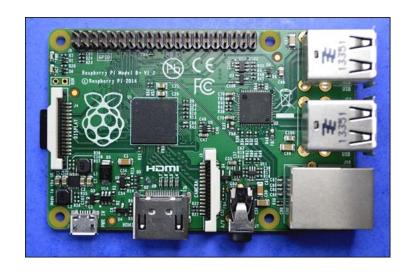


Figura 29- Raspberry Modelo B+ com 512 Mb

Fonte: Autor

Validação: A validação foi dada por usuários do grupo de pesquisa que fizeram parte do estudo de caso.

Como resultado aumentou-se a flexibilidade de se conectar diferentes sistemas mecatrônicos, pois utiliza-se agora baixo acoplamento entre os sistemas e a comunicação entre os elementos se realiza através de troca de mensagens e baseada em eventos. Com isso foi possível testar de forma simples em clientes instalados em dois notebooks com sistemas operacionais diferentes e em um dispositivo Raspberry. Foi possível exportar informações das antenas RFID para diferentes sistemas utilizando protocolos abertos baseados em XML e trocando informações via TCP/IP. Todos os clientes receberam as mensagens com os dados corretos. Foi detectado um atraso médio de cerca de 50 ms no envio e processamento das mensagens, foi testado somente em rede local.

Plano de melhoria e atualização: Não foi realizado para este estudo de caso.

O objetivo deste estudo de caso foi implementar a metodologia e arquitetura AMOS para transformar o leitor de RFID que utilizava funções e sistemas de forma tradicional e fortemente acoplado, para a mudança de paradigma voltado a serviços. Para tal, foi desenvolvido uma interface de software para que pudessem trocar dados via mensagens codificadas pelo modelo de dados gSOAP que utiliza WSDL e XML para troca de mensagens. Desta forma muda-se o paradigma de acesso e interação com o dispositivo.

Maiores detalhes deste estudo de caso e fontes podem ser encontrados no apêndice.

A metodologia e arquitetura AMOS propiciou uma referência adequada ao desenvolvimento, sendo seguida pela equipe de desenvolvimento, reduzindo a curva de aprendizagem dos envolvidos, segundo relatos durante a validação do estudo de caso.

5.2 - Estudo de caso 2 - Módulo InterProcesso: Virtualização de serviços de Robô via Raspberry controlado por dispositivos móveis

Este estudo de caso foi desenvolvido e demostrado para a feira CEBIT² que, segundo dados do Wikipedia, é a maior feira na área de computação do mundo, com mais de 850.000 visitantes por ano.

Ele foi desenvolvido utilizando a arquitetura AMOS e aplica-se ao módulo InterProcesso. Propõem-se utilizar tecnologias orientada a serviços e via serviços para controlar virtualmente um robô móvel via dispositivos como tablets ou telefones celulares e ainda implementar a conceitos como autohabilitação dos serviços, implantação dos componentes autocapacidade e serviços autônomos, em especial autodetecção para concretizar o conceito conecte e produza.

Seguindo a arquitetura AMOS e a metodologia proposta nesta tese, as etapas do estudo de caso foram:

Alinhamento da regra de negócio: Trata-se de um robô educacional, desassociado de outros elementos de produção aos quais pode servir (figura 30).

Deseja-se com esse estudo de caso oferecer funcionalidades de um robô para dispositivos móveis por meio da orientação a serviços, onde qualquer celular, ou tablet pode se conectar e gerenciar o robô.

Informação e Telecomunicações.

² CEBIT Website: http://www.cebit.de/home CEBIT é o acrônimo do alemão (Centrum für Büroautomation, Informationstechnologie und Telekommunikation) que significa Centro de Automação, Tecnologia da

Figura 30- Imagem do robô Robotino

Fonte: Autor

Definição da Arquitetura: Este estudo de caso se adequa a comunicação Intra-Processo e Inter-Processo.

Seleção de funcionalidades:

O estudo de caso presente do robô Robotino da Festo selecionou as seguintes funcionalidades que foram disponibilizadas como um serviço:

Os serviços são de mobilidade do robô.

- Forward (para frente)
- Backward (para trás)
- Andar para a esquerda
- Andar para a direita
- Rotação no sentido horário
- Rotação no sentido anti-horário

Requisitos funcionais e não funcionais:

Requisito funcional

[RF001] Fácil conectividade ao Sistema					
Descrição: Este requisito permite que os usuários da Feira de forma simples					
possam se conectar a rede e ao sistema e tenha acesso a controlar o robô.					
Prioridade: (X) Essencial () Importante () Desejável					
Entradas e pré-condições: recebe como entrada o IP e a porta do servidor.					
Saídas e pós-condição: é informado ao usuário se foi possível ou não					
conectar ao sistema					

Requisitos Não-Funcionais

[RNF001] Usabilidade

Descrição: Como este estudo de caso utiliza dispositivos móveis, é essencial que a interface com o usuário permita o correto uso do sistema. Cada dispositivo móvel possui suas limitações de resolução de cores e tamanho, a interface deve ser responsiva e se ajustar para clientes heterogêneos.

O sistema terá uma interface amigável ao usuário com cores claras e botões representando as funcionalidades e como sair do sistema.

Prioridade:	(X) Essencial	() Importante	() Desejável	

[RNF002] Desempenho

Descrição: O robô deve responder ao comando do usuário em menos de 300ms, após isso deve se tentar enviar o comando novamente, na terceira tentativa informar-se ao usuário problema na comunicação com o robô.

Prioridade: (X) Essencial () Importante () Desejável

Estratégia de segurança e autenticação: O acesso ao sistema e controle do robô é feito via uma rede Wifi, com usuário e senha, criada e emitida pelo Raspberry, somente pessoas com este *login* validado poderia se conectar ao Raspberry e acessar a interface de controle do robô.

Estratégias de migração: Foram mitigados os riscos de tecnologia e houve treinamento com os participantes deste estudo de caso.

Definição componentes: Os selecionados dos componentes e implementados foram os de autocapacidade, com o autoreconhecimento e autohabilitação dos serviços, pois no Robô que é controlado por um sistema Linux, e instalou-se um módulo ao iniciar o sistema operacional e finalizar o robô que sempre se conectava ao servidor Raspberry ao ligar e ao desligar e informava o seu status. além de informar automaticamente que determinados serviços estavam disponível aos clientes com o(s) robô(s) ativos(s).

Definição da interface a ser utilizada: Para atingir este objetivo foi desenvolvida uma interface própria, pois o robô não possuía nenhum modelo de dados previamente configurado. Os serviços disponibilizados se alinham com o componente autocapacidades e foram implementados com chamadas REST (vide glossário), via Webservices e linguagem JAVA.

Definição dos eventos controlados: A interação neste estudo de caso é assíncrona, com eventos controlados via o Raspberry e comandados via

dispositivos móveis autorizados e conectados ao sistema, com telas sensíveis ao toque, em especial tablets e telefones celulares.

Definição das integrações necessárias: Para que a integração seja realizada, um dispositivo Raspberry Pi que atua como servidor foi selecionado. Este servidor fornece uma rede sem fio própria. Além disso, como o Raspberry Pi atua como servidor via uma rede sem fio, qualquer dispositivo móvel com a senha de rede Wifi e do sistema pode se conectar e pode utilizar os serviços do robô via uma interface web (figura 31). A idéia neste estudo de caso é a gerenciamento de um Robotino FESTO. O dispositivo cliente irá se conectar ao raspberry, este por sua vez ao Robotino que oferece serviços por uma interface web. O cliente (tablet, *smartphone*, PC) controla o robô via o navegador Web, maiores detalhes na etapa de desenvolvimento.

HTTP
WIAN
REST
Outros dispositivos

Clients
Server

Figura 31- Comunicação entre o Servidor Raspberry, o robô Robotino e os clientes

Fonte: Autor

A figura 31 também mostra os protocolos utilizados entre o cliente (dispositivo móvel) e o servidor (Raspbery Pi) que intermedia os serviços oferecidos pelo robô.

Desenvolvimento da solução: Segue um resumo dos componentes do estudo de caso e descreve como foi realizado o desenvolvimento. Foram utilizados os seguintes elementos:

- 01 robô Robotino com as seguintes conexões de entrada e saída:
- 2 portas USB
- 01 Ethernet
- 8 portas digital e 8 entradas analógicas
- 8 saídas digitais
- Saída adicional do motor para conduzir cargas elevadas Bateria 12V / 5Ah baterias de chumbo.
- 01 Raspberry Pi com placa que um sistema on-chip da Broadcom com um processador de 700MHz (ARM11) e interface HDMI, Ethernet e Wifi.
- Pacote GlassFish: é um framework para Java EE com código aberto para responder as requisições Web e gerenciar a comunicação via REST, que foi o protocolo selecionado neste estudo de caso. O Glassfish foi lançado pela Sun Microsystems e é patrocinado pela Oracle Corporation desde 2010.
- -Linguagem Java para desenvolver os programas, e HTML5 para a interface Web do Cliente.

Configuração necessária

- 1- Instalar, no Raspberry pi, o sistema operacional Raspbian, que é uma variante do Linux adaptado ao Raspberry.
- 2- Instalar o Java e o Glassfish no Raspberry pi.
- 3- Desenvolvimento da servidora e gerenciadora de serviços em Java no Raspberry.

- 4- Desenvolver a interface Web (HTML) para o controle do Robotino (vide figura 37).
- 5- Desenvolver o módulo Plug and play no Robotino,

O Raspberry Pi fornece uma interface de usuário baseada na web, com o qual os clientes podem usar os serviços dos respectivos terminais. Quem gerencia isso é o servidor de Aplicação GlassFish, que é um servidor Web para JavaServer.

Além disso, em execução neste servidor um pacote que escuta continuamente a rede para receber os pedidos. Isso é implementado utilizando soquetes (TCP). Dispositivos, tais como o Robotino podem se inscrever no servidor e registar os seus serviços disponíveis (sob a forma de REST Webservices), ou seja, o servidor tem uma lista de serviços disponíveis e cada dispositivo se conecta ao servidor e envia uma solicitação de registro para que possa passar a oferecer seus serviços a outros clientes.

Os clientes só usam a interface web para comunicação. Para isso, é utilizado simplesmente um navegador web para digitar o endereço do servidor Web do Raspberry Pi, não sendo necessário a instalação de qualquer software no dispositivo que irá controlar o robotino.

Foi desenvolvida uma página Web (figura 32) de forma que qualquer cliente que tenha um navegador Web (tablets, smartphones ou PC), podem agir como um usuário do serviço.

Figura 32- Interface Web desenvolvida para controle do Robotino



Fonte: Elaborado pelo autor

Testes: Foram realizados testes internos na universidade e durante a feira CEBIT. Foi montado um estande na feira (figura 33), onde qualquer visitante, com qualquer dispositivo de diferentes sistemas operacionais, poderi se conectar e utilizar os serviços do robô. O dispositivo móvel necessita como requisito mínimo de uma conexão Wifi, um navegador web HTML5.

Após isso foi informado o IP do servidor, para que fosse possível conectar à rede, depois executa-se internamente um programa que se registra no Glassfish via *socket*, e informar quando o serviço está habilitado ou desabilitado e inicia-se a interface para o controle do robô.

Service-Oriented Cyber-Physical System of the Service-Oriented Cyber

Figura 33 - Estande de Feira CEBIT 2015

Fonte: Elaborado pelo autor

Validação: A validação deste estudo de caso foi realizada na feira CEBIT, entre os dias 16 a 20 de março de 2015 na cidade de Hannover na Alemanha, onde este estudo de caso gerou um protótipo e foi apresentado e utilizado por diversos participantes da feira.

A feira neste ano tinha como tema Indústria 4.0, o protótipo do robô controlado por dispositivos móveis desenvolvido seguindo orientação a serviços segue os princípios do tema da feira.

Os usuários do estudo de caso relataram facilidade de uso, mesmo não tendo nenhum treinamento prévio. Os recursos do robô são oferecidos, abstraindo a complexidade da configuração e a instalação por serviços foi um ponto positivo. A facilidade de integração também foi outro ponto positivo. A restrição ocorreu quando o robô afastou-se mais de 10 metros e perdeu-se o comando do mesmo. Uma outra limitação é o tempo de resposta que às vezes demorava alguns segundos para realizar o movimento, não

sendo instantâneo, provavelmente devido a interferência elétrica e de outras redes móveis.

Plano de melhoria e atualização: Não foi realizado para este estudo de caso.

Análise do estudo de caso:

A aplicação da arquitetura e metodologia AMOS subsidiou o desenvolvimento da solução, baseada na arquitetura foi implementado o componente de autocapacidades, com a autodetecção de robôs disponíveis. No estudo de caso 02 (dois) robôs estavam disponíveis para o controle e autodescrição dos serviços oferecidos pelo robô.

O componente autocapacidade foi materializado de duas formas:

- Uma rotina foi desenvolvida nos robôs que ao iniciarem propagam a informação de que estão online. Ou seja, de forma autônoma e automática os robôs ao serem ligados propagam a informação de que estão ativas e os serviços que oferecem.
- Outra rotina foi desenvolvida no servidor (Raspberry) que ao receber a informação de um robô de sua disponibilidade informa a todos os clientes conectados qual o recurso e quais serviços estão disponíveis via uma chamada no Glassfish para avisar aos clientes sempre que um novo serviço fosse registrado. Ou seja, a autodetecção e autodescrição para concretizar o conceito conecte e produza se materializa com interações entre o sistema operacional da máquina (robô) com o submódulo gerenciador de serviços e desenvolveu-se uma interface de aplicação que realiza operações de forma automática, sem a interação do usuário.

5.3 - Estudo de Caso 3 - Módulo InterProcesso: Implementação de um modelo fábrica flexível e reconfigurável.

Este estudo de caso exemplifica como a arquitetura e metodologia AMOS foi aplicada em um estágio 2.0 e como as rotinas foram reescritas utilizando protocolos abertos. Neste estudo de caso buscou-se implementar os componentes repositório de serviços e serviços dinâmicos do módulo InterProcesso.

Este estudo de caso desenvolveu-se da Fábrica digital da Universidade de Emden-Leer (figura 34), que é composta por cinco robôs, esteiras, uma área de armazenamento, sensores que serão detalhados ao longo do estudo de caso. Esta planta é voltada à pesquisa e vem sendo transformada para trabalhar sob o conceito de *webservices* e orientação a serviços por diversos projetos de pesquisa e trabalhos de graduação e pós-graduação. Trata-se de uma planta de manufatura flexível na qual vários conceitos de automação diferentes têm sido implementados. Atualmente uma arquitetura orientada a serviços é implementada, porém com restrições.

O objetivo do estudo de caso é utilizar o módulo Inter Processo da arquitetura AMOS e a metodologia proposta, para integrar os desenvolvimentos oriundos de diversos projetos anteriores orientados a serviços para se ter uma gerenciamento global da fábrica e resolver a limitação de interoperabilidade. Como a Fábrica Digital usa uma arquitetura orientada a serviços, os dispositivos reagem por eventos, não mais por ciclo como na operação de automação tradicional.

Service Servic

Figura 34- Foto real da fábrica digital do Laboratório da Universidade de Emden/Leer

Fonte: Grupo de pesquisa I2AR

Segue a aplicação da metodologia ao estudo de caso na fábrica digital.

Alinhamento da regra de negócio: Partes anteriores da fábrica digital foram implementadas utilizando o novo conceito orientado a serviço usando o protocolo DPWS. Porém estes desenvolvimentos foram parciais, um robô da marca Mitsubishi ainda não havia sido adaptado a trabalhar com orientação a serviços. Além disso, parte da linha de produção não foi adaptada para o conceito de serviços. Com o estado em que se encontrava a fábrica, não oferecia a possibilidade de realizar um processo de produção completo, havia falta de interoperabilidade entre todos os elementos de automação.

Definição da Arquitetura: Este estudo de caso se adequa ao estágio 2.0 e aplica conceitos do módulo InterProcesso.

Definição do Processo: O processo selecionado foi a produção de casas com peças LEGO (figura 34) através de 05 robôs, que devem trabalhar de forma

harmônica para fornecer o máximo de flexibilidade na criação dos produtos e nos processos de fabricação. Para tal, busca-se monitorar, controlar e atuar entre sistemas de software e sistemas ciberfísicos contidos na planta.



Figura 35- Esteira e palete que são utilizadas para transporte de peças LEGO

Fonte: Grupo de pesquisa I2AR

Seleção de funcionalidades: Para entendimento das funcionalidades escolhidas faz-se necessário uma descrição da planta da fábrica e seus componentes.

A planta da fábrica digital (figura 36) possui cinco robôs com diferentes atribuições. Esses robôs atuam em quatro áreas de trabalho e um centro de armazenamento, maiores detalhes abaixo:

Workplace 1

High Rack Storage

Workplace 4

Workplace 4

Workplace 4

Workplace 2

Workplace 3

Workplace 0

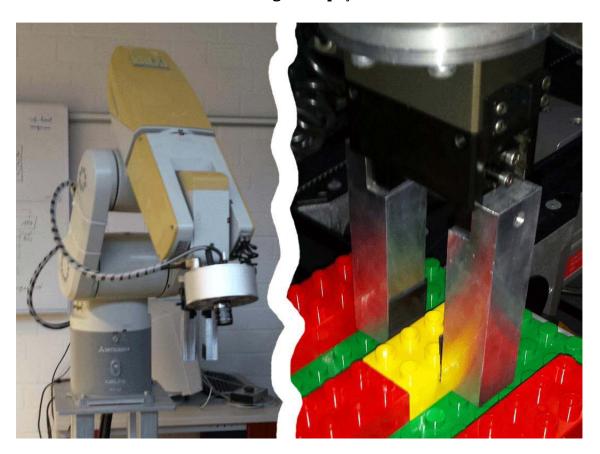
Figura 36- Planta da fábrica digital da Universidade de Emden/Leer

Fonte: Grupo de pesquisa I2AR

- Entrada de transporte (*Workplace* 0). Esta estação de trabalho está montada para a transferência de entrada e saída de partes e consiste de uma correia transportadora e um robô de transferência (Mitsubishi RV-4a S312). O robô de interligação leva paletes com peças LEGO da correia de entrada e gira-se para interligar com a correia principal para transferi-los entre diferentes estações de trabalho ou o para o armazém. Quando o palete com os produtos acabados retorna, o robô transfere-os para a correia de saída.
- Estação de trabalho 1 (Workplace 1) Robô no local de trabalho é um robô articulado Mitsubishi RV, 6 eixos que tem de entregar tarefas de montagem de peças de LEGO que estão vindo da correia principal (figura 37). O Robô tem uma garra (gripper) com pinça especial para pegar peças LEGO e realizar montagem de objetos. O robô está instalado, mas funções de trabalho não estavam trabalhando via serviços com o protocolo DPWS.
- Estação de trabalho 2 (*Workplace* 2) está equipada com um robô Mitsubishi RV 12 SL que é também um robô articulado 6 eixos e tem a tarefa de fazer o mesmo como o anterior. O controle de ambos os robôs é realizado por PC industrial CR2A -572, que é equipado com uma interface Ethernet.

