

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

GREYCE NOGUEIRA SCHROEDER

**METODOLOGIA DE MODELAGEM E
ARQUITETURA DE REFERÊNCIA DO
DIGITAL TWIN EM SISTEMAS CIBER
FÍSICOS INDUSTRIAIS USANDO
AUTOMATIONML**

Porto Alegre
2018

GREYCE NOGUEIRA SCHROEDER

**METODOLOGIA DE MODELAGEM E
ARQUITETURA DE REFERÊNCIA DO
DIGITAL TWIN EM SISTEMAS CIBER
FÍSICOS INDUSTRIAIS USANDO
AUTOMATIONML**

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

Área de concentração: Controle e Automação

ORIENTADOR: Prof. Dr. Carlos Eduardo Pereira

Porto Alegre
2018

GREYCE NOGUEIRA SCHROEDER

**METODOLOGIA DE MODELAGEM E
ARQUITETURA DE REFERÊNCIA DO
DIGITAL TWIN EM SISTEMAS CIBER
FÍSICOS INDUSTRIAIS USANDO
AUTOMATIONML**

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

Orientador: _____

Prof. Dr. Carlos Eduardo Pereira, UFRGS

Doutor pela Technische Universität Stuttgart – Stuttgart, Alemanha

Banca Examinadora:

Prof. Dr. Silvia da Costa Botelho Botelho, FURG

Doutor pelo Centre National de la Recherche Scientifique/CNRS – França

Prof. Dr. Danúbia Bueno Espíndola, FURG

Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Prof. Dr. Ivan Müller, UFRGS

Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Prof. Dr. Marcelo Götz, UFRGS

Doutor pela Universität Paderborn – Paderborn, Alemanha

Prof. Dr. Walter Fetter Lages, UFRGS

Doutor pelo Instituto Tecnológico de Aeronáutica – São José dos Campos, Brasil

Coordenador do PPGEE: _____

Prof. Dr. Valner João Brusamarello

Porto Alegre, junho de 2018.

AGRADECIMENTOS

Primeiro de tudo, gostaria de agradecer a Deus por me guiar, iluminar e me dar tranquilidade para seguir em frente e não desanimar com as dificuldades. Agradeço a Ele também por manter a minha mãe ao meu lado, com a saúde que ela está hoje.

Agradeço a minha família e especialmente a minha mãe e aos meus sobrinhos/filhos: Eliziane e Luiz Eduardo, que sempre me motivaram e entenderam as minhas faltas.

Agradeço aqueles que partiram antes da conclusão deste trabalho, mas que eram meus incentivadores: meu pai e minha vó.

Ao meu orientador Prof. Dr. Carlos Eduardo Pereira, pela orientação, sugestões e por compartilhar o seu conhecimento comigo e, pelas oportunidades de trabalho. Por ser um visionário, lá no início, e me propor o Digital Twin como tema de pesquisa.

Ao meu co-orientador, Prof. Dr. Renato Ventura, por ser um incentivador e por sua colaboração.

Ao Programa de Pós-Graduação da Engenharia Elétrica - PPGEE pela oportunidade de estudo e pesquisa.

À Capes pela concessão da bolsa de estudos.

Aos colegas e amigos do PPGEE que dividiram a sala de estudo e contribuíram de alguma forma para a minha formação. Em especial ao colega e amigo Charles Steinmetz, no qual dividimos momentos de angústia, incertezas, mas juntos nós conseguimos terminar nossos trabalhos.

Por fim, um agradecimento especial a pessoa que sempre esteve ao meu lado para a conclusão desta Tese: ao meu esposo, Ricardo Nagel Rodrigues, que foi meu maior incentivador e por entender a minha ausência em diversos momentos. Obrigada por seu amor incondicional, pelas trocas de idéias e ajuda para terminar este trabalho.

RESUMO

Com as evoluções tecnológicas nas áreas de hardware, microeletrônica, sistemas de informação e computação, o conceito de sistemas ciberfísicos (do inglês Cyber-Physical Systems) vem ganhando importância. Estes sistemas se referem à junção entre sistemas computacionais distribuídos e processos físicos da natureza e, são base fundamental para a nova revolução industrial que está sendo introduzida. Esta revolução industrial é marcada pela completa descentralização do controle dos processos produtivos e uma proliferação de dispositivos inteligentes interconectados, ao longo de toda a cadeia de produção e logística. Sistemas de automação, e particularmente os sistemas de automação industrial, nos quais elementos computacionais controlam e automatizam a execução de processos físicos em plantas industriais, são um exemplo de sistemas ciber-físicos.