- 4. Estação de trabalho 3 (Workplace 3) O robô de 6 eixos KUKA KR 6/2 realiza as atividades no workplace 3. Quando tijolos LEGO são entregues em um palete pela correia transportadora, este robô é capaz de montar produtos. Tem uma garra com pinça e grampos e uma câmera de vídeo para monitorar e controlar o processo.
- Estação de trabalho 4 (*Workplace* 4) O robô Mitsubishi RH- 5AH está no *workplace* 4, que é robô SCARA (*Selective Compliance Assembly Robot Arm*). Este robô encontrava-se com defeito e não foi utilizado.
- Rack de Armazenamento (High Rack Storage) Este rack de armazenamento pode ser acessado por duas de correia do sistema de transporte principal. Os nomes para as entradas são "leste" e "oeste". Há cerca de 100 espaços disponíveis em duas prateleiras para armazenar produtos produzidos pela fábrica.

Figura 37- Robô Mitsubishi que atua na estação de trabalho 1, com garra especial para montagem de peças LEGO



Fonte: Grupo de Pesquisa I2AR

Requisitos funcionais e não funcionais:

Requisitos Não-Funcionais

[RNF001] Desempenho

Descrição: O sistema deve responder ao comando do usuário em menos de 200ms, após isso deve se tentar enviar o comando novamente, na terceira tentativa informar-se ao usuário problema na comunicação.

Prioridade: (X) Essencial () Importante () Desejável

Estratégia de segurança e autenticação: A fábrica digital da universidade de Emden-Leer, fica em uma rede isolada das demais e tem finalidade educacional, não sendo estabelecida estratégias de segurança e autenticação.

Estratégias de migração: Este estudo de caso é uma extensão de trabalhos anteriores e a fábrica digital já estava funcionando com orientação a serviços, porém limitada. Não houve treinamento com os participantes deste estudo de caso, tendo que ser descobertos ao longo do experimento as limitações e dificuldades de integração, diferentes abordagens o que resultou em um desenvolvimento com tempo maior que o esperado.

Definição dos componentes: Foram implementados durante o desenvolvimento deste estudo de caso os componentes do módulo Inter Processo: Orquestramento, implementação dinâmica de serviços, localizador de serviços e monitoramento de elementos.

Definição da interface a ser utilizada: Como a planta já possuía instalada a interface XML/SOAP INICO S1000 (figura 38) que utiliza internamente o modelo de dados DPWS (vide seção 3.3) esta interface foi a

escolhida. O dispositivo Inico S1000, é fabricado pela empresa Inico³. O Inico S1000 é um CLP inteligente, capaz de controle em tempo real, processamento de dados de campo, com monitoramento baseado em Web e integração de sistemas SCADA/HMI. Este dispositivo fornece uma interface XML/SOAP via protocolo DPWS, em tempo real que permite a integração dos processos industriais em uma arquitetura orientada a serviços. Além disso, a S1000 provê um servidor Web acoplado e uma HMI baseada na Web que oferece uma alternativa visualmente atraente para as interfaces de operação dedicadas.

O dispositivo tem oito entradas e saídas digitais, uma entrada analógica e interfaces de saída e de serviços web. Ele utiliza o padrão Ethernet para se comunicar com os outros elementos. Se for necessário, o controle pode ser expandido com módulos de entrada e de saída adicionais.



Figura 38 -Inico S100 - dispositivo para interface XML/SOAP

Fonte: Autor

³ Website da empresa Inico: http://www.inicotech.com/

Definição dos eventos controlados: Os eventos podem ter origem via usuário, comandos, através da interação entre processos (início, término, alarmes), interação entre elementos de produção. Os eventos disparam gatilhos via *Webservice*, que são tratados por um sistema orquestrador. Cada evento que reflete uma necessidade de produção ou que determinado elemento terminou a execução da tarefa a ele alocada.

Definição das integrações necessárias: Foi necessário criar uma interface entre um robô Mitsubishi, que não trabalhava orientado a serviços, e os demais elementos já haviam sido migrados em trabalhos anteriores. Para criação da interface algumas etapas foram necessárias:

- Foi atualizada a versão do driver do robô Mitsubishi.
- Foi estudado o manual específico das funcionalidades providas pelo fabricante.
- Foi criada uma rotina em linguagem C, que implementa o conceito de chamada remota (RPC) via *socket* (vide Glossário) e onde o robô aguarda chamadas na porta 3789 de seu IP.
- Criou-se uma rotina para ficar "escutando" a porta 3789 para atender cada chamada realizada nesta porta.
- Rotinas específicas com os serviços promovidos pelo robô foram desenvolvidas e associadas aos arquivos WSDL, onde para cada serviço, desenvolveu-se por arquivo WSDL em separado.

Desenvolvimento da solução: O desenvolvimento do estudo de caso envolve os passos necessários para se implementar uma arquitetura orientada a serviço com o uso de Web Services no nível Inter processo desenvolvendo um controle orientado a serviços buscando-se uma execução automatizada de diferentes processos de produção na Fábrica Digital.

Devido à complexidade do cenário, grande quantidade de elementos envolvidos e quantidade de etapas para se atingir o objetivo, um resumo das principais atividades será exposto abaixo. Inicialmente uma descrição dos componentes da Fábrica Digital se faz necessária.

Os processos de produção são controlados de forma flexível através de uma orquestração dos serviços.

Para atingir esta finalidade, uma sequência de passos foi necessária:

 Testar e analisar se os dispositivos Inico estavam se funcionando e sendo gerenciados via sua Web-HMI. A fábrica digital possuía 03(três) controladores inteligentes Inico S100, porém foi detectado que um dispositivo estava com problemas e foi necessário solicitar a compra de um novo. Testar e analisar se a comunicação entre os robôs estava ocorrendo mediante o protocolo DPWS. Para isso o software SOAPUI foi utilizado, este software que auxilia o desenvolvimento e testes dos arquivos SOAP/WSDL. Como é complexo esse cenário com muitos serviços e muitos dispositivos interagindo, essa ferramenta agiliza desenvolvimento arquivos WDSL, sua formatação e testes de validação, que descrevem os serviços oferecidos por cada dispositivo e seu endereço. Segue um exemplo de descrição de partes dos serviços oferecidos pelo Robô Mitsubishi do Workplace 3, descritos no arquivo WSDL com os serviços de receber peça lego, devolver a peça lego e abrir e fechar a garra do robô:

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="MitsubishRoboWP01"
                                              // Nome do elemento a ser definido
targetNamespace="urn:digfabrik"
                                      //Espaço definito de trabalho
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:HTTP="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:WSDL="http://schemas.xmlsoap.org/wsdl/"
xmlns="http://schemas.xmlsoap.org/wsdl/">
<xs:schema</pre>
                       targetNamespace="http://www.technik-emden.de/MitsubishiAP1"
                        xmlns:tns="http://www.technik-emden.de/MitsubishiAP1"
                        xmlns:xs="http://www.w3.org/2001/XMLSchema"
                        elementFormDefault="qualified" blockDefault="#all">
<wsdl:definitions>
 <xs:element name="MoveLegoPartResponse" type="tns:ResponseType" />
  <xs:element name="OpenClawResponse" type="tns:ResponseType" />
  <xs:element name="CloseClawResponse" type="tns:ResponseType" />
  <wsdl:message name="MoveLegoIn">
                                                    <!-- Servico de recebimento da peca LEGO -->
         <wsdl:part name="parameters" element="tns:MoveLegoPart" />
   </wsdl:message>
        <wsdl:message name="MoveLegoOut">
                                                   <!-- Servico de recebimento da peca LEGO -->
         <wsdl:part name="parameters" element="tns:MoveLegoPartResponse" />
   </wsdl:message>
</wsdl:definitions>
```

• Definição das regras da operação da manufatura, quais que deverão ser executadas, cálculos que devem ser feitos, logica a ser utilizada. Esta fase estabelece como deve ocorrer a comunicação M2M (máquina a máquina). Este desenvolvimento é feito com linguagem estruturada na interface Web fornecida pelo dispositivo inteligente INICO. Segue um exemplo das regras de operação da esteira codificada no dispositivo INICO S1000.

```
PROGRAM simple conv control
3
     (* Wait until the operator requests a product. *)
     WAIT UNTIL (push button = true);
     (* Move the conveyor belt until a product is detected. *)
6
     conv motor := true;
    WAIT_UNTIL(proximity_sensor = true);
8
9
    Conv_motor := false;
10
11
    (* Wait until the operator is done. *)
    WAIT_UNTIL (push_button = true);
12
13
14
     (* Move the conveyor belt until the product is no longer
15
        detected, plus a delay to allow enough transfer time. *)
     conv motor := true;
16
    WAIT_UNTIL(proximity_sensor = false);
17
18
    WAIT(2000); (* Additional 2000 msec. *)
19
     Conv motor := false;
20
21 END PROGRAM
```

Com o resultado da etapa de definição dos eventos, será utilizado como entrada nesta etapa de desenvolvimento, que deve seguir as etapas anteriores. Por exemplo: Evento Sensor detecta palete, ativa o robô para pegar as peças LEGO da palheta. Cada evento deve estar descrito no arquivo WSDL e também na interface Inico (figura 39).

Figura 39- Exemplo da criação de um evento



Fonte: Elaborado pelo autor

Na figura acima, ServiceID é o nome do serviço, que será ativado pelo evento ItemTrasferin que é o Alias (atalho) nome que deve estar configurado no arquivo WSDL no *Header* (cabeçalho da mensagem SOAP). E a parte de baixo Edit Message é o que estará descrito no corpo da SOAP *message*.

- Após o desenvolvimento, testes sobre os serviços isoladamente, faz-se necessário um sistema gerenciador do processo que monitore a comunicação via Webservices. Para tal como é um sistema que utiliza padrões abertos pode-se buscar opções de ferramentas de desenvolvimento, sendo feita uma análise sobre possíveis SDKs ou bibliotecas em diferentes linguagens.
- Após uma pesquisa do item acima, foi selecionado o pacote WS4D (Web Services for devices) é uma iniciativa que disponibilizou os resultados do Projeto de pesquisa europeu ITEA SIRENA (BOHN, BOBEK et al., 2006) desenvolveu tecnologia orientada a serviços para a área de automação industrial, de automação residencial, sistemas automotivos e sistemas de telecomunicação. O projeto WS4D disponibiliza ferramentas para "escutar" a troca de dados via protocolo DPWS, e gerenciar os elementos podendo executar ações, além de bibliotecas para desenvolvimento em linguagens C, C++ e Java. Este projeto ainda continua avançando e disponibilizando melhorias sendo gerido pela Universidade de Rostock, Universidade de Dortmund e MATERNA. WS4D disponibiliza bibliotecas em diversas linguagens. O desenvolvimento foi realizado segundo o componente orquestramento da figura 12 descrita no capítulo 3.
- O desenvolvimento do orquestramento deste estudo de caso um sistema utiliza a biblioteca JMEDS, que é a uma biblioteca que implementa o protocolo DPWS com webservices para desenvolvimento usando a linguagem Java (http://ws4d.org/jmeds/). O componente

orquestrador deve monitorar, gerenciar e alocar serviços aos sistemas ciberfísicos via troca de mensagens entre os elementos.

Testes: Na concepção da fábrica digital cada elemento fornece serviços que podem ser utilizados por todos os entes da rede. O sistema orquestrador desenvolvido em Java utiliza os serviços de deslocamento das esteiras para a entrada e saída de palete com a matéria prima para o produto final (casa de peças LEGO). A figura 40 descreve uma sequência de passos para a confecção de uma casa feita de peça LEGO e no fim ser armazenada no Armazém. A interação é assíncrona e baseadas em mensagens SOAP, pois usam o protocolo DPWS, (maiores informações no capítulo 3). Cada elemento envia uma mensagem quando ocorre um evento (término de execução, falha na execução, sensor avisa o recebimento de mercadoria).

Um problema detectado foi que um tempo de espera de 300ms precisou ser manualmente inserido no código após o termino da execução da tarefa de construção da casa LEGO pelo robô, e antes do início do transporte pela esteira que deve transportar o palete com o produto. Senão ocorria um problema de comunicação, este problema de comunicação poderia levar a que mais de uma matéria-prima (palete lego) chegasse no robô, gerando uma falha. Isto poderia acontecer em um sistema industrial real e esse tempo deve ser medido em cada processo. Esse atraso em uma linha de produção com grande quantidade de produtos em série é um problema que teria impacto negativo na planta.

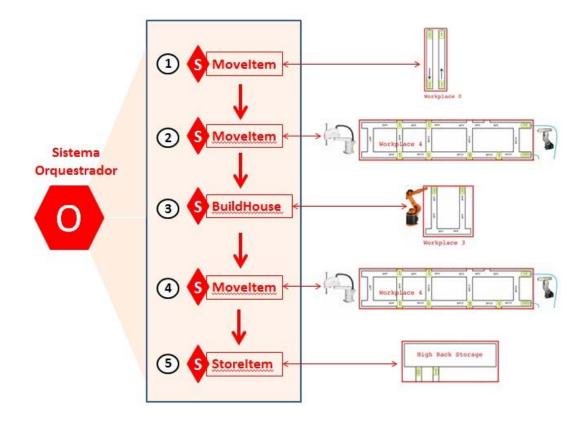


Figura 40- Principais ações controladas pelo sistema orquestrador

Fonte: Elaborado pelo autor baseado em figuras do grupo de pesquisa I2AR

Com essa informação Depois de todo o sistema foi desenvolvido, fazer vários testes de rotas e avaliar as capacidades dos sistemas de produção baseados em SOA. Realizar os testes e analisar os tempos de resposta médio, buscando se otimizar o processo para que possa ser executado no menor tempo possível e com a melhor rota (figura 41).

O tempo médio para a construção de uma casa LEGO considerando todas as etapas (entrada do palete, deslocamento nas esteiras, mudança de esteira por transporte via robô, construção da casa e retorno a saída via esteira) dura em média 10 minutos, isso se estiverem livres as esteiras e robôs, senão terá que ser aguardado o processamento e deslocamento.

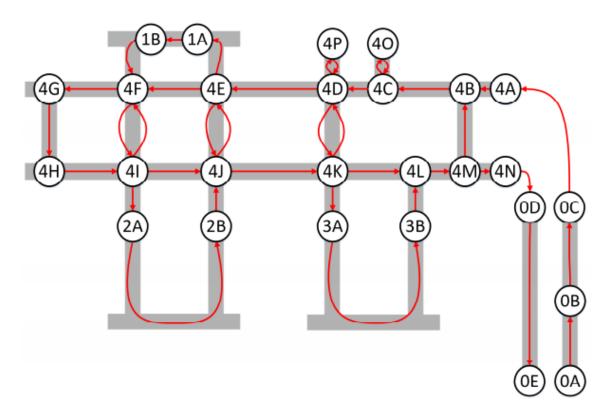


Figura 41- Planta da fábrica e as possíveis rotas para desenvolver o produto final

Fonte: Elaborado pelo autor, adaptado de I2AR Group

Validação: A validação foi realizada juntamente com pesquisadores do grupo de pesquisa I2AR.

Plano de melhoria e atualização: Não foi realizado para este estudo de caso.

Avaliação do estudo de Caso

A arquitetura e metodologia AMOS subsidiou o desenvolvimento deste estudo de caso e os componentes orquestramento, repositório de serviços e serviços dinâmicos foram implementados do módulo Inter Processo.

O componente orquestrador foi desenvolvido de uma forma flexível de modo que diferentes composições sejam mapeadas para o roteamento correto dos paletes, isto foi feito da seguinte forma:

- 1- Para cada combinação de posição entre nós adjuntos um serviço em WSDL/SOAP foi definido.
- 2- Por exemplo, para o encaminhamento de paletes no sistema de transporte existe um mapeamento dos serviços como visto na figura 40. A figura contém referências às operações de serviço web responsáveis para o transporte de paletes entre as posições nas arestas entre os nós. Por exemplo, a aresta que liga o nó "OA" ao nó "OB" contém uma referência à operação de serviço Web "Transport_OA_to_OB". Além disso, , com isso ele sabe se
- 3- O orquestrador contém informações sobre as posições de paletes atuais no sistema de transporte, então o orquestrador sabe que um palete estando em determinada posição ele poderá seguir por determinadas rotas, estas rotas são analisadas em tempo de execução se estão disponíveis o não através de troca de mensagens.
- 4- Então o orquestrador indica a rota mais adequada para a execução da tarefa.
- 5- Ao término da execução um evento será gerado e avisará ao orquestrador para que atualize os estados dos serviços.

O sistema orquestrador tem a informação sobre quais máquinas estão ocupadas e modifica dinamicamente as rotas de produção, e o posicionamento dos paletes para que outro robô possa produzir o produto final adequado no menor tempo e com a melhor rota possível. O orquestramento desenvolvido neste estudo de caso foi centralizado e é informado da situação de cada evento (iniciado, finalizado, bloqueado, pronto, executando, etc.). A figura 40 demostra resumidamente as etapas para a construção da casa LEGO e a atuação do componente orquestrador.

A arquitetura e metodologia AMOS foi avaliada informalmente em debates do grupo de pesquisa, e foram destacados esses pontos:

- 1. Reduzir a curva de aprendizagem para novos desenvolvedores sem experiência em desenvolvimento orientado a serviços.
- 2. Fonte de referência para possíveis funcionalidades e características da manufatura avançada.
- 3. Considerada importante por dar uma diretriz ao desenvolvimento e com etapas claras que subsidiaram o desenvolvimento.

5.4 - Estudo de Caso 4 - Módulo Comunicação baseado nas nuvens: Proposta e implementação de Cloud Manufacturing Via OPC-UA

Este estudo de caso exemplifica como a metodologia AMOS foi aplicada e nível comunicação baseada nas nuvens e como ocorreu o desenvolvimento do componente segurança e a integração entre o processo real e o virtual.

O estudo de caso em tela busca desenvolver uma solução utilizando a arquitetura e metodologia proposta na tese, entre o laboratório de pesquisa de automação WMG (Warwick Manufacturing Group, Warwick University)⁴, e a Universidade de Emden-Leer na Alemanha, para isso virtualizar-se-á a planta e os processos industriais para que sejam gerenciados remotamente e colaborativamente. Neste estudo de caso implementou-se o componente segurança. A AMOS foi utilizada para o desenvolvimento de um módulo orientado a serviços capaz de realizar o monitoramento de uma planta remotamente via o modelo de dados OPC UA.

⁴ WMG- Warwick manufacturing group http://www2.warwick.ac.uk/fac/sci/wmg

O projeto foi iniciado pela WMG e visa proporcionar uma solução de modelagem e gerenciamento virtual voltada para a manufatura avançada, que facilita a instalação de plantas, renovação e sistemas de manufatura usando uma ferramenta própria CCE (Core Component Editor, figura 41), que realiza a engenharia do processo industrial via software, além de permitir a simulação de produção, geração de código pra CLP e exportação independente do hardware e/ou fornecedores de software. Para maiores informações vide (PARK, HARRISON et al., 2014) que descreve o processo de geração do design, até a compilação em código pelo CLP (MORAES, LEPIKSON et al., 2015), que descreve um exemplo de monitoramento com o CCE.

Este estudo de caso envolve virtualização e simulação de processos através de uma ferramenta própria ferramenta desenvolvida pelo WMG Group composta pelos módulos descritos na figura 42.

A ferramenta CCE permite a importação de arquivos com a planta e peças industriais desenvolvidos em ferramentas CAD como Solidworks e CATIA, e com isso realizar a criação de processos industriais, visualizar e simular ações no projeto, planejar as regras de negócio e restrições do projeto, realizar treinamento e interações com operadores humanos e exportação para diferentes CLPs.

V-Lib
Component library

CCE Viewer
Lightweight Viewer

Editor

V-man
Operator modelling

CCE Core
Editor

Figura 42- Componentes da ferramenta CCE

Fonte: WMG Group (2015)

A informação aqui descrita é resguardada por segredo industrial, por regras internacionais e por isso só se discute aqui a parte que interessa à tes. Esse projeto é financiado por fundações europeias de pesquisa para o desenvolvimento de novos padrões e ferramentas de engenharia nas fases de criação, concepção, simulação e validação de projetos voltados à manufatura avançada.

Segue-se agora o detalhamento da metodologia proposta aplicada ao estudo de caso.

Alinhamento da regra de negócio: Este projeto usa o protocolo OPC-UA para estabelecer uma comunicação cliente-servidor para monitorar os processos de produção em tempo real, usando o padrão OPC-UA para monitorar o chão de fábrica e exportar dados via serviços na nuvem para diferentes aplicações. O projeto investiga o desempenho de se monitorar via comunicação nas nuvens uma planta situada na Universidade de Emden-Leer, busca-se analisar se é possível monitorar e atuar em tempo real, via

tecnologia OPC UA, e extraindo dados pela ferramenta de software CCE para simulação e execução.

A figura 43 mostra o diagrama geral do estudo de caso com os elementos envolvidos e a comunicação entre os componentes. Ele foi concebido de forma que houvesse flexibilidade, para não se exigir obrigação sobre onde o servidor web e banco de dados devem ser alocados. Eles podem ser colocados quer ou localmente na fábrica ou remotamente, como parte dos serviços em nuvem. Um software que utiliza o modelo de dados OPC UA foi desenvolvido para monitorar e exportar dados do chão de fábrica.

Figura 43 - Diagrama geral do estudo de caso, onde o chão de fábrica se encontra na Alemanha e o cliente na Inglaterra

Fonte: Elaborado pelo autor

O diagrama do sistema, na figura 43, é composto por componentes diferentes, cada componente tem uma certa função, que é detalhada abaixo:

Sistema -OPC-UA:

- OPC-UA Server, que coleta dados em tempo real a partir dos CLPs
- OPC UA-cliente, que recupera dados do servidor OPC-UA e envia através da interface para o banco de dados.
- Banco de dados armazena as atualizações de dados à medida que acontecem ao longo do tempo. O banco de dados MySQL foi utilizado com uma estrutura que faz o mapeamento entre os componentes físicos situados no chão de fábrica e aqueles dados que devem ser armazenados e são relevantes para a aplicação gerencial.
- O sistema CCE já descrito acima para design, virtualização e monitoramento de processos de automação.

Definição da Arquitetura: Este estudo de caso se adequa a comunicação baseada nas nuvens.

Definição do processo: O próximo passo dentro da metodologia proposta é o processo, que irão ser focadas, neste caso, foi o processo de usinagem, realizado em uma planta FESTO educacional.

Definição dos elementos do processo: que os elementos escolhidos para o estudo de caso foram: uma mesa rotatória, um perfurador, uma bomba e um aferidor. A figura 45 demostra o resultado da virtualização dos objetos selecionados.

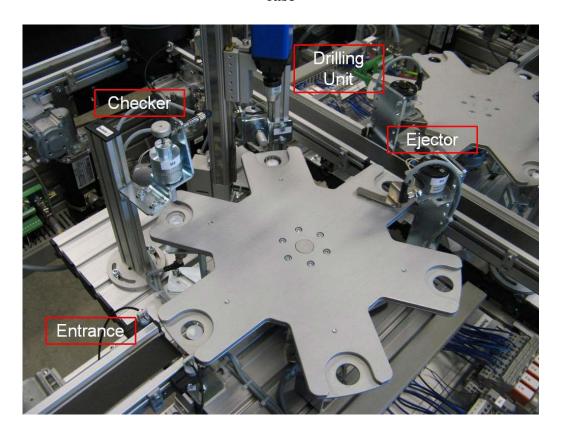
Seleção de funcionalidades: Para explicar as funcionalidades selecionadas, um detalhamento sobre o ambiente do estudo de caso se faz necessário.

Para os testes experimentais do projeto desta e desenvolvimento de simulação de processos virtuais, somente uma parte de uma das estações de usinagem foi escolhido (Figura 44). Esta parte é constituída pelos módulos que são semelhantes à plataforma festo educacional utilizados na planta real.

A parte da planta escolhida consiste das seguintes partes descritas na figura 43:

- Entrada é a correia transportadora de entrada para entregar as peças de trabalho para a estação da mesa rotativa.
- Mesa giratória seis estações de mesa rotativa que transporta as peças de trabalho a partir da entrada para as posições de trabalho e da saída a partir desta unidade de sistema.
- Unidade de Controle verifica a medida de profundidade da peça de trabalho e valida que a peça na posição apropriada para a tarefa adicional de usinagem.
- Unidade de furação simula o processo de furação da peça de trabalho.
- Ejetor esta unidade ejeta a peça para o sistema de transporte (correia transportadora) após a tarefa completa.

Figura 44- Parte do sistema utilizado para a virtualização e validação do estudo de caso



Requisitos funcionais e não funcionais:

[RNF001] Usabilidade Descrição: Deve-se desenvolver uma interface simples e amigável ao usuário que servirá como base para o monitoramento do processo. Prioridade: (X) Essencial () Importante () Desejável

[RNF002] Desempenho

Descrição: O sistema deve monitorar o desempenho e informar aos administradores quando o tempo de resposta entre o envio as mensagens de início de execução para o término do processo demorar mais de 6,5 segundos.

Prioridade: (X) Essencial () Importante () Desejável

Estratégia de segurança e autenticação: Projeto implementa segurança através de criptografia RSA 128 bits e autenticação de usuário conforme descrito no decorrer do estudo de caso.

Estratégias de migração: Não foi realizado para o estudo de caso em tela.

Definição dos componentes: No módulo comunicação baseada nas nuvens, buscou-se desenvolver o componente: Segurança.

Definição da interface a ser utilizada:Devido aos equipamentos utilizados no estudo de caso serem habilitados a utilizar o OPC UA, ele foi escolhido modelo de dados orientados a serviços.

Definição dos eventos controlados: Os eventos controlados foram chegada de peça nova, início e conclusão da usinagem.

Definição das integrações necessárias: Foi necessário a integração com o sistema CCE, para atingir esse objetivo, o sistema cliente OPC, exportará os dados para o banco de dados MySQL de acordo com a estrutura utilizada pelo software CCE e este se responsabilizará por criar um histórico com os dados do processo.

Desenvolvimento da solução: O desenvolvimento do estudo de caso dividese em 2 partes:

1- Processo de virtualização da planta real;

2- Processo de monitoramento da planta virtual e acompanhamento utilizando o protocolo OPC UA.

Descreve-se agora o processo de virtualização da planta real via software CCE. Introduzem-se passos para criar um protótipo e modelá-lo de acordo com a proposta de se atingir uma comunicação baseada nas nuvens, com apoio da ferramenta de engenharia virtual do CCE. Resumem-se, a seguir, as principais etapas e explica os detalhes para a modelagem, prototipagem e geração de código de controle do CLP. Em seguida descreve-se o desenvolvimento da parte do monitoramento com os protocolos OPC UA usado como interface para integrar sistemas heterogêneos e a implementação do componente segurança.

Serão descritas abaixo em maiores detalhes as etapas necessárias para se virtualizar o processo de usinagem e aplicar a metodologia e arquitetura AMOS. Para maiores detalhes do estudo de caso, diagramas de classe, configuração vide apêndice C.

Algumas atividades prévias são necessárias para o desenvolvimento do estudo de caso e serão descritas abaixo:

1- Virtualizar os componentes da parte da planta selecionada para o estudo de caso. Para tal pode-se importar arquivos externos provenientes de outros softwares como o 3D CAD e Solidworks. Além disso, pode-se reutilizar objetos, pois o CCE já possui internamente componentes de objetos prontos para serem reutilizados (figura 45).

Figura 45- Objetos virtualizados para o estudo de caso

- 2- Define-se geometria, que se refere a descrever os atributos físicos do componente. O componente pode ser constituído por várias partes mecânicas, por exemplo, o verificador de profundidade consiste no atuador e base. Cada parte é projetada na aplicação 3D CAD e mais depois compilada para um arquivo VRML para em seguida, poder ser adicionado ao modelo.
- 3- Define-se a cinemática dos objetos no sistema, que são os movimentos de translação e rotação que permite o desenvolvimento de movimentos de dos componentes em relação a algum eixo. A Figura 45 representa a opção cinemática disponível no momento da aplicação de CCE. No lado esquerdo há a descrição do movimento de rotação do anel. Já no lado direito da figura tem-se um exemplo de cinemática de movimento vertical, definindo-se a profundidade da unidade de acionamento.

| ASD | Common | Section | Common | Com

Figura 466- Exemplo de objeto com suas características

4- Definição dos comportamentos: o comportamento define e descreve o fluxo de trabalho do componente pelos estados. Os estados são definidos durante a criação de um novo componente no início e no número de estados não pode ser aumentada durante o desenvolvimento componente.

A Figura 47 mostra o diagrama de transição de estado que descreve o comportamento da unidade verificador de profundidade. Existem três tipos de estados disponíveis: estado inicial, estados dinâmicos e estado estático final. Para cada estado se definem a cinemática e com isso os movimentos necessários para cada etapa do fluxo do processo.

Core Component Editor Version 3.2.3.10 File Edit View Tools Help Route Editor CCE Editor Route: Rout ☐ 🌠 = → D Refresh Export → Name Test System Details - Test 1.1 - Checking Actuator 1 - Rotary_Table 2 None Sensor 1 CheckerDown Sensor 3 WorkpieceArrived[~]on Sensor 2 WorkpieceWorkPos Sensor 3 WorkpieceArrived Floor Festo 1 Workpiece_Red 2 Rotary_Table 2~Move Table Sensor 3 WorkpieceArrived~off Rotary_Table 2~Table off finished Sensor 2 WorkpieceWorkPos~on Checking Actuator 1~S-Pos 1 Manual and Auto Interlocks
 ✓ Require Interlock To Progress None ✓ Show Non Control Items Manual Automatic Checking Actuator 1~S-Pos 2 Cycle Timing Diagram Sensor 1 CheckerDown~on ePosition Rout 1 Checking Actuator 1 Rotary_Table 2 Sensor 1 CheckerDown Sensor 2 WorkpieceWorkPos 0

Figura 47- Descrição da tela para definição de comportamentos

5- A próxima etapa é a do mapeamento entre as variáveis do processo através da interface do sistema CCE Mapper (vide Figura 47), que serve para criar e editar blocos de função e com isso poder gerar o código de CLP automaticamente (vide Apêndice C). Três plataformas CLP estão disponíveis para geração do código CLP: Schneider Electric, Siemens e PLCOpen. Para a criação de um bloco de função, é necessário descrever suas propriedades, variáveis, tipos de variáveis, organizando as entradas e saídas com seus parâmetros.

CCE Mapper File View Build Help frmReusableDataMaintain FB Import Code Import Platform FB Propertie FB4 ⊚ SE PartCheckerSpecia PLCOpen Siemens Туре Options FB a FB Open FB Add FB 1.0 24/02/2014 15:45:27 Edit FB Delete FB FB Vars Data Type ISIO InitialValue Туре FBs in DB SymbolicName INPUT OpMode INT FiveStatePneumatic INPUT BOOL FB3 GripperSpecial a FB OUTPUT PDOWorkPos BOOL Two State Electronic a FB a FB The FC for sensors FB8 SevenStateHP FC4 FB6 FALSE Checking BOOL FiveStateElectronic a FB CheckingDone BOOL FC200 LogicEngine S1_00_Main FUNCTION_BLOCK FB4 VAR_INPUT StateCom:INT;//State Command OpMode:INT; ResetFault:BOOL:=0;//Reset Fault END_VAR VAR_OUTPUT state : INT ; //status of the FB PDOWorkPos : BOOL := 0 ; //Move to Work Fault : INT := 0 ; END_VAR

Figura 48- Interface de mapeamento entre Funções e CLPs via CCE Mapper

Com o código entendível pelo CLP traduzido inicia-se o desenvolvimento para utilizar a interface escolhida. Neste estudo de caso optou-se pelo modelo de dados OPC UA para o monitoramento da planta. Foi escolhido esse modelo de dados, pois o CLP da marca SIEMENS e os softwares da mesma marca já estavam aptos a trabalhar com OPC UA e o dispositivo estava disponível para teste na planta.

Para o estudo de caso utilizou-se a o monitoramento do desempenho com a conectividade via protocolo OPC UA, com um CLP Siemens S7-300, mas poderia ser com qualquer outro CLP que possua capacidade de "entender" OPC UA. (Figura 49). O software fornecido pela Siemens foi atualizado para trabalhar com OPC UA via TCP/IP Ethernet.

SIEMENS

Figura 49- CLP Siemens S7-300 utilizado

Fonte: Autor

Foi definido que o componente segurança seria implementado e, para tal, um software OPC UA foi desenvolvido em C# (vide apêndice C). Para a escolha da biblioteca e linguagem para desenvolvimento do software servidor e cliente OPC UA uma análise das ferramentas disponíveis no mercado foi realizada e será mostrada ao final do estudo de caso.

O modelo de dados OPC UA permite o transporte de dados via diferentes protocolos de transporte, pode-se usar o SOAP-HTTP na porta 443 ou uma configuração binária e própria denominada OPC.TCP no desenvolvimento analisou-se as duas opções.

O servidor OPC UA monitora o CLP e as variáveis do processo, para isso define-se um IP e porta para acesso. Além disso, é necessário fazer o mapeamento das variáveis e serviços para o modelo de dados OPC UA. Para esta função utilizou-se o aplicativo UAModeler (figura 50), desenvolvido pela empresa Unified Automation. Com ele definiu-se o mapeamento entre as variáveis virtuais e reais e se criou os objetos do estudo de caso, tags, tipos

de variáveis e arquivo no padrão SOAP/WSDL que é gerado e lido pelo servidor OPC UA. Ver apêndice C.

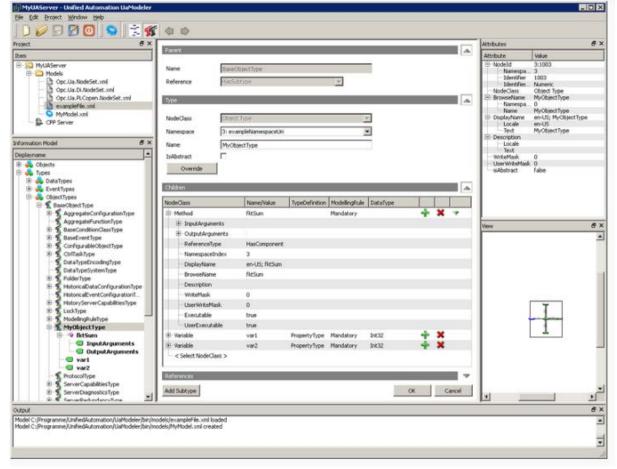


Figura 50 - Aplicativo UA Modeler

Fonte: Elaborado pelo autor

Para uma melhor experiência do usuário, decidiu-se usar uma ferramenta de descoberta de servidor OPC UA. Essa é uma possibilidade de pesquisa possível para localizar servidores: quando se insere um endereço, o sistema busca todos os servidores OPC UA em diferentes portas e os mapeia. Para o usuário isso simplificará o acesso, pois o sistema efetua uma busca pelos servidores OPC UA disponíveis.

Devido à grande importância da segurança para o meio industrial e seguindo o componente Segurança e autenticação da AMOS, utilizou-se duas camadas:

- Uma camada de certificado RSA com 128 bits que gera uma chave que deve ser armazenada entre o servidor e cliente e toda a comunicação entre cliente e servidor é criptografada. Essa camada é gerada pelo próprio modelo de dados OPC UA e cabe ao desenvolvedor habilitar essa funcionalidade.
- A segunda camada faz uso de autenticação por meio de usuário e senha para ter acesso ao software de monitoramento (figura 51). Essa estratégia de autenticação busca limitar o acesso de pessoas não-autorizadas. Esta parte do código foi criada usando uma interface com o software MySQL, que é um banco de dados open source e armazenou os dados referentes a este estudo de caso, vide apêndice C.

DET VEW PROJECT BUILD DEBUG TEAM SQ. TOOLS TEST ANALYSI VERNOUS PELLS

| CISHIP Publicates calculated by Teaching Sq. to the Committee of the

Figura 51- Tela do Aplicativo Cliente OPC desenvolvimento próprio

Fonte: Elaborado pelo autor

O padrão OPC UA define cada elemento como um objeto. Cada objeto tem propriedades e contém outros objetos. Foi representada toda essa estrutura para o usuário com uma árvore (figura 50). Cada nó representa um objeto para o servidor. O usuário pode encontrar facilmente e rapidamente os dados que desejam. Então, durante a conexão com o servidor, o cliente requisita, em tempo de execução, os nós da raiz da árvore,

e para cada objeto, faz-se o mesmo, sendo a estrutura criada em tempo de execução. Então, obtém-se assim uma referência para todo o nó da árvore.

Na janela superior esquerda da figura 52 vê-se a janela de navegação do cliente OPC. Nela escolhe-se qual servidor, qual equipamento, qual variável será monitorada. Na janela superior direita encontram-se os atributos do componente selecionado, e a partir daí, pode-se escolher se este atributo é relevante e deve ser adicionado a Janela de monitoramento. Para monitorar o usuário pode arrastar e soltar a *Tag* (Variável) na janela inferior. Também é possível alterar os intervalos de amostragem para estes itens, gravar valores e remover itens.

Endpoints são os dispositivos que estão sendo monitorados pelo modelo de dados OPC UA. Cada *endpoint* possui atributos (*Tags*) e valores para os atributos. Os atributos são as variáveis criadas pelo próprio modelo de dados OPC UA e também as variáveis definidas para o processo monitorado.

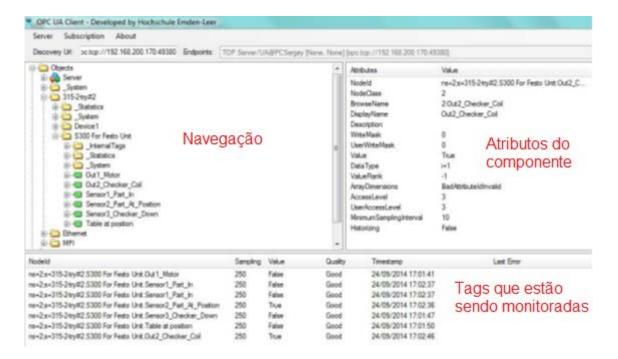


Figura 52- Tela do Aplicativo OPC Client

Fonte: Elaborado pelo autor

A interface do cliente exibe as informações atributos, permite gerenciar as *tags* e monitoramento *on-line* dos valores do processo (figura 52).

As varáveis monitoradas são exportadas para o banco de dados de acordo com a estrutura do software CCE, com isso a integração entre as ferramentas foi possível. Foi estabelecido um ciclo de que pode ser ajustado pelo usuário para gravação das variáveis e somente as variáveis que foram alteradas são enviadas, com isso reduz-se o tráfego da rede.

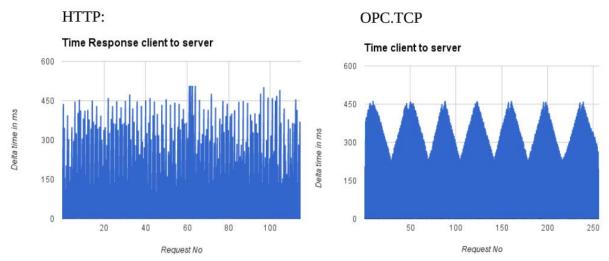
Testes:

Testou-se o OPC UA via SOAP-HTTP e via OPC.TCP. O primeiro é o mais compatível com outras tecnologias ou plataforma porque usa o protocolo HTTP, mas é mais lento porque usa um formato XML. O segundo é o mais rápido porque usa dados binários, que são dados compactados (comprimidos).

Para realizar os testes de monitoramento selecionou-se ele três valores (dois inteiros e um número real). Realizou-se mil solicitações entre o cliente e o servidor durante a análise. O tráfego da rede foi capturado e analisado com o software "Wireshark" (www.wireshark.org) que captura, analisa protocolos e provê ferramentas de análises e relatórios.

Buscou-se analisar o atraso do tempo de resposta entre o cliente e o servidor com os dois protocolos. Observou-se que há uma certa equivalência entre o tempo de resposta entre ambos os protocolos, mesmo o binário demandando o processamento para a descompactação do pacote de dados (figura 53).

Figura 53 - Comparação tempo de resposta HTTP e binário OPC.TCP

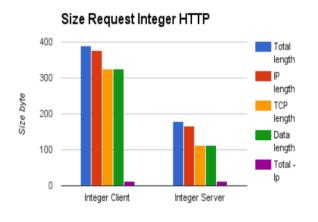


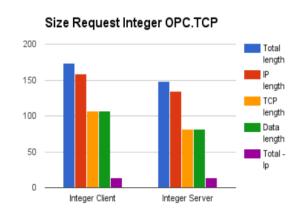
Fonte: Elaborado pelo Autor

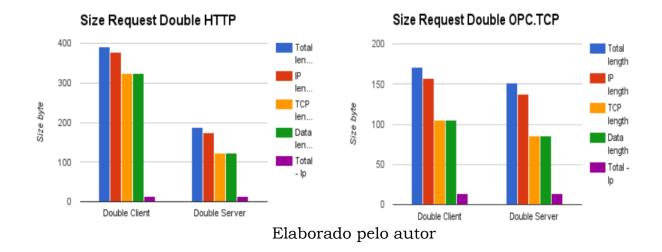
Outro teste realizado foi sobre o tamanho do pacote de dados utilizando os dois protocolos (figura 54). Observou-se que o tamanho do pacote de dados do protocolo binário OPC.TCP representa menos da metade de um pacote de dados HTTP, e não houve diferença significativa entre os tamanhos dos pacotes com uma variável numérica inteira e real.

Figura 54 – Comparação tamanho de pacote de dados entre HTTP e OPC.TCP









Validação: O Projeto foi validado por integrantes do grupo de pesquisa Engenharia de Sistemas de Automação da universidade de Warwick, Inglaterra e por integrantes do Instituto I2AR da Universidade de Emden-Leer na Alemanha.

A arquitetura e metodologia AMOS subsidiou o desenvolvimento desse componente de software para a integração entre os sistemas CCE e a planta do chão de fábrica, Festo Educacional. O componente segurança seguiu as diretrizes da arquitetura proposta, em duas camadas. A metodologia proposta facilitou o mapeamento realizado entre a estrutura real e a estrutura virtual, foram definidos os elementos, eventos e funcionalidades conforme a AMOS.

Conseguiu-se monitorar o processo a partir do cliente remotamente instalado. O objetivo do projeto futuramente é com os dados armazenados e integrados se conseguir fechar o ciclo de design, simulação, propotipagem, análise com algoritmos (inteligência artificial, *Big Data*, aprendizagem de máquina, etc.) monitoramento com diferentes aplicações clientes, entre elas dispositivos móveis e interface WEB.

Plano de melhoria e atualização:

Limitações encontradas:

- 1- A rede não era dedicada, foi utilizada a rede normal da universidade e com isso a latência da rede para acesso, leitura e processamento dos dados.
- 2- Uma atualização seria para um monitoramento da produção em série seria necessário se instalar um link dedicado.

Considerações sobre o estudo de caso

Para se estabelecer uma conexão via OPC UA, foi necessário realizar uma busca no mercado por soluções de sistemas pagos ou gratuitos que atendessem melhor a necessidade do projeto. Foram testadas diversas ferramentas e kits de desenvolvimento, A tabela 1 compara as ferramentas analisadas e mostra as opções disponíveis para realizar este estudo de caso. Optou-se por desenvolver um software OPC UA próprio, utilizando a biblioteca fornecida pela Unified Automation, pois as ferramentas tinham limitações de tempo de conexão ou número de usuários e ainda, por serem ferramentas proprietárias, não davam a liberdade necessária para a customização necessária ao desenvolvimento do projeto.

Tabela 01-Tabela comparativa sobre soluções em OPC UA

	Fabricante				
Características	Siemens	Software Toolbox	Matricon OPC	Unified Automation	
Aplicação Server	~	>	>		

Aplicação Client	~	~	✓	~
SDK* para Server			~	~
SDK* para Client			~	~
SDK* para desenvolvimento de sistemas embarcados			>	~
SDK* para 4 linguagens de programação (C++, Ansi C, .NET, Java)				~
Drivers para conectividade OPC SERVER para várias plataformas		•	~	
Licença <i>trial</i> gratuita para testes no lado servidor	~	~		