Com isso, percebe-se que é necessário relacionar objetos físicos a informações associadas a este objeto no mundo cibernético. Para isso, destaca-se o conceito e o uso do Digital Twin, que é uma representação virtual de objetos físicos. O Digital Twin possibilita a virtualização e centralização do controle no produto. Este estudo irá explorar uma metodologia de modelagem genérica e flexível para o Digital Twin usando a ferramenta AutomationML e propor uma arquitetura de comunicação para a troca de dados sob a ótica de Cyber Physical Systems. Com a implementação dessa metodologia, pretende-se validar o conceito proposto e oferecer um método de modelagem e configuração para obter dados, extrair conhecimento e proporcionar sistemas de visualização para os usuários.

Palavras-chave: Cyber physical system, arquitetura, sistemas de automação industrial, digital twin, Industria 4.0, modelagem.

ABSTRACT

With technological advances in the fields of hardware, microelectronics and computer systems, Cyber Physical Systems is a new concept that is gaining importance. These systems are integrations of computation, networking, and physical processes. Cyber Physical Systems are one of the pillars for the new industrial revolution, and it is marked by the complete decentralization of the control of production processes and, marked by a proliferation of interconnected intelligent devices throughout the production and logistics chain. Embedded computers and networks monitor and control the physical processes, with feedback loops where physical processes affect computations and vice versa. An industrial automation system, is an example of cyber physical systems where computational elements control and automate the execution of physical processes in industrial plants. Thus, it is clear the need to relate physical objects to information associated with this object in the cyber world. For this, this work pretends to use the concept of Digital Twin, that is a virtual representation of physical objects. Digital Twin enables the virtualization of physical components and decentralization of control. This study will explore a generic and flexible modeling methodology for Digital Twin using the AutomationML tool. Also this work proposes a communication architecture for the exchange of data from the perspective of Cyber Physical Systems. With the implementation of this methodology, we intend to validate the proposed concept and offer a modeling and configuration method to obtain data, extract knowledge and provide visualization systems for users.

Keywords: cyber physical system, architecture, industrial automation systems, digital twin, Industry 4.0, modeling.

LISTA DE ILUSTRAÇÕES

Figura 1:	Evolução das revoluções Industriais. Adaptado de: (KAGERMANN et al., 2013)	22
Figura 2:	Internet das Coisas como resultado da convergência de diferentes visões. Fonte: (ATZORI; IERA; MORABITO, 2010)	26
Figura 3:	Elementos básicos de um sistema ciber-físico. Fonte: (ROARKE, 2014)	29
Figura 4:	Processo Industrial. Fonte: Adaptado de (PEREIRA, 1996)	31
Figura 5:	Digital twin no contexto de sistemas ciber-físicos. [Fonte: autor]	32
Figura 6:	Ciclo de vida do Digital Twin. [Fonte: autor]	35
Figura 7:	Estrutura do automationML. Fonte: (AUTOMATIONML, 2018)	40
Figura 8:	Estrutura do automationML. Fonte: (HENßEN; SCHLEIPEN, 2014)	41
Figura 9:	Principais elementos do AutomatioML. Fonte: (HENßEN; SCHLEIPEN, 2014)	42
Figura 10:	Detalhes dos principais objetos em AutomationML. Fonte: (HENßEN; SCHLEIPEN, 2014)	43
Figura 11:	Esquema gráfico do exemplo usado para criar o modelo em AutomationML. [Fonte: autor]	44
Figura 12:	Exemplo da modelagem de um PLC, módulo de E/S, motor e um sensor em AML. [Fonte: autor]	45
Figura 13:	Conceito do twin model. Fonte:(GRIEVES, 2005)	52
Figura 14:	Fases do ciclo de vida e principais sistemas de software apresentado em (RIOS et al., 2015). Fonte: (RIOS et al., 2015)	59
Figura 15:	Esquema de construção da ferramenta virtual para digital twins usando integração de sensores e fusão de informações de manufatura. Fonte: (CAI et al., 2017)	63
Figura 16:	Sincronização da dados do módulos com o uso de representação digital. Fonte: (UM; WEYER; QUINT, 2017)	64
Figura 17:	Idéia geral do trabalho de Sierla. Fonte: (SIERLA et al., 2018)	65
Figura 18:	Tipos de comunicação, proposto em (ALAM; SADDIK, 2017).	66
Figura 19:	Ciclo de vida que engloba a Engenharia de Sistemas Baseada em Simulação usando EDTs. Fonte: (SCHLUSE et al., 2018)	68
Figura 20:	Framework para o digital twin de linhas de montagem no chão de fábrica. Fonte: (ZHUANG; LIU; XIONG, 2018)	69
Figura 21:	Visão geral da arquitetura de referência.	76
Figura 22:	Diagrama UML do modelo de referência para o digital twin.	79
Figura 23:	Digital twin desconectado.	82
Figura 24:	Digital twin embarcado.	83