Licença <i>trial</i> gratuita para testes no lado cliente	~	~	~	~
SDK* com licença gratuita para desenvolvimento				~
Suporte e serviços online	~	~	~	~

*SDK – Standard Development Kit – Kit para desenvolvimento de sistemas

Um outro ponto relevante detectado pelo estudo de caso é a diferença entre o transporte de dados via HTTP e o protocolo binário OPC.TCP, sendo tamanho do pacote OPC.TCP menos da metade do HTTP. Isso impacta consideravelmente no tráfego da rede, e na integração entre sistemas. Uma solução que tem como foco a integração entre sistemas pode considerar o protocolo HTTP mesmo sendo mais pesado, tendo em vista sua facilidade de

se trabalhar com aplicativos *Web* e exportação sem a necessidade de descompactação. E para soluções legadas que já utilizam o protocolo OPC tradicional e onde o tempo de resposta e trafego de rede sejam restrições pode-se utilizar o protocolo binário OPC.TCP.

CAPÍTULO 6 - CONCLUSÃO

Pôde-se observar, ao longo da Tese, que a abordagem orientada a serviços promove interoperabilidade e uma forma aberta de interação entre os elementos de produção. Um dos maiores problemas no uso de serviços é a falta de aplicações industriais reais, sendo em sua maioria até hoje casos de uso realizados em laboratório e com determinado modelo de dados atrelado. Esta tese inova ao propor um modelo, arquitetura e metodologia voltada a soluções orientadas a serviço na manufatura, tendo sido aplicados com diferentes modelos de dados (OPC UA, gSOAP, REST e DPWS) em quatro estudos de caso.

A tese propôs utilizar os benefícios advindos da facilidade de integração, publicação, recuperação de informação em uma nova geração da automação industrial, buscando com a AMOS primeiramente estabelecer princípios e características para o desenvolvimento ou migração, modelo de integração e depois, com a arquitetura referência, definir o desenvolvimento, qual nível de complexidade e quais módulos ou componentes são adequados para a solução de casos concretos.

Esta proposta colabora para o amadurecimento da nova geração da automação industrial com foco na inserção de serviços na manufatura discreta inteligente, mas certamente poderá ter aplicação prática em outras áreas que utilizem e necessitem de processos distribuídos, cujo controle e monitoramento em tempo real seja crítico e realizado de forma distribuída.

A concepção, o desenvolvimento e implantação de processos industriais inteligentes e sistemas integrados de tecnologia de informação e comunicação está cada vez mais presente no chão de fábrica. Há no momento um processo de maturação desta tecnologia orientada a serviços, onde o protocolo de rede tende a ser baseado no padrão Ethernet, o que facilita a integração do chão de fábrica com a área e sistemas gerenciais.

Verifica-se a redução de custos de dispositivos "inteligentes", com maior capacidade de "sentir" o contexto, processar e interagir com outros elementos ou sistemas ao seu redor. Esta tese demonstra que a abordagem orientada a serviços pode ser aplicada na área industrial, mesmo com as limitações todavia expostas.

Pesquisas financiadas por grupos empresariais e diversos países buscam ser líderes neste mercado que tem um potencial bilionário, como descrito no capítulo 1, e por si só demonstram o potencial dessa nova abordagem. Esta tese buscou propor modelo, metodologia e arquitetura para implementar a orientação a serviços no chão de fábrica para que os próximos trabalhos venham a se beneficiar no desenvolvimento desta linha do conhecimento.

Espera-se, como resumo, os seguintes principais beneficios dessa nova visão de uma nova geração de automação industrial:

- Aumento da capacidade de decisão gerencial, pois com maior interoperabilidade, extrai-se informações de diversas fontes e combina-se com ferramentas como data analitics e Big data e indicadores para predição.
- Melhoria da resposta a eventos não previstos, como descrito na tese: pode-se configurar via serviços componentes com capacidades e que em caso de não-conformidades possam de forma autônoma adequar seus serviços oferecidos.
- Melhor utilização das máquinas industriais com a produção de produtos multivariados, pois com serviços é facilitado o setup e configuração de novos dispositivos.

6.1 - Contribuições científicas da Tese

Buscou-se com a Tese responder à hipótese: se seria possível se aplicar o paradigma orientado a serviços para que sistemas

heterogêneos inteligentes possam se comunicar de forma flexível e interoperável sem perda de desempenho.

A resposta pode ser dividida em duas partes:

- 1- Conectar sistemas heterogêneos de forma flexível interoperável: A tecnologia orientada a serviços e os estudos de caso realizados aumentaram a flexibilidade de se conectar diferentes sistemas mecatrônicos, pois utiliza-se baixo acoplamento entre os sistemas e a comunicação entre os elementos se realiza através de troca de mensagens e baseada em eventos. Utilizando servicos pode-se soluções customizadas. flexíveis ter automatizadas para a integração de produtos tecnológicos, buscando aplicar o conceito conecte e produza, com maior facilidade de configuração e replicação, com isso reduzindo de custos, e com um aumento da qualidade sobre a produção, pois tem-se mais informação disponível e mais facilmente integrada com outros sistemas.
- 2- Estabelecer comunicação sem perda de desempenho através de serviços: Neste quesito avaliou-se que a aplicação de SOA é limitada. pois existe tempo de empacotamento, um compactação/descompactação e processamento que atualmente é superior ao tempo de comunicação entre elementos que não se baseiam em serviços na automação industrial. O tempo médio varia de acordo com fatores como: o modelo de dados usado, quantidade de informação processada, distância entre elementos, porém para dispositivos críticos ou que necessitem resposta em tempo real ainda não há tecnologia no mercado disponível para tal fim. O protocolo OPC UA é promissor, pois é originado do mundo da automação industrial e é suportado por diversas grandes companhias. Porém ainda possui uma perda de desempenho superior às tecnologias tradicionais da automação industrial.

Esta tese oferece subsídios para que a academia e indústria possam aprofundar seus conhecimentos nos avanços e possibilidades provenientes das novas tecnologias e paradigmas de TICC (Tecnologia da Informação, controle e Comunicação), associado a uma nova visão da automação, para que possam baseado no presente trabalho avançar na pesquisa e desenvolvimento de soluções.

Além disso, a arquitetura de referência proposta facilita futuros projetos baseados em serviços com dispositivos inteligentes e SCFs, possibilitando uma solução modular e testada em estudos de caso com diferentes protocolos e linguagens de programação. Segue uma síntese das contribuições obtidas:

Contribuição	Descrição
Proposta de Arquitetura	Proposta permite às manufaturas, de
AMOS	forma modular e com uma visão
	integrada, a criação de soluções
	orientada a serviços, com diferentes
	portes, tendo em mente princípios da
	manufatura avançada e o conceito de
	conecte e produza, com uma visão
	baseada em sistemas reais e com um
	reflexo nas nuvens. A arquitetura tem
	base empírica e promove uma integração,
	respeitando a experiência e
	infraestrutura existentes, mas se
	adequando a nova realidade da indústria,
	que a cada dia evolui e desenvolve
	dispositivos com maiores capacidades de
	processamento e comunicação, e que
	podem ser monitorados e acessados
	diretamente via sistemas heterogêneos.

Proposta de metodologia Uma metodologia que auxilia as orientada a serviços um empresas a ter norte para desenvolver soluções para a manufatura avançada buscando simplificar e prover etapas necessárias para com isso implementar melhorias de seus processos, buscando as vantagens e o alinhamento com este novo modelo de produção e a manufatura avançada. Estudos de caso com Devido a ser um tema novo, poucos diferentes protocolos trabalhos ainda estão disponíveis para baseado em serviços acesso, esta tese desenvolveu estudos de casos onde a AMOS se adequou a diferentes plataformas e modelo de dados orientados a serviços. Combinando e analisando resultados, fornece subsídio para desenvolvedores que desejam se comunicar utilizando orientação de serviços. Foi realizada uma análise com as opções Comparativo entre Sistemas/SDK para disponíveis mercado no para desenvolvimento usando o desenvolvimento ou aquisição de Protocolo OPC UA soluções que trabalham com o protocolo OPC UA, e foi gerada uma tabela com as características das principais funcionalidades vantagens e auxiliará desvantagens. Isso pesquisadores que desejam desenvolver com o protocolo OPC UA. tese Modelo de comunicação e tem como uma de suas contribuições um modelo comunicação e

Integração	Integração que auxilia desenvolvedores e		
	integradores a visualizar de uma forma		
	abrangente o conceito proposto.		

6.2 - Limitações

Como a solução é baseada em tecnologias web, que derivam de um mundo onde as restrições de tempo real são mais suaves do que em uma planta industrial real, esta é uma limitação que carece de maiores testes e validação para avaliação dos impactos nos processos dependentes de controle em tempo real.

Por outro lado, acabando com a hierarquia tradicional dentro de um sistema de produção, ligando todos os dispositivos em serviços na nuvem, é possível se padronizar o modelo de dados e utilizarmos o padrão Ethernet para a troca de mensagens em rede.

Seguem algumas limitações detectadas ao longo do desenvolvimento da tese:

- 1- Apesar do interesse dos grandes fornecedores em promover o uso de serviços em seus produtos, existe uma carência de aplicações industriais reais, sendo em sua maioria até hoje casos de uso realizados em laboratório provenientes de projetos de pesquisa.
- 2- Devido à limitação de tempo não se inseriu conceitos de agentes inteligentes e ontologias, porém acredita-se que a combinação produza bons resultados.
- 3- Devido à necessidade de tratamento das mensagens, existe um processamento adicional pelo *parser* XML. Existe a limitação de tempo de aceite dos dados, processamento e retorno na troca de mensagens entre a comunicação via WebServices, onde esse atraso tem variado de

acordo com o modelo de dados e com o protocolo de transporte utilizado, tempo esse que na prática ainda é alto e inviabiliza a aplicação em determinadas máquinas de chão de fábrica, mas é um atraso tolerável em níveis gerenciais.

- 4- Embora o uso de protocolos abertos facilite a interoperabilidade, existe ainda uma disputa entre propostas que desejam ser o padrão orientado a serviços: OPC UA, DPWS, e gSoap são as mais utilizadas. Com isso outro dilema se cria, qual modelo de dados deve a solução seguir.
- 5- Existe a carência de novas ferramentas de engenharia, novas concepções de soluções integradas de sistemas que implementem serviços desde a concepção, com metodologias de integração entre fornecedores de maquinas industriais, sistemas de automação, TI e clientes de forma que tenhamos o mesmo nível de desempenho e robustez comparada com outros antigos paradigmas.

6.3 - Disseminação do Conhecimento

O trabalho desenvolvido no âmbito desta tese foi avaliado em várias publicações nacionais e internacionais. Os seguintes trabalhos de pesquisa foram aceitos e publicados:

Título	Autores	Publicação
New Technologies for	MORAES, E. C.;	Journal Advanced
Asset Management	TANAJURA, A. P.;	Materials Research, v.
Applied to Oil Fields	LEPIKSON, H. A	746, p. 581-587, ano
		2013
SISTEMAS DE	SOUZA, LEONARDO;	Congresso Norte
CONTROLE NA ÁREA	MORAES,EDUARDO.C.;	Nordeste de Pesquisa e

INDUSTRIAL:	LEPIKSON,HERMAN A.	Inovação, Salvador,
EVOLUÇÃO		BA, ano 2013
-		
Service-Oriented	MORAES, E. C.;	Journal Applied
Framework for Oil	LEPIKSON, H. A	Mechanics and
Fields Automation		Materials, v. 496, p.
		1438-1441, 2014
Potential Impact for	MORAES, E. C.;	Journal Advanced
Energy Consumption	LEPIKSON, HERMAN. A	Materials Research, v.
with Service Oriented		1030-1032, p. 1925-
Architecture		1929, ano 2014
Virtual Engineering	MORAES, EDUARDO;	IEEE 13th
Tool Integration with	LEPIKSON, HERMAN;	International
an Automation	KONSTANTINOV,	Conference on
System for Runtime	SERGEY; WERMANN,	Industrial Informatics
Monitoring through	JEFFREY; COLOMBO,	(INDIN), Ano 2015,
OPC-UA	ARMANDOW.; AHMAD,	Cambridge, Inglaterra
	BILAL ; HARRISON,	
	ROBERT	
Developing interfaces	MORAES, E. C.;	Lecture Notes in
based on services to	LEPIKSON, H. A,	Electrical Engineering
the cloud	COLOMBO, ARMANDO	Volume 349, 2015
manufacturing: Plug	W.	
and Produce		
A Service-Oriented	WERMANN, JEFFREY;	Lecture Notes in
Architecture	MORAES, E. C.;	Computer Science
Implementation in the	COLOMBO, ARMANDO	Volume 349, 2015
Digital Factory of the	W.	
University		

Impact of new ICT	WERMANN, JEFFREY;	Industrial Electronics	
trends for the	MORAES, E. C.;	Society, IECON 2015 -	
educational	COLOMBO, ARMANDO	41th Annual	
curriculum in the area	W.	Conference of the	
of Industrial		IEEE, ano 2015,	
Automation and		YOKOHAMA, Japão	
Engineering			
A indústria 4.0 e seus	MORAES, EDUARDO	Congresso Norte	
impactos na sociedade	C., Lepikson, Herman	Nordeste de Pesquisa e	
		Inovação, Rio Branco,	
		AC, ano 2015	
CPPM: A logical	MORAES, EDUARDO	International	
	MORAES, EDUARDO		
architecture for	LEPIKSON, HERMAN	Conference on Material	
Cyber-Physical		Science and	
Production Model		Engineering	
		Technology, 2016,	
		Phuket, Tailândia	

6.4 - Trabalhos Futuros

Os resultados são promissores e motivam o aprofundamento de pesquisas com resultados mais amplos e sustentáveis que tornem as novas tecnologias maduras e seguras e que possam se tornar produtos comerciais que venham a trazer beneficios reais a sociedade.

A partir da experiência e dos frutos propostos desta Tese, podem evoluir para uma serem aplicados a novas abordagens e estudos de caso provenientes da academia e indústria.

O conhecimento e a ciência não param, a tese tem duração limitada e a tecnologia deverá evoluir. Novos desafios são elencados e trabalhos futuros são propostos baseados/complementares aos resultados deste trabalho:

- 1- Estender a metodologia proposta para outras áreas envolvidas no desenvolvimento de um produto (design, manutenção, engenharia, treinamento) e análise de riscos e análise de viabilidade, ROI (retorno do investimento), riscos de projeto, fatores humanos, etc.
- 2- Investigar a aplicação de SOA e da abordagem proposta e sua aplicação em estratégias eficazes para gestão da produção como *Lean manufacturing* ou *time-to-market*.
- 3- Desenvolver um modelo de negócios para as indústrias trabalharem e serem geridas nas nuvens de forma remota (de forma terceirizada ou não) em um modelo "pay as you go", ou pague conforme o uso.
- 4- Avaliar soluções que incluam o uso de agentes inteligentes associado com a abordagem SOA.
- 5- Desenvolver ontologia com conhecimentos sobre o fluxo de operação e reconfigurabilidade, servindo como uma base de dados em caso de uma não conformidade ocorrer na operação.
- 6- Desenvolver ontologia de domínio para disseminar a metodologia modular proposta para acessos sistemas nas nuvens, acoplando este conhecimento sobre de sistemas heterogêneos a partir de metadados comum (perfis, níveis de acesso, alias entre as bases de dados, etc.).
- 7- Estender o estudo de caso vislumbrando o uso de ferramentas de *Big Data*, BI (*Business Intelligence*), mineração de dados, visualização em 3D com visualização em tempo real, e ferramentas de simulação.
- 8- Avaliar estratégias para elevar a segurança (real e dos dados) e a confiabilidade (humana e da máquina) em um ambiente de manufatura avançada.

- 9- Desenvolver uma solução híbrida ou um *gateway* entre os protocolos OPC UA e DPWS.
- 10- Integrar o modelo proposto a um sistema de fluxo (*workflow*) com interface Web, e adaptável a dispositivos móveis a fim de que se possa acompanhar as atividades das operações em tempo real.
- 11-Avaliar o impacto da adoção de um novo paradigma na automação industrial impacta na área social e ambiental, relativa ao número de postos de trabalho, produtividade, impacto ambiental.
- 12-Avaliar a proposta em outros setores que não sejam de manufatura, buscando-se analisar seus benefícios e limitações.

Conclui-se que a tecnologia orientada a serviço tem um fator de impacto positivo sobre a capacidade de uma empresa moderna de lidar de forma ágil com os requisitos mutáveis de sistemas e processos, manutenção de máquinas muitas vezes inesperadas, reduzindo o tempo de configuração e garantindo simultaneamente um desempenho competitivo e confiável. Porém como a tecnologia é recente, há a falta de produtos no mercado e exemplos em grande escala para mensurarmos até onde poderemos ir em termos de tempo de resposta, capacidade de autoresposta e autodecisão, entre outras questões continuam abertas com esse paradigma.

REFERÊNCIAS

ACATECH (2014)."acatech-Deutsche Akademie der Umsetzungsempfehlungen Technikwissenschaften (2013): für das Industrie 4.0." Zukunftsprojekt URL: www. plattformi40. de/sites/default/files/Abschlussbericht_Industrie4% 200_barrierefr ei. pdf. Abgerufen am **17**.

AESOP. (2014). "Architecture for service-oriented process." Retrieved 27 November, 2014, from http://www.imc-aesop.eu.

Ahmad, B., X. Kong, R. Harrison, J. Watermann and A. W. Colombo (2013). Automatic generation of Human Machine Interface screens from component-based reconfigurable virtual manufacturing cell. Industrial Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE.

Baheti, R. and H. Gill (2011). "Cyber-physical systems." The impact of control technology **12**: 161-166.

Bohn, H., A. Bobek and F. Golatowski (2006). **SIRENA - Service Infrastructure for Real-time Embedded Networked Devices: A service oriented framework for different domains**. Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, 2006. ICN/ICONS/MCL 2006. International Conference on.

Brettel, M., N. Friederichsen, M. Keller and M. Rosenberg (2014). "How virtualization, decentralization and network building change the manufacturing landscape: An industry 4.0 perspective." International Journal of Mechanical, Industrial Science and Engineering 8(1): 37-44.

Cândido, G., F. Jammes, J. Barata and A. W. Colombo (2009). **Generic Management Services for DPWS-enabled devices.** Industrial Electronics, 2009. IECON'09. 35th Annual Conference of IEEE, IEEE.

Cândido, G. M. (2013). "Service-oriented architecture for device lifecycle support in industrial automation."

Cannata, A., M. Gerosa and M. Taisch (2008). **SOCRADES: A framework for developing intelligent systems in manufacturing**. Industrial Engineering and Engineering Management, 2008. IEEM 2008. IEEE International Conference on, IEEE.

Cannata, A., M. Gerosa and M. Taisch (2008). A Technology Roadmap on SOA for smart embedded devices: Towards intelligent systems in manufacturing. Industrial Engineering and Engineering Management, 2008. IEEM 2008. IEEE International Conference on, IEEE.

Cassiolato, J. E. (2013). "As Políticas de ciência, tecnologia e inovação na China."

Chambers, J. (2014). Are you ready for the Internet of everything. World Econ Forum Blog.

CNI (2016). Desafios para Indústria 4.0 no Brasil. Brasilia. 1: 37.

Colombo, A. W., T. Bangemann, S. Karnouskos, S. Delsing, P. Stluka, R. Harrison, F. Jammes and J. L. M. Lastra (2014). **Industrial Cloud-Based Cyber-Physical Systems**, Springer.

Colombo, A. W. and R. Harrison (2008). "Modular and collaborative automation: achieving manufacturing flexibility and reconfigurability." International Journal of Manufacturing Technology and Management 14(3-4): 249-265.

Colombo, A. W., S. Karnouskos and T. Bangemann (2013). A system of systems view on collaborative industrial automation. Industrial Technology (ICIT), 2013 IEEE International Conference on, IEEE.

Colombo, A. W., R. Schoop, P. Leitao and F. Restivo (2004). **A collaborative automation approach to distributed production systems.** Industrial Informatics, 2004. INDIN '04. 2004 2nd IEEE International Conference on.

Commission, E. (2015). **Digital Transformation of European Industry and Enterprises**: A report of the Strategic Policy Forum on Digital Entrepreneurship. Brussels, Strategic Policy Forum on Digital Entrepreneurship.

Delsing, J., J. Eliasson, R. Kyusakov, A. W. Colombo, F. Jammes, J. Nessaether, S. Karnouskos and C. Diedrich (2011). **A migration approach towards a SOA-based next generation process control and monitoring**. IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society.

Economic, F. W. (2016). "What is the theme of Davos 2016?" What is the theme of Davos 2016? Retrieved 04/05/2016, 2016.

Evans, D. (2012). "The internet of everything: How more relevant and valuable connections will change the world." Cisco IBSG: 1-9.

Fernandes, C. (2012). "O que é cloud computing?", from http://www.techtudo.com.br/artigos/noticia/2012/03/o-que-e-cloud-computing.html.

Force, O. T. (1998). "**OPC overview.**" OPC Foundation.

Fremantle, P. (2014). "A Reference Architecture for the Internet of Things." WSO2 White Paper.

Gil, A. C. (1995). "**Como elaborar projetos de pesquisa**. São Paulo: Atlas, 1991." Métodos e técnicas de pesquisa social **5**.

Group, G. (2015). "Report." from http://www.gartner.com/newsroom/id/3114217.

Gubbi, J., R. Buyya, S. Marusic and M. Palaniswami (2013). "Internet of Things (IoT): A vision, architectural elements, and future directions." Future Generation Computer Systems 29(7): 1645-1660.

Heiler, S. (1995). "Semantic interoperability." ACM Computing Surveys (CSUR) 27(2): 271-273.

Jammes, F. and H. Smit (2005). **Service-oriented architectures for devices-the SIRENA view**. Industrial Informatics, 2005. INDIN'05. 2005 3rd IEEE International Conference on, IEEE.

Kagermann, H. (2015). **Change Through Digitization**—Value Creation in the Age of Industry 4.0. Management of Permanent Change, Springer: 23-45. Kagermann, H., J. Helbig, A. Hellinger and W. Wahlster (2013). **Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0**: Securing the Future of German Manufacturing Industry; Final Report of the Industrie 4.0 Working Group, Forschungsunion.

Karnouskos, S. and A. W. Colombo (2011). Architecting the next generation of service-based SCADA/DCS system of systems. IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society, IEEE. Knepper, R. A., T. Layton, J. Romanishin and D. Rus (2013). Ikeabot: An autonomous multi-robot coordinated furniture assembly system. Robotics and Automation (ICRA), 2013 IEEE International Conference on, IEEE.

Leitão, P., A. W. Colombo and S. Karnouskos (2016). "Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges." Computers in Industry 81: 11-25.

Leitão, P., V. Marik and P. Vrba (2013). "Past, present, and future of industrial agent applications." Industrial Informatics, IEEE Transactions on 9(4): 2360-2372.

Mahnke, W. (2010). "2.4 **OPC Unified Architecture**." Collaborative Process Automation Systems: 86.

Memória, C. V. (2015). "Incentivos para a inovação tecnológica: um estudo da renúncia fiscal no Brasil."

Monostori, L. (2014). "Cyber-physical production systems: roots, expectations and R&D challenges." Procedia CIRP 17: 9-13.

Moraes, E., H. Lepikson, S. Konstantinov, J. Wermann, A. W. Colombo, B. Ahmad and R. Harrison (2015). **Improving connectivity for runtime simulation of automation systems via OPC UA**. Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on, IEEE.

Nagorny, K., A. W. Colombo and J. Barata (2014). A survey of service-based systems-of-systems manufacturing systems related to product life-cycle support and energy efficiency. Industrial Informatics (INDIN), 2014 12th IEEE International Conference on, IEEE.

Nascimento, C. (2012). "**Web services com Restful e Soap**." from http://pt.slideshare.net/CCCarl/web-service-soap-vs-rest.

Park, Y. S., R. Harrison, D. Vera and J. Lee (2014). Improving fault diagnosis and accessibility in Manufacturing automation systems using **X3DOM**. Industrial Technology (ICIT), 2014 IEEE International Conference on, IEEE.

 ${
m Piazza,\ A.\ P.\ (2007).}$ "Uma abordagem para interoperabilidade Entre plataformas heterogêneas de serviços web para redes colaborativas de organizações."

Rathwell, G. (2001). "Design of plant control and information systems within an enterprise architecture." PERA—Standard documentation ISA S 95: 2001.

Reboredo, P. and M. Keinert (2013). **Integration of discrete manufacturing field devices data and services based on OPC UA**. Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE, IEEE.

Rivera, R., I. Teixeira, L. O. Reiff, C. E. Azen and D. d. S. Moreira (2014). "Política de Inovação no Complexo Eletrônico: o papel da Portaria 950/06 MCT." BNDES Setorial, Rio de Janeiro(39): 55-95.

Russell, J. D. and J. C. Fielding (2014). **America Makes**: The National Additive Manufacturing Innovation Institute (NAMII) Status Report and Future Opportunities (Postprint), DTIC Document.

Rüßmann, M., M. Lorenz, P. Gerbert, M. Waldner, J. Justus, P. Engel and M. Harnisch (2015). "Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries." Boston Consulting Group.

Santos, D. B. P., E. E. F. Barbosa and F. C. Automobiles (2015). "MANUFATURA DIGITAL NO PLANEJAMENTO DA AUTOMAÇÃO DA USINAGEM DE COMPONENTES POWERTRAIN." Blucher Engineering Proceedings 2(1): 531-537.

Schumacher, D. (2016). "Industry 4.0: How the New Interaction Between Man and Machine in Smart Factories Will Help Create More Intelligent Products in the Aerospace Industry."

Shipp, S. S., N. Gupta, B. Lal, J. A. Scott, C. L. Weber, M. S. Finnin, M. Blake, S. Newsome and S. Thomas (2012). **Emerging global trends in advanced manufacturing**, DTIC Document.

Sjøberg, D. I., B. Anda, E. Arisholm, T. Dybå, M. Jørgensen, A. Karahasanovic, E. F. Koren and M. Vokác (2002). **Conducting realistic experiments in software engineering**. Empirical Software Engineering, 2002. Proceedings. 2002 International Symposium n, IEEE.

Sommerville, I. and P. Sawyer (1997). Requirements engineering: a good practice guide, John Wiley & Sons, Inc.

Soriano, J., C. Heitz, H.-P. Hutter, R. Fernández, J. J. Hierro, J. Vogel, A. Edmonds and T. M. Bohnert (2013). **Internet of Services**. Evolution of Telecommunication Services, Springer: 283-325.

Turine, M. A. S. and P. C. Masiero (1996). "Especificação de requisitos: Uma introdução." Relatórios técnicos do ICMC-USP 39.

Van Engelen, R. and K. Gallivan (2002). **The gSOAP toolkit for web services and peer-to-peer computing networks**. Cluster Computing and the Grid, 2002. 2nd IEEE/ACM International Symposium on, IEEE.

Williams, T. (1998). "The Purdue enterprise reference architecture and methodology (PERA)." Handbook of life cycle engineering: concepts, models, and technologies 289.

Xu, X. (2012). "From cloud computing to cloud manufacturing." Robotics and computer-integrated manufacturing 28(1): 75-86.

Zachman, J. (1987). "A framework for information systems architecture." IBM systems journal **26**(3): 276-292.

APÊNDICE A

Modelo de planejamento de Migração

Equipe envolvida				
Nome	Cargo	Contato		

Este modelo busca subsidiar e fornecer informação suficiente para ajudar à tomada de decisão final, migrar ou não migrar. Com base neste planejamento estratégico, todas as métricas definidas, informação recolhida, levantamento de necessidades, deve proporcionar a uma empresa informação que permita responder a questões:

- Deve o projeto de migração ser abandonado, reduzido ou adiado?
- O modelo orientado a serviços é apropriado para o negócio?
- Após a análise da migração realizada, ela foi suficiente? Ou deve ser retomada e ser aprofundada?

Elementos de Produção envolvidos

Empresa:

Processo a se migrado:

Nome do Elemento (Hardware ou Software)	Descrição e funcionalidades	Trabalha orientado a serviços (Sim/Não)

Objetivos

Vantagens Esperadas	Limitações

O processo de migração identifica, analisa e controla os riscos envolvidos. Risco é a possibilidade de um acontecimento futuro e incerto; perigo. O risco alia-se à incerteza, que define o nosso conhecimento sobre eventos futuros e o seu desfecho (M.Talabis, et al., 2013).

Identificar Riscos

Busca identificar riscos de projeto, determinando quais riscos podem afetar o projeto e documenta suas características, suas causas.

Deve-se ater especial atenção para os riscos tecnológicos relativos à tecnologia e aos processos na migração para se trabalhar com orientação a serviços. Estes riscos derivam do estado da arte da tecnologia aplicada em relação ao projeto definido. Identificar e registrar os riscos que podem ser classificados como:

Riscos de requerimentos tecnológicos: referentes a incerteza de escopo, condições de uso, complexidade da tecnologia selecionada.

Riscos referentes a adequações tecnológicas: incluem uso de *gateways*, riscos relacionados com a maturidade e limites da tecnologia.

Riscos de aplicação tecnológicas: experiência da equipe e organização com a tecnologia e recursos físicos necessários para a tecnologia.

Deve-se listar os riscos e associar suas causas e áreas que afetam.

Causa

Analisar Riscos

A avaliação do risco é definido pela probabilidade da concretização de uma ameaça e determinando o resultado desse evento medindo o impacto negativo sobre os sistemas, dados ou informação.

Para a análise dos riscos, uma metodologia mista de qualitativa e quantitativa, baseada no Instituto de Gerenciamento de Projetos(PMI) e PMBOK(Guia de conjuntos de conhecimentos em gerenciamento de projetos) deve ser realizada para definir os indicadores de risco(IR); são classificados de acordo com a medida de risco de causa.

Medida de risco = Probabilidade X Impacto

Cada empresa deve definir quais os indicadores que se enquadram para o risco do processo em análise. Para cada risco elencado uma avaliação sobre seu nível de impacto (insignificante, baixo, moderado, alto, catastrófico) deve ser realizado. Após isso a probabilidade de ocorrência do risco deve ser mensurado (remoto, pouco provável, possível, provável e praticamente certo).

Com essa análise qualitativa realizada, ela deve ser transformada em uma análise quantativa e poder extrair informações de forma mais clara para a tomada de decisões.

Segue a tabela de medida de risco proposta.

1	Praticamente certo (5)	5	10	15	20	25
lade	Provável (4)	4	8	12	16	20
Probabilidade	Possível (3)	3	6	9	12	15
Prob	Pouco provável (2)	2	4	6	8	10
	Remoto (1)	1	2	3	4	5
·	rau de prioridade dos riscos	1	2	3	4	5
_	Alta prioridade	Insignifi- cante (1)	Baixo (2)	Moderado (3)	Alto (4)	Catastró- fico (5)
	Média prioridade Baixa prioridade	Magnitude do Impacto				

Tabela de análise de riscos

Risco	Probabilidade	Prioridade	Medida de risco

Após essa etapa tem-se uma classificação dos riscos e pode-se estabelecer uma prioridade em seu controle.

Controlar Riscos

Nesta etapa deve-se planejar e executar planos de redução de risco e avalia sua eficácia durante todo o ciclo do projeto.

Risco	Plano de Ação	Responsável

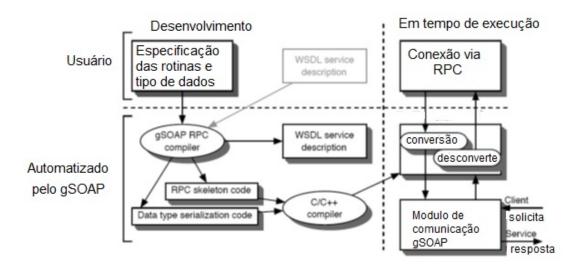
Cronograma

Atividade	Data Início	Data Fim	Responsável	Percentual Realizado

Apêndice B - Estudo de Caso Leitor RFID

Segue o diagrama de comunicação e código fonte e serviços codificados em WSDL/XML.

Figura 55- Fluxo de comunicação entre as funções definidas pelo usuário via gSOAP e outras funções internas do protocolo



Fonte: Autor

Descrição dos serviços do estudo de caso em WSDL/XML. Arquivo RFID.wsdl