Figura 25:	Digital twin conectado.	84
Figura 26:	Digital twin multi-dispositivo.	85
Figura 27:	Digital twin agregado.	86
Figura 28:	Visão geral da modelagem e implantação do digital twin.	89
Figura 29:	Biblioteca de papéis do digital twin.	93
Figura 30:	digital twin System Unit Classes.	95
Figura 31:	Exemplo da modelagem do robô para criar o seu digital twin.	98
Figura 32:	Atributos 'Endereço IP' (twinAddress) e 'porta' (twinPort) para a configuração do servidor WEB.	101
Figura 33:	Fluxograma geral de modelagem do digital twin.	102
Figura 34:	Exemplo de modelo de digital twin desconectado.	102
Figura 35:	Exemplo de modelo de digital twin conectado.	103
Figura 36:	Exemplo de modelo de digital twin embarcado.	104
Figura 37:	Exemplo de modelo de digital twin multi-dispositivo.	105
Figura 38:	Exemplo de modelo de digital twin agregado.	106
Figura 39:	Fluxograma de implantação em alto nível.	108
Figura 40:	Fluxograma multifuncional de implantação.	110
Figura 41:	Conjunto Válvula-Atuador	117
Figura 42:	Modelo da válvula-atuador Coester CSR6 em AutomationML	119
Figura 43:	Visualização 3D da válvula via digital twin.	121
Figura 44:	Dashboard da válvula.	124
Figura 45:	Fluxo NodeRed para a dashboard da válvula.	125
Figura 46:	Visão geral do cenário 1 - Digital Twin conectado.	125
Figura 47:	Modelo do digital twin para o cenário 1.	126
Figura 48:	Página Html gerada pelo componente HtmlView do digital twin.	128
Figura 49:	Visão geral do cenário 2 - Digital Twin Multi-dispositivo	129
Figura 50:	Modelo do digital twin para o cenário 2.	130
Figura 51:	Visão geral da topologia do cenário 3.	131
Figura 52:	Modelo do digital twin para o cenário 3.	132
Figura 53:	Diagrama de sequência da execução do watchdog para o digital twin agregado.	133
Figura 54:	Modelo de atuador da válvula em AutomationML.	134
Figura 55:	Esquema do funcionamento do estudo de caso da válvula.	135
Figura 56:	Exemplo do processo como um todo, tem-se o modelo em AML, os atributos selecionados para enviar dados para o Fiware e, por último, os dados sendo exibidos em outra plataforma.	136
Figura 57:	Robô Edubot.	137
Figura 58:	Modelo em AutomationML do robô Edubot.	139
Figura 59:	Labirinto Virtual para a simulação de envio de dados do robô.	140
Figura 60:	Modelo em AutomationML do digital twin do robô Edubot.	140

LISTA DE TABELAS

Tabela 1:	Número de publicações referente ao termo digital twin nas bases de dados	48
Tabela 2:	Principais <i>Tags</i> selecionadas para cada publicação.	50
Tabela 3:	Definições do digital twin encontradas na literatura. Adaptado de (NEGRI; FUMAGALLI; MACCHI, 2017)	55
Tabela 4:	Publicações e tópicos de estudo.	71
Tabela 5:	Bibliotecas e frameworks utilizados na implementação do software para automatização da implantação do digital twin	112
Tabela 6:	Exemplos de uma requisição Restfull API. O Prefixo * representa <code>http://<ipaddress>:<port>/api/Robot</code>	113
Tabela 7:	Testes de requisições ao digital twin do cenário 1. O Prefixo * representa <code>http://127.0.0.1:8080/api/ValveHV20073</code> . . .	127

LISTA DE ABREVIATURAS

AML	Automation Markup Language
API	Application programming interface
AR	Augmented Reality
BOL	Begining of life
CAEX	Computer Aided Engineering Exchange
CPS	Cyber Physical System
DB	Database
DT	Digital Twin
EOL	End of life
ERP	Enterprise Resource Planning
HTTP	Hypertext Transfer Protocol
I4.0	Industria 4.0
IE	Internal Element
IoT	Internet of Things
IP	Internet Protocol
JSON	JavaScript Object Notation
MES	Manufacturing execution system
MOL	Middle of life
PLM	Product Lifecycle Management
PDKM	Product Data and Knowledge Management
REST	Representational state transfer
RC	Role Classes
RM	Realidade Mista
SOA	Service Oriented Architecture
SOAP	Service Oriented Architecture Protocol
SUC	System Unit Classes