```
1 <?xml version=" 1.0 " encoding="UTF-8" ?>
2 < d e f i n i t i o n s name=" RFIDservice "
3 targetNamespace=" h t t p : / /www. technik-emden . de / r f i d . wsdl "
4 xmlns:tns=" h t t p : / /www. technik-emden . de / r f i d . wsdl "
5 xmlns:SOAP-SOAP-Eh t t p : / / schemas . xmlsoap . org / soap / envelope / "
6 xmlns:SOAP-SOAP-ENC=" h t t p : / / schemxmlsoap . org / soap / encoding / "
7 xmlns:xsi=" h t t p : / /www.w3. org /2001/XMLSchema-instance"</pre>
```

```
8 xmlns:xsd=" h t t p : / /www.w3. org /2001/XMLSchema"
9 xmlns:ns=" u r n : r f i d "
10 xmlns:SOAP=" h t t p : / / schemas . xmlsoap . org / wsdl / soap / "
11 xmlns:HTTP=" h t t p : / / schemas . xmlsoap . org / wsdl / h t t p / "
12 xmlns:MIME=" h t t p : / / schemas . xmlsoap . org / wsdl / mime / "
13 xmlns:DIME=" h t t p : / / schemas . xmlsoap . org / ws/2002/04/ dime / wsdl / "
14 xmlns:WSDL=" h t t p : / / schemas . xmlsoap . org / wsdl / "
15 xmlns=" h t t p : / / schemas . xmlsoap . org / wsdl / ">
16
17 <types>
18
19 <schema targetNamespace=" u r n : r f i d "
20 xmlns:SOAP-ENV=" h t t p : / / schemas . xmlsoap . org / soap / envelope / "
21 xmlns:SOAP-ENC=" h t t p : / / schemas . xmlsoap . org / soap / encoding / "
22 xmlns:xsi=" h t t p : / /www.w3. org /2001/XMLSchema-instance "
23 xmlns:xsd=" h t t p : / /www.w3. org /2001/XMLSchema"
24 xmlns:ns=" u r n : r f i d "
25 xmlns=" h t t p : / /www.w3. org /2001/XMLSchema"
26 elementFormDefault=" u n q u a l i f i e d "
27 attributeFormDefault =" u n q u a l i f i e d ">
28 <import namespace=" h t t p : / / schemas . xmlsoap . org / soap / encoding / " / >
29 <complexType name=" i n t e r n a l -s t r u c t-tag ">< ! -- ns_internal_struct_tag -->
30
31 <sequence>
32 <element name=" epcStr " type=" x s d : s t r i n g " minOccurs=" 0 " maxOccurs=" 1 "
n i 11 a b 1 e = " true " / >< ! -- ns_internal_struct_tag::epcStr -->
33 <element name=" timeStr " type=" x s d : s t r i n g " minOccurs=" 0 " maxOccurs=" 1 "
```