TCP	Transmission Control Protocol
VR	Virtual Reality
XML	Extensible Markup Language
WS	Web Services

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Motivação	16
1.2	Caracterização do Problema	16
1.3	Objetivos	17
1.3.1	Objetivo Geral	17
1.3.2	Objetivos Específicos	18
1.4	Organização da tese	18
2	FUNDAMENTAÇÃO TEÓRICA	21
2.1	Indústria 4.0	21
2.2	Internet das Coisas	24
2.3	<i>Big Data</i>	27
2.4	Cyber Physical Systems	28
2.4.1	Sistemas ciber-físicos e sistemas de automação industrial	31
2.4.2	Sistemas ciber-físicos e o Digital Twin	32
2.5	Ciclo de vida	32
2.6	Estratégias de projeto	36
2.6.1	Engenharia baseada em modelo	36
2.6.2	Abordagem de “baixo para cima” e “cima para baixo”	37
2.6.3	Arquitetura baseada em serviços	37
2.7	AutomationML	39
2.7.1	Estrutura do AutomationML	40
2.7.2	Exemplo do AutomationML	43
2.8	Considerações finais	46
3	REVISÃO E DEFINIÇÃO DO DIGITAL TWIN	47
3.1	Metodologia de Pesquisa	47
3.2	Histórico	52
3.3	Aplicações do digital twin	56
3.4	Digital twin no ciclo de vida do produto	57
3.5	Tópicos de estudo	60
3.6	Trabalhos relacionados	62
3.7	Conclusões	70
4	ARQUITETURA E MODELO DE REFERÊNCIA PARA O DIGITAL TWIN	73
4.1	Estratégias de projeto adotadas	74
4.2	Arquitetura de referência	75

4.3	Modelo do digital twin	78
4.4	Análise das topologias de conectividade do digital twin	80
4.4.1	Digital twin desconectado	81
4.4.2	Digital twin embarcado	82
4.4.3	Digital twin conectado	83
4.4.4	Digital twin multi-dispositivo	84
4.4.5	Digital twin agregado	85
4.5	Conclusão	86
5	MODELAGEM COM AUTOMATIONML	88
5.1	Modelagem do Dispositivo Físico	89
5.2	Modelagem do digital twin	92
5.2.1	Definição dos papéis (<i>roles</i>)	93
5.2.2	Definição das Interfaces	94
5.2.3	Criação das classes de unidades do sistema (<i>System Unit Classes - SUC</i>)	95
5.2.4	Criação de uma hierarquia de instâncias (<i>Instance Hierarchy</i>)	98
5.2.5	Configuração dos atributos do digital twin	100
5.3	Padrões de modelagem para as topologias	101
5.3.1	Digital twin desconectado	101
5.3.2	Digital twin conectado	102
5.3.3	Digital twin embarcado	103
5.3.4	Digital twin multi-dispositivo	104
5.3.5	Digital twin agregado	104
5.4	Conclusão	105
6	IMPLANTAÇÃO AUTOMATIZADA BASEADA EM MODELO DO DIGITAL TWIN	107
6.1	Metodologia geral de implantação	108
6.2	Software para automatização da implantação	109
6.3	Implementação da API do digital twin	112
6.4	Conclusão	113
7	ESTUDOS DE CASO	115
7.1	Estudo de Caso 1: Válvula Industrial	115
7.1.1	Objetivo	116
7.1.2	Modelagem do dispositivo físico no AutomationML	119
7.1.3	Modelagem da Geometria 3D	120
7.1.4	WatchDog	121
7.1.5	Dashboard	124
7.1.6	Cenário 1: Digital Twin Conectado	125
7.1.7	Cenário 2: Digital Twin Multi-dispositivo	128
7.1.8	Cenário 3: Digital Twin Agregado	130
7.1.9	Cenário 4: Digital Twin com Fiware	133
7.1.10	Discussão	135
7.2	Estudo de caso 2: Robô Edubot	137
7.2.1	Objetivo	138
7.2.2	Modelagem do robô	138
7.2.3	Modelagem do digital twin em AML	139
7.2.4	Configuração e Implantação	139

7.2.5	Resultados e discussão	141
8	CONCLUSÃO	143
8.1	Principais contribuições	145
8.2	Trabalhos Futuros	146
	REFERÊNCIAS	148

1 INTRODUÇÃO

A indústria, nos últimos séculos, passou por mudanças em sua forma de atuação, gerando inovação e mudanças na sociedade e no setor econômico. Atualmente, vive-se no limiar da quarta revolução industrial onde é previsto um alto nível de automação e inteligência em máquinas. Os processos de produção tendem a se tornar cada vez mais eficientes, autônomos e customizáveis. Dispositivos de campo, máquinas industriais, plantas, fábricas e até mesmo os produtos manufaturados por estas, estarão conectados em rede, comunicando-se entre si sem intervenção humana. Isso irá permitir que os objetos tornem-se exploráveis, controláveis e analisáveis através da rede, levando a novos tipos de serviços e modelos de negócios. Conceitos de Indústria 4.0 (KAGERMANN et al., 2013) e *SmartFactories* (WANG et al., 2016) surgem para definir os novos paradigmas dessa indústria.

A combinação de tecnologias da Internet e tecnologias orientadas para o futuro, como os objetos “inteligentes” (máquinas e produtos), resultam em uma nova mudança de paradigma fundamental na produção industrial. A visão de produção futura contém sistemas de fabricação modulares e eficientes e caracteriza cenários em que os produtos controlam seu próprio processo de fabricação. Isto visa permitir a fabricação de produtos individuais, mantendo as condições econômicas de produção em massa.

Sistemas ciber-físicos (*Cyber Physical Systems - CPS*)(LEE et al., 2008) e a Internet das Coisas (*Internet of Things - IoT*)(ATZORI; IERA; MORABITO, 2010) têm sido propostos como tecnologias base da Indústria 4.0. Um sistema ciber-físico pode ser descrito como um conjunto de dispositivos físicos, objetos e equipamentos que possuem uma representação virtual e estas interagem no ciberespaço através de uma rede de comunicação, e controlam suas entidades físicas. A Internet das Coisas permite que esta conectividade aconteça. O modelo cibernético de cada entidade física pode ser visto como uma representação digital da entidade real, e referenciado por digital twin¹ (DT) (LEE; BAGHERI; KAO, 2015) (RIOS et al., 2015). Nesse sentido, o digital twin pode monitorar e controlar a sua entidade física, e essa pode enviar dados para atualizar seu gêmeo virtual.

O digital twin pode servir como base para a digitalização dos produtos manufaturados, das máquinas de produção a até mesmo para a virtualização da fábrica inteira. Um ambiente digital da fábrica pode ser criado, na qual a empresa pode otimizar a operação diretamente através da cadeia de produção, manipular parâmetros e processos de produção; adaptando o produto aos requisitos do mercado. Os dados gerados pelo digital twin podem criar uma imagem abrangente do produto manufaturado e, do processo de produção. Portanto, o digital twin pode coletar e avaliar a informação de forma contínua, permitindo, entre outras coisas, reduzir custos, agilizar o ciclo de produção, e reduzir o tempo de introdução de novos produtos, detectando configurações ineficientes dos processos subjacentes. O conceito do digital twin, portanto, está diretamente relacionado com a Indústria 4.0.

¹O termo *Digital Twin* não possui um tradução em português, portanto, o termo será usado ao longo deste trabalho em letras minúsculas, visto que, será muito usado durante todo o texto.

1.1 Motivação

Em CPS, para cada produto, máquina ou equipamento, ao lado de sua descrição física real, uma representação virtual (lógica) precisa ser desenvolvida. Esse processo é uma das consequências da Indústria 4.0, e para isso, o digital twin mostra sua importância: através dele, para cada produto, máquina ou equipamento, ao lado de sua descrição física real, uma representação virtual (lógica) precisa ser desenvolvida.

O digital twin permite a coleta de dados durante todo o ciclo de vida e informações podem ser inferidas a partir destes dados com o uso de sistemas industriais inteligentes. Por exemplo, o digital twin pode melhorar a produtividade da indústria e a qualidade dos produtos, através de prognósticos e diagnósticos inteligentes utilizando conhecimento adquirido por dados de sensores, máquinas e sistemas. O digital twin pode fornecer informações aos usuários e a outros digital twins em qualquer momento e em qualquer lugar. Além disso, o digital twin pode ser usado para realizar simulações obtendo-se melhorias no funcionamento da sua contraparte física.

Portanto, vê-se que o digital twin é um tema de pesquisa relevante e importante para o setor industrial. No entanto, percebe-se que o seu desenvolvimento encontra-se em estado inicial de pesquisa. Isso serve como motivação para este trabalho, devido a sua importância na indústria 4.0, visto que, a previsão é que cada componente e máquina poderá ter um digital twin no espaço cibernético.