```
n i l l a b l e =" true " / >< ! -- ns_internal_struct_tag::timeStr -->
34 <element name="memData-p t r " type=" ns:tag-memData" minOccurs=" 0 "
maxOccurs=" 1 " n i l l a b l e = " true " / >< ! --ns__internal_struct_tag::memData_ptr -->
35 <element name=" port " type=" x s d : i n t " minOccurs=" 1 " maxOccurs=" 1 " / ><! --
ns internal struct tag::port -->
36 <element name=" r s s i " type=" x s d : i n t " minOccurs=" 1 " maxOccurs=" 1 " / ><!--
ns__internal_struct_tag::rssi -->
37 < / sequence>
38 < / complexType>
39 <complexType name=" tag-memData"><!-- ns_tag_memData -->
40
41 <sequence>
42 <element name="memData" type=" ns:Array64OfunsignedByte " minOccurs=" 1 "
maxOccurs=" 1 " n i 11 a b 1 e =" true " / >< ! -- ns_tag_memData::memData -->
43 < / sequence>
44 < / complexType>
45 <complexType name=" UserDataInput "><!-- ns_UserDataInput -->
46
47 <sequence>
48 <element name=" t a r g e t " type=" x s d : s t r i n g " minOccurs=" 0 " maxOccurs=" 1 "
n illable="true"/><!--ns_UserDataInput::target-->
49 <element name=" data " type=" ns:Array64OfunsignedByte " minOccurs=" 1 "
maxOccurs=" 1 " n i l l a b l e =" true " / >< ! -- ns_UserDataInput::data -->
50 < / sequence>
51 < / complexType>
52 <complexType name=" Array64OfunsignedByte ">
53 < complexContent>
```

```
54 < r e s t r i c t i o n base="SOAP-ENC:Array">
55 < a t t r i b u t e r e f = "SOAP-ENC:arrayType " WSDL:arrayType=" xsd:unsignedByte
[]"/>
56 < /restriction >
57 < / complexContent>
58 < / complexType>
59 <!-- operation request element -->
60 <element name=" GetAntennaInfo ">
61 < complexType>
62 <sequence>
63 < / sequence>
64 < / complexType>
65 < / element>
66 < ! -- operation response element -->
67 <element name=" antenna-l i s t ">
68 < complexType>
69 <sequence>
70 <element name=" port-ant1 " type=" x s d : i n t " minOccurs=" 1 " maxOccurs=" 1 " /
><!-- ns__antenna_list::port_ant1 -->
71 <element name=" port-ant2 " type=" x s d : i n t " minOccurs=" 1 " maxOccurs=" 1 " /
><!--ns_antenna_list::port_ant2 -->
72 <element name=" port-ant3 " type=" x s d : i n t " minOccurs=" 1 " maxOccurs=" 1 " /
><!-- ns_antenna_list::port_ant3 -->
73 <element name=" port-ant4 " type=" x s d : i n t " minOccurs=" 1 " maxOccurs=" 1 " /
><!-- ns_antenna_list::port_ant4 -->
74 < / sequence>
75 < / complexType>
```

```
76 < / element>
77 <!-- operation request element -->
78 <element name=" ReadReaderInfo ">
79 <complexType>
80 <sequence>
81 < / sequence>
82 < / complexType>
83 < / element>
84 <!-- operation response element -->
85 <element name=" reader-i n f o ">
86 < complexType>
87 <sequence>
88 <element name=" model " type=" x s d : s t r i n g " minOccurs=" 0 " maxOccurs=" 1 "
n i l l a b l e =" true " / >< ! -- ns_reader_info::model -->
89 <element name=" s e r i a l " type=" x s d : s t r i n g " minOccurs=" 0 " maxOccurs=" 1 "
n i l l a b l e =" true " / >< ! -- ns_reader_info::serial -->
90 <element name=" u r i " type=" x s d : s t r i n g " minOccurs=" 0 " maxOccurs=" 1 "
n i l l a b l e =" true " / >< ! -- ns_reader_info::uri -->
91 <element name="hw-rev" type=" x s d : s t r i n g " minOccurs=" 0 " maxOccurs=" 1 "
n i l l a b l e =" true " / >< ! -- ns_reader_info::hw_rev -->
92 <element name="sw-rev" type=" x s d : s t r i n g " minOccurs=" 0 " maxOccurs=" 1 "
n i 1 1 a b 1 e = " true " / >< ! -- ns_reader_info::sw_rev -->
93 < / sequence>
94 < / complexType>
95 < / element>
96 < ! -- operation request element -->
97 <element name="ReadEPC">
```

```
98 < complexType>
99 <sequence>
100 < / sequence>
101 < / complexType>
102 < / element>
103 < ! -- operation response element -->
104 <element name=" s t r u c t-tag ">
105 < complexType>
106 < sequence>
107 <element name=" item " type=" n s : i n t e r n a l -s t r u c t-tag " minOccurs=" 0 "
maxOccurs=" unbounded " n i 11 a b 1 e =" true " / >< ! --
ns_struct_tag::_size -->
108 < / sequence>
109 < / complexType>
110 < / element>
111 <!-- operation request element -->
112 <element name=" WriteEPC ">
113 < complexType>
114 <sequence>
115 <element name=" t a r g e t " type=" x s d : s t r i n g " minOccurs=" 0 " maxOccurs=" 1 "
n illable="true"/><!--ns__WriteEPC::target-->
116 <element name=" input " type=" x s d : s t r i n g " minOccurs=" 0 " maxOccurs=" 1 "
n illable="true"/><!--ns__WriteEPC::input -->
117 </ sequence>
118 </ complexType>
119 < / element>
120 < ! -- operation response element -->
121 <element name=" WriteEPCResponse ">
122 < complexType>
123 <sequence>
124 <element name=" response " type=" x s d : i n t " minOccurs=" 0 " maxOccurs=" 1 "
```

```
n illable="true"/><!--ns__WriteEPC::response-->
125 </ sequence>
126 </ complexType>
127 </element>
128 < ! -- operation request element -->
129 <element name="ReadRSSI">
130 < complex Type>
131 <sequence>
132 </ sequence>
133 </ complexType>
134 </element>
135 <! -- operation request element -->
136 <element name=" ReadTime ">
137 < complexType>
138 <sequence>
139 </ sequence>
140 </ complexType>
141 </element>
142 < ! -- operation request element -->
143 <element name=" ReadUserData ">
144 < complex Type>
145 <sequence>
146 <element name=" t a r g e t " type=" x s d : s t r i n g " minOccurs=" 0 " maxOccurs=" 1 "
n illable="true"/><!--ns__ReadUserData::target-->
147 <element name=" single-tag-request " type=" x s d : i n t " minOccurs=" 1 "
maxOccurs=" 1 " / >< ! -- ns__ReadUserData::single_tag_request -->
148 </ sequence>
149 < / complexType>
150 < / element>
151 < ! -- operation request element -->
152 <element name=" WriteUserData ">
153 < complexType>
154 <sequence>
155 <element name=" input " type=" ns:UserDataInput " minOccurs=" 0 " maxOccurs
=" 1 " n i 11 a b 1 e =" true " / >< ! -- ns__WriteUserData::input -->
156 </ sequence>
157 </complexType>
158 < /  element>
159 < ! -- operation response element -->
160 <element name=" WriteUserDataResponse ">
161 < complexType>
162 <sequence>
163 <element name=" response " type=" x s d : i n t " minOccurs=" 0 " maxOccurs=" 1 "
```

```
n illable="true"/><!--ns__WriteUserData::response-->
164 </ sequence>
165 </ complexType>
166 </ element>
167 <! -- operation request element -->
168 <element name=" ReadAll ">
169 < complexType>
170 <sequence>
171 </ sequence>
172 </ complexType>
173 </ element>
174 < ! -- operation request element -->
175 <element name=" ChangeReadTime ">
176 < complex Type>
177 <sequence>
178 <element name=" time " type=" x s d : i n t " minOccurs=" 1 " maxOccurs=" 1 " />< ! ---
ns__ChangeReadTime::time -->
179 </ sequence>
180 </complexType>
181 < / element>
182 < ! -- operation response element -->
183 <element name=" signalResponse ">
184 < complex Type>
185 <sequence>
186 </ sequence>
187 </ complexType>
188 < /  element>
189 </schema>
190
191 </types>
192
193 <message name=" GetAntennaInfo ">
194 < part name=" Body " element=" ns:GetAntennaInfo " / ><! --
ns__GetAntennaInfo::ns__GetAntennaInfo -->
195 </message>
196
197 <message name=" antenna-l i s t ">
198 < part name=" Body " element=" ns:antenna-l i s t " / >< ! --
ns__GetAntennaInfo::response -->
199 </message>
201 <message name=" ReadReaderInfo ">
```

```
202 < part name=" Body " element=" ns:ReadReaderInfo " / >< ! --
ns__ReadReaderInfo::ns__ReadReaderInfo -->
203 </message>
204
205 <message name=" reader-i n f o ">
206 < part name=" Body " element=" ns:reader-i n f o " / ><! --
ns__ReadReaderInfo::response -->
207 </message>
208
209 <message name="ReadEPC">
210 < part name=" Body " element=" ns:ReadEPC " / >< ! -- ns_ReadEPC::ns_ReadEPC -->
211 </ message>
212
213 <message name=" s t r u c t-tag ">
214 < part name=" Body " element=" n s : s t r u c t -tag " / >< ! -- ns__ReadEPC::response -->
215 </message>
216
217 <message name=" WriteEPCRequest ">
218 < part name=" Body " element=" ns:WriteEPC " / >< ! -- ns __WriteEPC::ns __WriteEPC
219 </message>
220
221 <message name=" WriteEPCResponse ">
222 < part name=" Body " element=" ns:WriteEPCResponse " / >
223 </message>
224
225 <message name="ReadRSSI">
226 < part name=" Body " element=" ns:ReadRSSI " / >< ! -- ns_ReadRSSI::ns_ReadRSSI
-->
227 </message>
228
229 <message name=" ReadTime ">
230 < part name=" Body " element=" ns:ReadTime " / >< ! -- ns __ReadTime::ns __ReadTime
231 </message>
232
233 <message name=" ReadUserData ">
234 < part name=" Body " element=" ns:ReadUserData " / ><! --
ns__ReadUserData::ns__ReadUserData -->
235 </message>
237 <message name=" WriteUserDataRequest ">
```

```
238 < part name=" Body " element=" ns:WriteUserData " / ><! --
ns__WriteUserData::ns__WriteUserData -->
239 </message>
240
241 <message name=" WriteUserDataResponse ">
242 < part name=" Body " element=" ns:WriteUserDataResponse " / >
243 </message>
244
245 <message name=" ReadAll ">
246 < part name=" Body " element=" ns:ReadAll " / >< ! -- ns__ReadAll::ns__ReadAll -->
247 </message>
248
249 <message name=" ChangeReadTime ">
250 < part name=" Body " element=" ns:ChangeReadTime " / >< ! --
ns_ChangeReadTime::ns_ChangeReadTime -->
251 </ message>
253 <message name=" signalResponse ">
254 < part name="Body" element="ns:signalResponse"/><!--
ns__ChangeReadTime::out -->
255 </message>
256
257 <portType name=" RFIDservicePortType ">
258 < operation name=" GetAntennaInfo ">
259 <documentation>Service d e f i n i t i o n of f u n c t i o n ns_GetAntennaInfo</
documentation>
260 < input message=" tns:GetAntennaInfo " / >
261 <output message=" tns:antenna-l i s t " />
262 </operation>
263 < operation name=" ReadReaderInfo ">
264 <documentation>Service d e f i n i t i o n of f u n c t i o n ns_ReadReaderInfo</
documentation>
265 < input message=" tns:ReadReaderInfo " / >
266 <output message=" tns:reader-i n f o " / >
267 </ operation>
268 < operation name="ReadEPC">
269 < documentation > Service d e f i n i t i o n of f u n c t i o n ns_ReadEPC < / documentation
270 < input message=" tns:ReadEPC " />
271 < output message=" t n s : s t r u c t -tag " / >
272 </ operation>
273 < operation name=" WriteEPC ">
274 <documentation>Service d e f i n i t i o n of f u n c t i o n ns__WriteEPC</
```

```
documentation>
275 < input message=" tns:WriteEPCRequest " />
276 <output message=" tns:WriteEPCResponse " / >
277 </ operation>
278 < operation name="ReadRSSI">
279 < documentation > Service d e f i n i t i o n of f u n c t i o n ns_ReadRSSI < /
documentation>
280 < input message=" tns:ReadRSSI " / >
281 < output message=" t n s : s t r u c t -tag " / >
282 </operation>
283 < operation name=" ReadTime ">
284 <documentation>Service d e f i n i t i o n of f u n c t i o n ns__ReadTime</
documentation>
285 < input message=" tns:ReadTime " / >
286 <output message=" t n s : s t r u c t -tag "/>
287 </ operation>
288 < operation name=" ReadUserData ">
289 < documentation > Service de finition of function ns ReadUserData < /
documentation>
290 < input message=" tns:ReadUserData " / >
291 <output message="tns:struct-tag"/>
292 </operation>
293 < operation name=" WriteUserData ">
294 <documentation>Service d e f i n i t i o n of f u n c t i o n ns_WriteUserData</
documentation>
295 < input message=" tns:WriteUserDataRequest " / >
296 <output message=" tns:WriteUserDataResponse " / >
297 </operation>
298 < operation name=" ReadAll ">
299 <documentation>Service d e f i n i t i o n of f u n c t i o n ns_ReadAll< / documentation
300 < input message=" tns:ReadAll "/>
301 <output message=" t n s : s t r u c t -tag " / >
302 </operation>
303 < operation name=" ChangeReadTime ">
304 <documentation>Service d e f i n i t i o n of f u n c t i o n ns_ChangeReadTime</
documentation>
305 < input message=" tns:ChangeReadTime " / >
306 <output message="tns:signalResponse"/>
307 </operation>
308 
309
310 <br/> sinding name=" RFIDservice " type=" tns:RFIDservicePortType ">
```

```
311 <SOAP:binding s t y l e = " document " t r a n s p o r t = " h t t p : // schemas . xmlsoap . org / soap
/ h t t p "/>
312 < operation name=" GetAntennaInfo ">
313 <SOAP:operation soapAction=" "/>
314 < input >
315 < SOAP:body parts="Body" use="literal"/>
316 < / \text{ input} >
317 <output>
318 < SOAP:body parts="Body" use="literal"/>
319 </output>
320 </ operation>
321 < operation name=" ReadReaderInfo ">
322 <SOAP:operation soapAction=" "/>
323 < input >
324 < SOAP:body parts="Body" use="literal"/>
325 </iinput >
326 <output>
327 <SOAP:body parts=" Body " use=" 1 i t e r a 1 " / >
328 < / output> 329 < / operation>
330 < operation name="ReadEPC">
331 <SOAP:operation soapAction=" "/>
332 < input >
333 <SOAP:body parts=" Body " use=" literal"/>
334 < / input >
335 <output>
336 <SOAP:body parts=" Body " use=" l i t e r a l " / >
337 < / output>
338 </operation>
339 <operation name=" WriteEPC ">
340 <SOAP:operation soapAction=" "/>
341 < input >
342 <
SOAP:body parts=" Body " use=" 1 i t e r a l " / >
343 < / input >
344 <output>
345 <SOAP:body parts=" Body " use=" l i t e r a l " / >
346 < / output>
347 </ operation>
348 < operation name="ReadRSSI">
349 <SOAP:operation soapAction=" "/>
350 < input >
351 < SOAP:body parts="Body" use="literal"/>
352 < / input >
353 <output>
354 <SOAP:body parts=" Body " use=" literal"/>
```