1.2 Caracterização do Problema

Sendo o digital twin uma representação do produto real no mundo virtual, esse pode conter informações desde o início até o fim do ciclo de vida do seu representante real. O digital twin pode ser considerado um sistema que fornece funcionalidades, como ar-

mazenamento de dados, protótipos de simulação, programas para predição de falhas dos componentes físicos, etc. Com isso, o digital twin necessita de interfaces de acesso para seus usuários; uma identificação única; e sistemas de compartilhamento e gerenciamento dos seus dados. Portanto, para a sua implantação ocorrer, muitos aspectos devem ser observados, como por exemplo:

- O digital twin pode ser composto por diferentes modelos e diferentes tipos de dados. Com isso, percebe-se um problema em como ter um modelo que agrega estas diferenças e, um modelo que permita que sistemas heterogêneos acessem os dados do digital twin;
- O digital twin precisa ser um centralizador de informações;
- O digital twin precisa ter uma estrutura para fornecer funcionalidades e serviços para outros sistemas;
- O digital twin deve ter um meio de comunicação com a sua parte física, assim como para outros digital twins.

Um desafio central para a realização do digital twin é, portanto, uma metodologia de modelagem genérica e uma arquitetura de organização dos elementos que podem compor um digital twin. Uma arquitetura e um modelo flexível para o digital twin poderiam atender sua implantação em várias áreas da manufatura e automação industrial.

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo geral deste trabalho é propor uma metodologia de modelagem do digital twin usando AutomationML para a sua implementação em sistemas ciber-físicos industriais juntamente com uma arquitetura de referência para a organização dos seus componentes. Essa modelagem deve ser abrangente podendo conter diversas informações

relevantes do dispositivo físico, tais como especificações do seu projeto construtivo, propriedades técnicas, identificação única, descrição dos seus componentes, estado atual de suas propriedades dinâmicas, especificação de operação, etc.

1.3.2 Objetivos Específicos

Para atingir o objetivo geral, os seguintes objetivos específicos foram determinados:

- Definir serviços para digital twin que permitam o compartilhamento dos dados e acesso às funcionalidades providas pelo digital twin;
- Implementar um armazenamento de dados para manter o histórico de operação do dispositivo físico e do digital twin;
- Gerenciar o disparo de eventos e possível notificação da ocorrência de tais eventos para outros componentes do sistema, aplicações ou usuários;
- Implementar controle e definir regras específicas de controle de acesso aos recursos do digital twin, incluindo informações e funcionalidades;
- Implementar interfaces de comunicação com o dispositivo físico, sendo assim responsável pela atualização das propriedades do digital twin quando o dispositivo físico sofrer alguma alteração no seu estado e vice-versa;
- Propor uma nomenclatura para as topologias observadas atualmente na literatura e propor novas topologias suportadas pela arquitetura e modelo de referência, visto que um sistema com digital twin pode apresentar diferentes formas de organização.

1.4 Organização da tese

Esta Tese de Doutorado está organizada da seguinte forma:

O Capítulo 2 apresenta a fundamentação teórica a respeito dos principais conceitos necessários para o entendimento deste trabalho. No capítulo são descritos: a fundamentação conceitual sobre Indústria 4.0, sistemas ciber-físicos, Internet das coisas, Big Data, Web Services e manutenção inteligente. Na Seção sobre a descrição do AutomationML é apresentada uma visão geral de uso desta ferramenta para modelagem.

O Capítulo 3 apresenta a revisão e a definição do conceito digital twin. É descrita a metodologia de pesquisa na literatura e baseada nessa pesquisa, o restante do capítulo aborda a história do digital twin e suas aplicações encontradas na literatura, assim como os tópicos de estudo para a sua implantação. Por fim, tem-se uma descrição e detalhamento dos trabalhos relacionados ao tema desta Tese, estes trabalhos apresentam soluções existentes no contexto da abordagem proposta fazendo um comparativo em termos de vantagens e limitações da proposta desta Tese.

No Capítulo 4 é apresentada a arquitetura e o modelo de referência para o digital twin. É proposta uma arquitetura de referência para o digital twin, para a organização de seus componentes, seguido de um modelo genérico e flexível que atende as necessidades da arquitetura. Também é proposta uma categorização do digital twin baseada na topologia de seus componentes.

O Capítulo 5 descreve uma metodologia para a modelagem e implantação do digital twin utilizando AutomationML para a modelagem, e serviços Web para a disponibilização dos serviços via internet, que segue a arquitetura e modelo de referência do capítulo anterior. Além disso, será demonstrado que a metodologia proposta pode ser usada para modelagem de qualquer uma das topologias identificadas no capítulo anterior.

No Capítulo 6 é apresentada uma sistematização e automatização do processo de implantação do digital twin baseada na modelagem descrita no capítulo 5. Na Seção 6.1 é

apresentada uma visão geral da metodologia de implantação proposta. Na Seção 6.2 é proposto um software que recebe um arquivo contendo o modelo do digital twin e implementa todos os seus componentes de forma automatizada. O resultado da implantação é um serviço web com todas as funcionalidades em execução conforme definidas no modelo. A Seção 6.3 descreve com detalhes a interface de programação de aplicações ou API (*Application programming interface*) implementada.

O Capítulo 7 mostra os resultados gerados a partir da implantação da metodologia. A validação do modelo e do sistema é realizada no contexto de dois estudos de caso, com diferentes cenários.

O Capítulo 8 discute e apresenta as principais contribuições, inovações e limitações desta Tese. Por fim, são apresentados os trabalhos futuros no âmbito da continuidade desta Tese.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão abordados os conceitos fundamentais para o leitor se contextualizar no âmbito deste trabalho. Inicialmente, os seguintes conceitos serão introduzidos neste capítulo: Indústria 4.0, Internet das Coisas e *Big Data*. Na sequência, será abordado o conceito de sistemas ciber-físicos, e a sua relação com sistemas de automação. Também será abordada a relação entre o digital twin e sistemas ciber-físicos. Além desses conceitos-chaves, o leitor terá uma breve explicação sobre o ciclo de vida do produto. Por fim, será abordada algumas estratégias de projeto (que serão usadas para a implantação do digital twin), o que são os *Web Services* e uma introdução sobre a linguagem de modelagem AutomationML usada neste trabalho.

2.1 Indústria 4.0

A Indústria 4.0 é chamada de a quarta revolução industrial. A cada nova era industrial, os avanços tecnológicos tiveram um impacto fundamental no aumento da produtividade das indústrias. Deste modo, quatro avanços tecnológicos mudaram profundamente a dinâmica industrial e provocaram novas revoluções: a máquina a vapor, a eletricidade, a revolução digital e os sistemas ciber-físicos, conforme ilustrado na Figura 1.

A primeira revolução industrial começou com o aperfeiçoamento da máquina a vapor

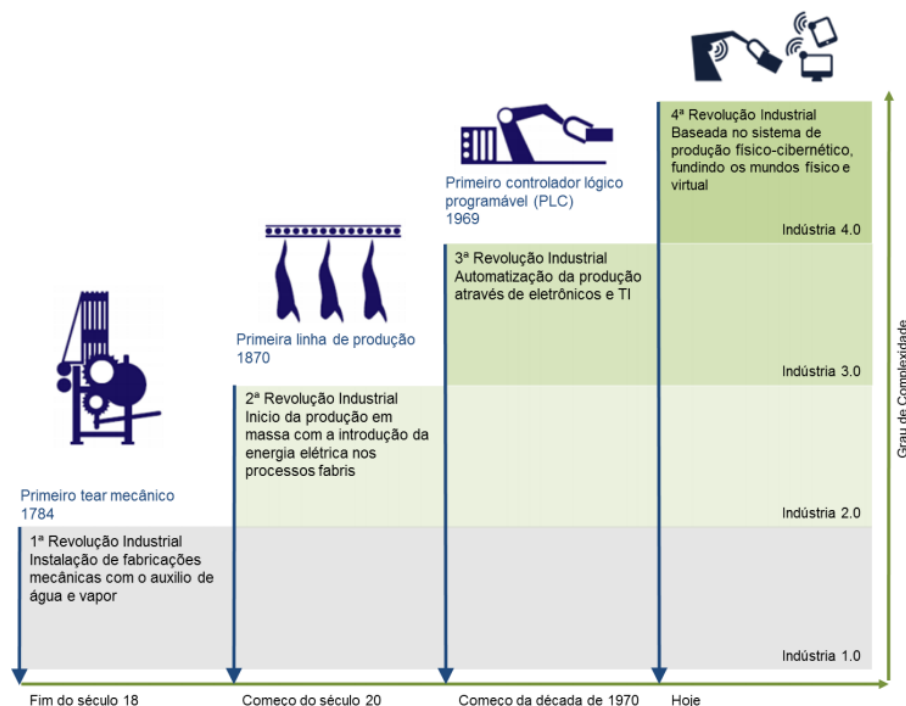


Figura 1: Evolução das revoluções Industriais. Adaptado de: (KAGERMANN et al., 2013)