```
355 < / output>
356 </operation>
357 <operation name=" ReadTime ">
358 <SOAP:operation soapAction=" "/>
359 < input >
360 < SOAP:body parts="Body" use="literal"/>
361 </iinput>
362 <output>
363 < SOAP:body parts="Body" use="literal"/>
364 < / output>
365 </ operation>
366 <operation name=" ReadUserData ">
367 <SOAP:operation soapAction=" "/>
368 < input >
369 < SOAP:body parts="Body" use="literal"/>
370 < / input >
371 <output>
372 < SOAP:body parts="Body" use="literal"/>
373 < / output>
374 </ operation>
375 < operation name=" WriteUserData ">
376 <SOAP:operation soapAction=" "/>
377 < input >
378 <SOAP:body parts=" Body " use=" literal"/>
379 </iinput >
380 <output>
381 <SOAP:body parts=" Body " use=" literal"/>
382 < / \text{ output} >
383 </ operation>
384 < operation name=" ReadAll ">
385 <SOAP:operation soapAction=" "/>
386 < input >
387 <SOAP:body parts=" Body " use=" literal"/>
388 < / input >
389 <output>
390 < SOAP:body parts="Body" use="literal"/>
```

```
391 </ output>
392 </operation>
393 < operation name=" ChangeReadTime ">
394 <SOAP:operation soapAction=" "/>
395 < input >
396 < SOAP:body parts="Body" use="literal"/>
397 < / input >
398 <output>
399 < SOAP:body parts="Body" use="literal"/>
400 < / \text{ output} >
401 </ operation>
402 < / binding>
403
404 <service name=" RFIDservice ">
405<\!\!documentation\!\!>\!\!gSOAP\;2.8.21\;generated\;service\;d\;e\;f\;i\;n\;i\;t\;i\;o\;n</\;documentation>
406 < port name=" RFIDservice " binding=" tns:RFIDservice ">
407 <SOAP:address 1 o c a t i o n = " 192.168.157.240 :1000 " />
408 < / port >
409 < / service >
410
411 < / definitions >
```

Apêndice C - Estudo de Caso Cloud Manufacturing

Insere-se aqui maiores detalhes sobre o estudo de caso cloud manufacturing.

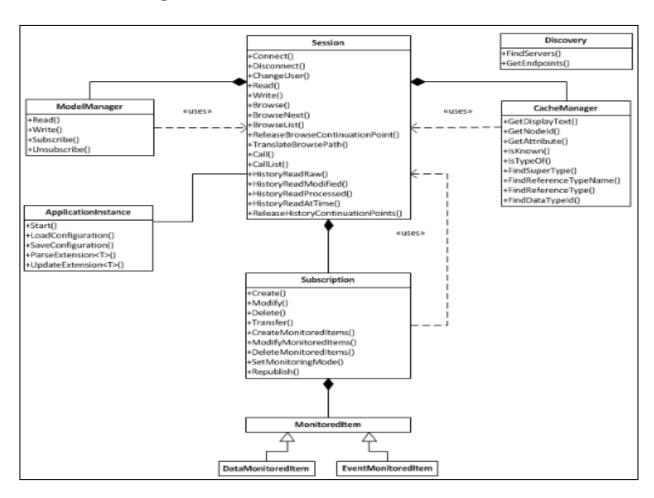
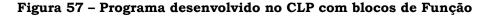


Figura 56 - Diagrama de Classe do estudo de caso



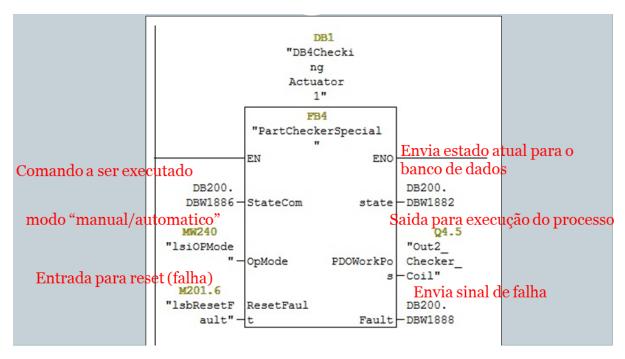
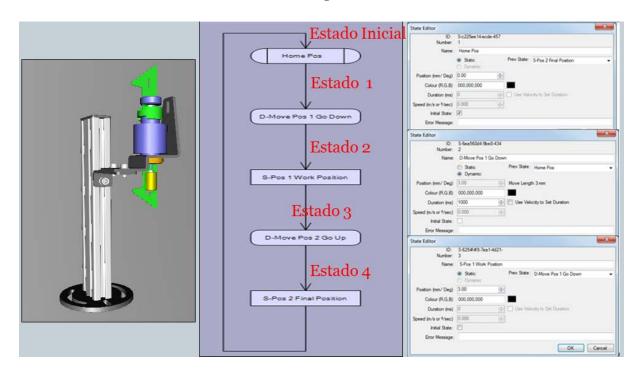


Figura 58 – Transição de estados do elemento e sua descrição no sistema em cada etapa



Descrição de um arquivo XML dos serviços da unidade de furação

```
<root nod_count="5" poco_xml_version="1.0">
                       e_id="C-643107a9-b02d-4d8f-ae29-3617a88206d5"
 <cmp
e nam="Pallet lock">
  <e_prt />
  <e_typ>cmp</e_typ>
  <e_chd>
   <cns e_id="CNS_1" e_nam="CNS_1">
    <e_prt>C-643107a9-b02d-4d8f-ae29-3617a88206d5</e_prt>
    <e_typ>cns</e_typ>
    <e chd>
      <geo0 e_id="GEO_1" e_nam="GEO_ Pallet lock">
       <e_prt>CNS_1</e_prt>
       <e_typ>geo</e_typ>
       <e_flh>False</e_flh>
       <e_vis>True</e_vis>
       <e_tch>True</e_tch>
       <e_gtp>0</e_gtp>
       <e_tra>0 0 0</e_tra>
       <e_rot>0 0 0</e_rot>
       <e_ctr>0 0 0</e_ctr>
       <e_geu>ASD\VRML Production line\General CAD\Stoppers\Pallet
lock.wrl</e_geu>
      </geo0>
      <lnk00 e_id="lnk_ypumxntrtr" e_nam="LNK_2_Conveyor">
       <e_prt>CNS_1</e_prt>
       <e_typ>lnk</e_typ>
       <e flh>False</e flh>
```

```
<e_vis>True</e_vis>
 <e_tch>True</e_tch>
 <e_sca>1</e_sca>
 <e_pos>-1397 -212 1224</e_pos>
 <e_nor>0 0 1</e_nor>
 e_{agl}>0</e_{agl}>
 <e_lvl>0</e_lvl>
 <e_occ>0</e_occ>
 <e_mat />
 <e_chd />
 <e_tra>0 0 0</e_tra>
 <e_rot>0 1 0 0</e_rot>
</lnk00>
<lnk10 e_id="lnk_iudsxsnuln" e_nam="LNK_1_2">
 <e_prt>CNS_1</e_prt>
 <e_typ>lnk</e_typ>
 <e_flh>False</e_flh>
 <e_vis>True</e_vis>
 <e_tch>True</e_tch>
 <e_sca>1</e_sca>
 <e_pos>0 0 0</e_pos>
 <e_nor>0 1 0</e_nor>
 <e_agl>0</e_agl>
 <e_lvl>1</e_lvl>
 <e_occ>0</e_occ>
 <e_mat />
 <e_chd />
 <e_tra>0 0 0</e_tra>
 <e_rot>0 1 0 0</e_rot>
```

```
</e_chd></e_chd></e_flh>False</e_flh></e_vis>True</e_vis></cns></e_chd></e_chd></e_flh>False</e_flh></e_vis>True</e_vis></cmp></root>
```

Código fonte do Cliente OPC

```
browseControl.SelectionChanged
BrowseControl.SelectionChangedEventHandler(browserControl SelectionCha
nged);
                        // Register for the Node Activated event of BrowseControl in order to
add item for monitoring in monitoredItemsControl
                       browseControl.NodeActivated
                                                                                                                                                                                                                           new
BrowseControl.NodeActivatedEventHandler(browserControl_NodeActivated);
                        // Register for the update statuslabel event of AttriuteListControl in
order to update
                       // the status label.
                       attributeListControl.UpdateStatusLabel +=
AttributeListControl.UpdateStatusLabelEventHandler(UserControl_UpdateSt
atusLabel);
                                                                              for
                                                                                                                         update
                                                                                                                                                         statuslabel
                                                                                                                                                                                                                                  of
                                                                                                   the
                                                                                                                                                                                                      event
MonitoredItemsControl in order to update
                       // the status label.
                       monitoredItemsControl.UpdateStatusLabel +=
                               new
Monitored Items Control. Update Status Label Event Handler (User Control\_Update Status Label Event Handler) and the Control\_Update Status Label Event Handler (User Control\_Update Status Label Event Handler). The Control\_Update Status Label Event Handler (User Control\_Update Status Label Event Handler) and the Control\_Update Status Label Event Handler (User Control\_Update Status Label Event Handler). The Control\_Update Status Label Event Handler (User Control\_Update Status Label Event Handler) and the Control\_Update Status Label Event Handler (User Control\_Update Status Label Event Handler). The Control\_Update Status Label Event Handler (User Control\_Update Status Label Event Handler) and the Control\_Update Status Label Event Handler (User Control\_Update Status Label Event Handler) and the Control\_Update Status Label Event Handler (User Control\_Update Status Label Event Handler) and the Control\_Update Status Label Event Handler (User Control\_Update Status Label Event Handler) and the Control\_Update Status Label Event Handler (User Control\_Update Status Label Event Handler) and the Control Event Handler (User Control Event Handler) and the Control Event Handler (User Control Event Handle
eStatusLabel);
                       m_timer = new Timer();
                       m_timer.Tick += new EventHandler(TimerElapsed);
                       m timer.Interval = 5000;
                       m_timer.Enabled = true;
               }
               #endregion
                #region Fields
```

```
/// <summary>
/// Provides access to the OPC UA server and its services.
/// </summary>
private Session m_Session = null;
/// <summary>
/// Provides access to the OPC UA server and its services.
/// </summary>
private ApplicationInstance m_Application = null;
/// <summary>
/// Flag indicates if the connection is successfully established or not.
/// </summary>
private bool m_bConnected = false;
/// <summary>
/// Timer to monitor if the license has expired
/// </summary>
private Timer m_timer = null;
#endregion
#region Properties
/// <summary>
/// Provides the text of the selected item of the combobox.
/// </summary>
public string ServerURL
  get { return UrlCB.Text; }
}
/// <summary>
/// Provides the status label toolstrip.
/// </summary>
```

```
public System. Windows. Forms. Tool Strip Status Label Status Label
                    {
                              get { return toolStripStatusLabel; }
                    }
                    #endregion
                    #region Calls to Client API
                     /// <summary>
                     /// Connect to server.
                    /// </summary>
                    private int Connect()
                              if (m_Session == null)
                               {
                                        m_Session = new Session(m_Application);
                                        m_Session.UseDnsNameAndPortFromDiscoveryUrl = true;
                                         // Attach to events
                                        m\_Session.SessionValidationError
                                                                                                                                                                                                                                                                                                new
SessionValidationEventHandler(Session_SessionValidationError);
                                        m_Session.ConnectionStatusUpdate
                                                                                                                                                                                                                                                                                                new
Server Connection Status Update Event Handler (Session\_Server Connection Status Update Event Handler (Session Server Update Event Handler (Session Server Update Event Handler
usUpdate);
                              }
                               // Check the content of the combobox.
                              if( UrlCB.Text.Length == 0 )
                                        return -1;
```

```
// Set wait cursor.
Cursor = Cursors.WaitCursor;
int result = 0;
try
{
  string endpointUrl;
  // Extract Url from combobox text.
  object item = UrlCB.SelectedItem;
  if ((item == null) | (item.GetType() == typeof(string)))
  {
     // The URL has been entered as text.
     endpointUrl = UrlCB.Text;
     // Call connect with URL
     m_Session.Connect(endpointUrl, SecuritySelection.None);
  }
  else
  {
     // The endpoint was provided through discovery.
     EndpointWrapper endpoint = (EndpointWrapper)item;
     // Call connect with endpoint
     m_Session.Connect(endpoint.Endpoint, null);
}
catch (Exception e)
```

```
result = -1;
          // Update status label.
          StatusException se = e as StatusException;
          if (se != null)
             toolStripStatusLabel.Text = String.Concat("Connect failed. Error
[", se.StatusCode.ToString(), "] ", e.Message);
          }
          else
             toolStripStatusLabel.Text = "Connect failed. Error: "
e.Message;
          toolStripStatusLabel.Image
global::UnifiedAutomation.Sample.Properties.Resources.error;
        }
        // Set default cursor.
        Cursor = Cursors.Default;
        return result;
     }
     /// <summary>
     /// Disconnect from server.
     /// </summary>
     private void Disconnect()
```

```
try
          // Call the disconnect service of the server.
         m_Session.Disconnect(SubscriptionCleanupPolicy.Delete, null);
       }
       catch (Exception exception)
          // Update status label.
          toolStripStatusLabel.Text = "Disconnect failed. Error:
exception.Message;
          toolStripStatusLabel.Image
global::UnifiedAutomation.Sample.Properties.Resources.error;
       }
     }
     #endregion
     #region User Actions
     /// <summary>
     /// Callback of the exception thrown event of BrowseControl and
AttributeListControl.
     /// </summary>
     /// <param name="node">The source of the event.</param>
     private void UserControl_UpdateStatusLabel(string strMessage, bool
bSuccess)
     {
       toolStripStatusLabel.Text = strMessage;
       if (bSuccess == true)
```

```
toolStripStatusLabel.Image
global::UnifiedAutomation.Sample.Properties.Resources.success;
       else
          toolStripStatusLabel.Image
global::UnifiedAutomation.Sample.Properties.Resources.error;
     }
     /// <summary>
     /// Callback of the selection changed event of BrowseControl.
     /// </summary>
     /// <param name="node">The source of the event.</param>
     private void browserControl_SelectionChanged(TreeNode node)
     {
       // Read all the attributes of the selected tree node.
       attributeListControl.ReadAttributes(node);
     }
     private void browserControl_NodeActivated(NodeId activatedNode)
       // Add monitored items
       monitoredItemsControl.addMonitoredItem(activatedNode);
     }
     /// <summary>
     /// Expands the drop down list of the ComboBox to display available
servers and endpoints.
     /// </summary>
```

```
/// <param name="sender">The source of the event.</param>
     /// <param name="e">The <see cref="System.EventArgs"/> instance
containing the event data.</param>
     private void UrlCB_DropDown(object sender, EventArgs e)
       // get discovery url
       string sUrl;
       // Check the text property of the Server textbox
       if (NodeTB.Text.Length == 0)
       {
          // Set the uri of the local discovery server by default.
          sUrl = "opc.tcp://localhost:4840";
       }
       else
          // Has the port been entered by the user?
          char seperator = ':';
          string[] strPortCheck = NodeTB.Text.Split(seperator);
          if (strPortCheck.Length > 1)
          {
            sUrl = NodeTB.Text;
          }
          else
          {
            sUrl = NodeTB.Text + ":4840";
```

}

```
// Set wait cursor.
       Cursor = Cursors.WaitCursor;
        // Clear all items of the ComboBox.
        UrlCB.Items.Clear();
       UrlCB.Text = "";
        // Look for servers
       List<ApplicationDescription> serverList = null;
       Discovery discovery = new Discovery(m_Application);
       try
          serverList = discovery.FindServers(sUrl);
       }
       catch (Exception exception)
       {
          // Update status label.
          toolStripStatusLabel.Text = "FindServers
                                                               failed:"
exception.Message;
          toolStripStatusLabel.Image
global::UnifiedAutomation.Sample.Properties.Resources.error;
          // Set default cursor.
          Cursor = Cursors.Default;
          return;
       }
       bool bGetEndpointsError = false;
       bool bIgnoreError = false;
```

```
string errorText = "";
       List<EndpointWrapper>
                                     lstEndpointWrappers
                                                                        new
List<EndpointWrapper>();
       // Populate the drop down list with the endpoints for the available
servers.
       foreach (ApplicationDescription server in serverList)
               (server.ApplicationType
          if
                                                ApplicationType.Client
                                                                          server.ApplicationType == ApplicationType.DiscoveryServer)
            continue;
          }
          try
            StringCollection lstEndpoint = new StringCollection();
            foreach (string discoveryUrl in server.DiscoveryUrls)
            {
               // Call GetEndpoints
               List<EndpointDescription> lstEndpoints = null;
               try
                  lstEndpoints = discovery.GetEndpoints(discoveryUrl);
                  foreach (EndpointDescription endpoint in lstEndpoints)
                  {
                    // build display name for combo
```

```
EndpointWrapper
                                            endpointWrap
                                                                       new
EndpointWrapper(endpoint);
                    if (!lstEndpoint.Contains(endpointWrap.ToString()))
                       lstEndpointWrappers.Add(endpointWrap);
                       lstEndpoint.Add(endpointWrap.ToString());
                       bIgnoreError = true;
               catch (Exception)
                 continue;
          catch (Exception exception)
          {
            // Update status label.
            errorText = "GetEndpoints failed. Error: " + exception.Message;
            bGetEndpointsError = true;
          }
       }
       // error occured during get endpoints
       if (bGetEndpointsError && !bIgnoreError)
       {
          // Update status label.
```

```
toolStripStatusLabel.Text = "GetEndpoints failed. Error: "
errorText;
         toolStripStatusLabel.Image
global::UnifiedAutomation.Sample.Properties.Resources.error;
       // add list of endpoints
       else
         UrlCB.Items.AddRange(lstEndpointWrappers.ToArray());
       }
       // Set default cursor.
       Cursor = Cursors.Default;
    }
    /// <summary>
    /// Handles the connect procedure being started from the menu bar.
    /// <summary>
    /// <param name="sender">The source of the event.</param>
    /// <param name="e">The <see cref="System.EventArgs"/> instance
containing the event data.</param>
    private void connectDisconnectTriggered(object sender, EventArgs e)
    {
       // Currently connected -> disconnect.
       if (m bConnected)
         Disconnect();
       // Currently not connected -> connect to server.
       else
```

```
{
         Connect();
       }
    }
    /// <summary>
    // Handles the publishing interval procedure started from the menu
bar.
    /// <summary>
    /// <param name="sender">The source of the event.</param>
    /// <param name="e">The <see cref="System.EventArgs"/> instance
containing the event data.</param>
    private void PublishingInterval_Click(object sender, EventArgs e)
    {
       if(monitoredItemsControl.Subscription != null)
       {
         PublishingIntervalDialog
                                            dlg
                                                                     new
PublishingIntervalDialog(monitoredItemsControl.Subscription);
         dlg.Show();
       }
    }
    /// <summary>
    /// Change the enabled state for the subscription
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void publishingEnabled_Click(object sender, EventArgs e)
    {
```

```
monitoredItemsControl.PublishingEnabled
publishingEnabledToolStripMenuItem.Checked;
       monitoredItemsControl.UpdateSubscription();
     }
     #endregion
     #region Event Handler
     /// <summary>
     ///
     /// </summary>
     private void Session_ServerConnectionStatusUpdate(Session sender,
ServerConnectionStatusUpdateEventArgs e)
     {
       if (this.InvokeRequired)
       {
          this.BeginInvoke(new
ServerConnectionStatusUpdateEventHandler(Session_ServerConnectionStat
usUpdate), sender, e);
         return;
       }
       // check that the current session matches the session that raised
the event.
       if (!Object.ReferenceEquals(m_Session, sender))
         return;
       }
       lock (this)
       {
```

```
bool bClearControls = false;
          switch (e.Status)
          {
            case ServerConnectionStatus.Disconnected:
               m_bConnected = false;
               // Update Button
               ConnectDisconnectBTN.Text = "Connect";
               // Update ToolStripMenu
               connectToolStripMenuItem.Enabled = true;
               disconnectToolStripMenuItem.Enabled = false;
               // Set enabled state for combobox
               UrlCB.Enabled = true;
               // Update status label.
               toolStripStatusLabel.Text = "Disconnected";
               toolStripStatusLabel.Image
global::UnifiedAutomation.Sample.Properties.Resources.warning;
               // clear controls
               bClearControls = true;
               break;
            case ServerConnectionStatus.Connected:
               m_bConnected = true;
               // Update Button
               ConnectDisconnectBTN.Text = "Disconnect";
               // Update ToolStripMenu
               connectToolStripMenuItem.Enabled = false;
               disconnectToolStripMenuItem.Enabled = true;
               // Set enabled state for combobox
               UrlCB.Enabled = false;
```

```
// Aggregate the UserControls.
               browseControl.Session = m_Session;
               attributeListControl.Session = m_Session;
               monitoredItemsControl.Session = m_Session;
               // Update status label.
               toolStripStatusLabel.Text
                                                "Connected
m_Session.EndpointDescription.EndpointUrl;
               toolStripStatusLabel.Image
global::UnifiedAutomation.Sample.Properties.Resources.success;
               // Browse first level.
               browseControl.Browse(null);
               break;
            case
ServerConnectionStatus.ConnectionWarningWatchdogTimeout:
               // Update status label.
               toolStripStatusLabel.Text = "Watchdog timed out";
               toolStripStatusLabel.Image
global::UnifiedAutomation.Sample.Properties.Resources.warning;
               break;
            case ServerConnectionStatus.ConnectionErrorClientReconnect:
               // Update status label.
               toolStripStatusLabel.Text = "Trying to reconnect";
               toolStripStatusLabel.Image
global::UnifiedAutomation.Sample.Properties.Resources.warning;
               break;
            case ServerConnectionStatus.ServerShutdownInProgress:
               // Update status label.
               toolStripStatusLabel.Text = "Server is shutting down";
```

```
toolStripStatusLabel.Image
global::UnifiedAutomation.Sample.Properties.Resources.warning;
               break;
            case ServerConnectionStatus.ServerShutdown:
               // Update status label.
               toolStripStatusLabel.Text = "Server has shut down";
               toolStripStatusLabel.Image
global::UnifiedAutomation.Sample.Properties.Resources.warning;
               break;
            case ServerConnectionStatus.SessionAutomaticallyRecreated:
               // Update status label.
               toolStripStatusLabel.Text = "A new Session was created";
               toolStripStatusLabel.Image
global::UnifiedAutomation.Sample.Properties.Resources.success;
               // clear controls
               bClearControls = true;
               break;
            case ServerConnectionStatus.Connecting:
               // Update status label.
               toolStripStatusLabel.Text = "Trying to connect to
((m_Session.EndpointDescription
m_Session.EndpointDescription.EndpointUrl: "<unknown>");
               toolStripStatusLabel.Image
global::UnifiedAutomation.Sample.Properties.Resources.warning;
               break;
            case ServerConnectionStatus.LicenseExpired:
               // Update status label.
               toolStripStatusLabel.Text = "The license has expired.";
               toolStripStatusLabel.Image
global::UnifiedAutomation.Sample.Properties.Resources.warning;
               // disable GUI
```

```
subscriptionToolStripMenuItem.Enabled = false;
               ConnectDisconnectBTN.Enabled = false;
               UrlCB.Enabled = false;
               bClearControls = true;
               break;
          }
          if (bClearControls)
             // Cleanup attribute list.
            this.attributeListControl.AttributeList.Items.Clear();
             // Cleanup treeview.
            browseControl.BrowseTree.BeginUpdate();
            browseControl.BrowseTree.Nodes.Clear();
            browseControl.BrowseTree.EndUpdate();
             // Aggregate the UserControls.
            browseControl.Session = null;
            attributeListControl.Session = null;
            monitoredItemsControl.Clear();
          }
       }
     }
                        Session_SessionValidationError(Session
     private
                void
                                                                     sender,
SessionValidationEventArgs e)
                                                                         232
```

serverToolStripMenuItem.Enabled = false;

```
if (InvokeRequired)
          Invoke(new
SessionValidationEventHandler(Session_SessionValidationError), sender, e);
          return;
       }
       // check that the current session matches the session that raised
the event.
       if (!Object.ReferenceEquals(m_Session, sender))
          return;
       }
       if (e.Error == StatusCodes.BadCertificateHostNameInvalid)
       {
          e.Ignore = true;
       }
     }
     private void TimerElapsed(object sender, EventArgs e)
       if(m_Session
                      !=
                           null
                                   &&
                                         m_Session.ConnectionStatus
ServerConnectionStatus.LicenseExpired)
          // show dialog once only - so we just stop the timer
          m_timer.Enabled = false;
          MessageBox.Show("The demo license has expired.\nRestart the
application.",
                    "License
                                   expired",
                                                   MessageBoxButtons.OK,
MessageBoxIcon.Warning);
```

Glossário

Nome	Definição
Abstração	Capacidade de se modelar determinado objeto de forma a se extrair dele suas características principais e funcionalidades, de forma a ser possível se digitalizar esse componente, inserindo-o em um sistema computacional
Alto acoplamento	Forte dependência entre componentes, dificil adicionar funcionalidades ao componente sem alterar consideravelmente seu código
Alta Coesão	Capacidade de reaproveitamento de código, quanto maior a coesão mais fácil a manutenção dos componentes que são muito interligados. Menor complexidade. Busca-se alta coesão. Objetos e componentes são especializados podem fazer e bem determinada tarefa.
Arquitetura	A arquitetura de software de um sistema
(Computacional)	consiste na definição dos componentes de software, suas propriedades externas, e seus relacionamentos com outros softwares. O termo também se refere à documentação da arquitetura de software do sistema. A arquitetura é centrada na ideia da redução da complexidade através da abstração e separação de interesses.
Broadcast	

	Broadcast é o envio informação a todos os
	computadores de um segmento de rede ao
	mesmo tempo. Esta mensagem é padronizada
	e todos os computadores recebem e
	normalmente um servidor (DHCP) retorna
	essa mensagem com a confirmação do
	recebimento e dados para a configuração do
	dispositivo.
Dispositivo	Um sensor, atuador, tag, elementos de
	produção. O dispositivo pode processar
	informações de contexto dos sinais ou de
	outros dispositivos e transmite informações
	selecionadas para outros. Além disso,
	dispositivos podem atuador no ambiente.
Firewall	Firovoll á uma acluação do compresso has codo
rirewall	Firewall é uma solução de segurança baseada
	em hardware ou software (mais comum) que,
	a partir de um conjunto de regras ou
	instruções, analisa o tráfego de rede para
	determinar quais operações de transmissão
	ou recepção de dados podem ser executadas.
	"Parede de fogo", a tradução literal do nome,
	rarede de 1050, a tradação interar do nome,
	já deixa claro que o firewall se enquadra em
	já deixa claro que o firewall se enquadra em
	já deixa claro que o firewall se enquadra em uma espécie de barreira de defesa. A sua
	já deixa claro que o firewall se enquadra em uma espécie de barreira de defesa. A sua missão, por assim dizer, consiste basicamente
Framework	já deixa claro que o firewall se enquadra em uma espécie de barreira de defesa. A sua missão, por assim dizer, consiste basicamente em bloquear tráfego de dados indesejado e
Framework	já deixa claro que o firewall se enquadra em uma espécie de barreira de defesa. A sua missão, por assim dizer, consiste basicamente em bloquear tráfego de dados indesejado e liberar acessos aceitos.
Framework	já deixa claro que o firewall se enquadra em uma espécie de barreira de defesa. A sua missão, por assim dizer, consiste basicamente em bloquear tráfego de dados indesejado e liberar acessos aceitos. Framework é um arcabouço, uma estrutura,
Framework	já deixa claro que o firewall se enquadra em uma espécie de barreira de defesa. A sua missão, por assim dizer, consiste basicamente em bloquear tráfego de dados indesejado e liberar acessos aceitos. Framework é um arcabouço, uma estrutura, um esqueleto que serve para a agregação de

	conceitos valores e práticas que orientam a
	execução.
77 / ^ / / /	1 1:
Heterogêneos (sistemas)	A gama de dispositivos com diferentes
	capacidades de processamento
	(computadores, robôs, controladores,
	interfaces com o usuário, sistemas
	embarcados, aparelhos celulares, etc.) que
	possuem diferentes sistemas operacionais e
	arquiteturas (Linux, Windows, Mac, Intel,
	AMD, ARM) e que se comunicam através de
	diferentes protocolos de redes (Ethernet,
	ISANET, CAN, MODBUS, TCP/IP, CoAP, etc.)
	onde necessitam interagir/cooperar de forma
	distribuída para realizar determinado
	processamento para atender a demanda de
	diferentes perspectivas da informação para
	diferentes usuários.
7	
Interface	Interfaces abstraem a complexidade da rede e
	do modelo de dados. Além disso, são camadas
	que norteiam o registro, descoberta,
	provimento e o gerenciamento dos serviços
Interoperabilidade	Capacidade de realizar de forma conjunta
	uma atividade específica, onde sistemas
	trocam parâmetros e funcionalidades se
	encaixam tanto sintática e semanticamente
LAN	(Local Area Network) rede Local, como se
	define uma rede de computadores de local
	com menor alcance.
Metadados	Metadados são comumente definidos como

	dados sobre dados. Quando se trata deste
	assunto, esta é a definição mais simples e
	comumente utilizada.
	Metadados são dados que descrevem
	atributos de um recurso. Eles suportam um
	número de funções: localização, descoberta,
	documentação, avaliação, seleção, etc.
	Metadados fornecem o contexto para
	entender os dados através do tempo.
	Metadados são dados associados com
	objetos que ajudam seus usuários potenciais
	a terem vantagem completa do conhecimento
	da sua existência ou características.
	Metadados são o instrumental para
	transformar dados brutos em conhecimento
	transformar dados brutos em connecimento
Parser	O <i>parser</i> se trata de um software que realiza a
Parser	
Parser	O <i>parser</i> se trata de um software que realiza a
Parser	O <i>parser</i> se trata de um software que realiza a função de carregar os dados de entrada e
Parser	O <i>parser</i> se trata de um software que realiza a função de carregar os dados de entrada e constrói com uma estrutura de dados com
Parser	O <i>parser</i> se trata de um software que realiza a função de carregar os dados de entrada e constrói com uma estrutura de dados com eles. Essa estrutura de dados pode se tratar
Parser	O <i>parser</i> se trata de um software que realiza a função de carregar os dados de entrada e constrói com uma estrutura de dados com eles. Essa estrutura de dados pode se tratar de uma árvore de análise, árvore abstrata de
Parser	O parser se trata de um software que realiza a função de carregar os dados de entrada e constrói com uma estrutura de dados com eles. Essa estrutura de dados pode se tratar de uma árvore de análise, árvore abstrata de sintaxe ou outras estruturas que dão ideia de
Parser	O parser se trata de um software que realiza a função de carregar os dados de entrada e constrói com uma estrutura de dados com eles. Essa estrutura de dados pode se tratar de uma árvore de análise, árvore abstrata de sintaxe ou outras estruturas que dão ideia de hierarquia, para que resulte em uma
Parser	O parser se trata de um software que realiza a função de carregar os dados de entrada e constrói com uma estrutura de dados com eles. Essa estrutura de dados pode se tratar de uma árvore de análise, árvore abstrata de sintaxe ou outras estruturas que dão ideia de hierarquia, para que resulte em uma representação estrutural da entrada que foi
Parser	O parser se trata de um software que realiza a função de carregar os dados de entrada e constrói com uma estrutura de dados com eles. Essa estrutura de dados pode se tratar de uma árvore de análise, árvore abstrata de sintaxe ou outras estruturas que dão ideia de hierarquia, para que resulte em uma representação estrutural da entrada que foi feita a análise. A análise pode proceder vários
Parser	O parser se trata de um software que realiza a função de carregar os dados de entrada e constrói com uma estrutura de dados com eles. Essa estrutura de dados pode se tratar de uma árvore de análise, árvore abstrata de sintaxe ou outras estruturas que dão ideia de hierarquia, para que resulte em uma representação estrutural da entrada que foi feita a análise. A análise pode proceder vários outros passos que são executados antes da
Parser	O parser se trata de um software que realiza a função de carregar os dados de entrada e constrói com uma estrutura de dados com eles. Essa estrutura de dados pode se tratar de uma árvore de análise, árvore abstrata de sintaxe ou outras estruturas que dão ideia de hierarquia, para que resulte em uma representação estrutural da entrada que foi feita a análise. A análise pode proceder vários outros passos que são executados antes da própria análise, ou estes passos podem ser
Parser	O parser se trata de um software que realiza a função de carregar os dados de entrada e constrói com uma estrutura de dados com eles. Essa estrutura de dados pode se tratar de uma árvore de análise, árvore abstrata de sintaxe ou outras estruturas que dão ideia de hierarquia, para que resulte em uma representação estrutural da entrada que foi feita a análise. A análise pode proceder vários outros passos que são executados antes da própria análise, ou estes passos podem ser executados em um único passo, onde eles

	português Transferência de Estado
	Representacional, é uma abstração da
	arquitetura da World Wide Web (Web), um
	estilo arquitetural que consiste de um
	conjunto coordenado de restrições
	arquiteturais aplicadas a componentes,
	conectores e elementos de dados dentro de
	um sistema de hipermídia distribuído. O
	REST ignora os detalhes da implementação de
	componente e a sintaxe de protocolo com o
	objetivo de focar nos papéis dos componentes,
	nas restrições sobre sua interação com outros
	componentes e na sua interpretação de
	elementos de dados significantes.
DEOUISITOS	Doguisitos são funções, objetivos
REQUISITOS	Requisitos são funções, objetivos,
	propriedades, restrições que o sistema deve
	possuir para satisfazer contratos, padrões ou
	possuir para satisfazer contratos, padrões ou especificações de acordo com o(s) usuário(s)
	possuir para satisfazer contratos, padrões ou
SOA	possuir para satisfazer contratos, padrões ou especificações de acordo com o(s) usuário(s)
SOA	possuir para satisfazer contratos, padrões ou especificações de acordo com o(s) usuário(s) (Sommerville,1997)
SOA	possuir para satisfazer contratos, padrões ou especificações de acordo com o(s) usuário(s) (Sommerville,1997) SOA é um paradigma orientado a serviços que
SOA	possuir para satisfazer contratos, padrões ou especificações de acordo com o(s) usuário(s) (Sommerville,1997) SOA é um paradigma orientado a serviços que cresceu rapidamente como uma solução
SOA	possuir para satisfazer contratos, padrões ou especificações de acordo com o(s) usuário(s) (Sommerville,1997) SOA é um paradigma orientado a serviços que cresceu rapidamente como uma solução padrão para publicação e acesso à informação
SOA	possuir para satisfazer contratos, padrões ou especificações de acordo com o(s) usuário(s) (Sommerville,1997) SOA é um paradigma orientado a serviços que cresceu rapidamente como uma solução padrão para publicação e acesso à informação em um mundo cada vez conectado. Com SOA
	possuir para satisfazer contratos, padrões ou especificações de acordo com o(s) usuário(s) (Sommerville,1997) SOA é um paradigma orientado a serviços que cresceu rapidamente como uma solução padrão para publicação e acesso à informação em um mundo cada vez conectado. Com SOA cada componente pode ser servidor e/ou cliente a depender do contexto
Socket	possuir para satisfazer contratos, padrões ou especificações de acordo com o(s) usuário(s) (Sommerville,1997) SOA é um paradigma orientado a serviços que cresceu rapidamente como uma solução padrão para publicação e acesso à informação em um mundo cada vez conectado. Com SOA cada componente pode ser servidor e/ou cliente a depender do contexto Socket é uma abstração computacional que
	possuir para satisfazer contratos, padrões ou especificações de acordo com o(s) usuário(s) (Sommerville,1997) SOA é um paradigma orientado a serviços que cresceu rapidamente como uma solução padrão para publicação e acesso à informação em um mundo cada vez conectado. Com SOA cada componente pode ser servidor e/ou cliente a depender do contexto Socket é uma abstração computacional que mapeia diretamente a uma porta de
	possuir para satisfazer contratos, padrões ou especificações de acordo com o(s) usuário(s) (Sommerville,1997) SOA é um paradigma orientado a serviços que cresceu rapidamente como uma solução padrão para publicação e acesso à informação em um mundo cada vez conectado. Com SOA cada componente pode ser servidor e/ou cliente a depender do contexto Socket é uma abstração computacional que mapeia diretamente a uma porta de transporte, através dos protocolos TCP ou
	possuir para satisfazer contratos, padrões ou especificações de acordo com o(s) usuário(s) (Sommerville,1997) SOA é um paradigma orientado a serviços que cresceu rapidamente como uma solução padrão para publicação e acesso à informação em um mundo cada vez conectado. Com SOA cada componente pode ser servidor e/ou cliente a depender do contexto Socket é uma abstração computacional que mapeia diretamente a uma porta de

	annomiam ilaan allalaran ilm daa maiss TOD
	conseguem usar qualquer um dos meios TCP
	ou UDP.
Serviços	Funcionalidades que são fornecidas por
3	componentes de hardware ou software, sendo
	•
	abstraídas de sua complexidade, podendo ser
	oferecidas ou solicitadas. Com SOA cada
	componente pode ser servidor e/ou cliente
Stakeholder	Stakeholder significa público estratégico e
Starcholder	
	descreve uma pessoa ou grupo que fez um
	investimento ou tem ações ou interesse em
	uma empresa, negócio ou indústria.
	Em inglês stake significa interesse,
	participação, risco. Holder significa aquele
	que possui. Assim, stakeholder também
	significa parte interessada ou interveniente. E
	uma palavra em inglês muito utilizada nas
	áreas de comunicação, administração e
	tecnologia da informação cujo objetivo é
	designar as pessoas e grupos mais
	importantes para um planejamento
	estratégico ou plano de negócios, ou seja, as
	partes interessadas.
URL	URL é o endereço de um recurso disponível
	em uma rede, seja a rede internet ou intranet,
	e significa em inglês <i>Uniform Resource</i>
	Locator, e em português é conhecido por
	Localizador Padrão de Recursos.
	é um endereço virtual com um caminho que
	indica onde está o que o usuário procura, e
	pode ser tanto um arquivo, como uma
	1

	máquina, uma página, um site, uma pasta
	etc. Url também pode ser o link ou endereço
	de um site.
WAN	(Wide Area Network) Rede de computadores
	com larga escala, para grandes áreas e longo
	alcance.