por James Watt, colocando a indústria têxtil como símbolo da produção excedente, gerando a riqueza da época, criando um novo modelo econômico. A segunda revolução foi idealizada por Henry Ford, fundador da Ford Motor Company e o primeiro empresário a aplicar a montagem em série de forma a produzir em massa automóveis em menos tempo e a um menor custo. A introdução de seu modelo Ford revolucionou os transportes e a indústria dos Estados Unidos. Ele criou a linha de produção em massa, reduzindo custos de produção e popularizando o produto. A terceira revolução industrial, chamada de a “Era da automação”, iniciou com a implantação de computadores no chão de fábrica, trazendo controles eletrônicos, sensores e atuadores capazes de otimizar a produção e torná-la mais rápida e com controle de qualidade. E, por fim, em 2011, surge a quarta revolução industrial ou Indústria 4.0, que engloba algumas tecnologias para automação e troca de dados e utiliza conceitos de Sistemas ciber-físicos, Internet das Coisas e Computação em Nuvem.

Nesta época, representantes da economia e política resolveram reforçar a competi-

tividade do setor industrial alemão, criando o termo *Industrie 4.0*. O Governo Federal alemão apoiou a iniciativa e anunciou as primeiras recomendações para implementação da Indústria 4.0 que foram publicadas no ano de 2013 (KAGERMANN et al., 2013). Em seguida, nos EUA, a Indústria 4.0 tornou-se o ponto de partida para o desenvolvimento de fábricas inteligentes (*Smart Factories*) (WANG et al., 2016). Ainda, outros países também iniciaram pesquisas, como por exemplo, o Japão (onde existe uma colaboração com a Iniciativa de *Robot Revolution*), China (onde o esboço da Indústria 4.0 está pautada no projeto “Made in China 2025” (WÜBBEKE et al., 2016)) e outras inúmeras iniciativas da União Europeia.

A indústria 4.0 introduz uma série de mudanças de paradigma que mudam a maneira do funcionamento da indústria e do processo de chegada do produto até o consumidor. Com a indústria 4.0 pretende-se: economizar recursos, aumentar a lucratividade, reduzir o desperdício, prever erros e atrasos, acelerar a produção, conseguir intervir rapidamente em casos de problemas da produção, e muito mais (BRETTEL et al., 2014). Portanto, as principais características são:

- Digitalização dos produtos em um ecossistema interconectado;
- Aumento da produtividade por meio da otimização e automação;
- Geração de conhecimento através dos dados;
- Prevenção de falhas de equipamentos antes da linha de produção parar;
- Customização dos produtos em larga escala

Para (HERMANN; PENTEK; OTTO, 2016), Indústria 4.0 é um termo coletivo para tecnologias e conceitos da cadeia de valor de uma fábrica inteligente que monitora os

processos físicos , cria cópias virtuais do mundo físico e toma decisões de forma descentralizada. O autor também afirma que a Indústria 4.0 possui seis princípios de projeto. Esses princípios orientam as empresas a identificarem e a implementarem os cenários previstos na Indústria 4.0, sendo eles: interoperabilidade, virtualização, descentralização, orientação a serviços e modularidade.

No entanto, uma mudança advinda da Indústria 4.0 importante, é o produto inteligente que contém informações sobre seu processo produtivo, comunica-se com a cadeia produtiva e decide quais os passos a serem dados. Os produtos podem ser rastreados durante todo o seu ciclo de vida, guardando a sua própria história. Eles também podem guardar o seu status atual e rotas alternativas para alcançar seu destino final (HERMANN; PENTEK; OTTO, 2016).

Máquinas inteligentes que antecipam a ocorrência de falhas ou problemas de qualidade no processo produtivo e organizam o processo de decisão e auto otimização são outros cenários para a Indústria 4.0 (LEE, 2015).

Segundo (RÜSSMANN et al., 2015), para a realização desses avanços citados, a Indústria 4.0 é baseada nos seguintes pilares: *Big Data*, robôs autônomos, execução de simulações, integração de sistemas, Internet das Coisas, segurança cibernética, sistemas em nuvem, manufatura aditiva e realidade aumentada. Nas próximas seções, será visto, dentre esses pilares, os conceitos relacionados a este trabalho que são: Internet das Coisas e *Big Data*.

2.2 Internet das Coisas

De acordo com Kagermann (KAGERMANN et al., 2013), a integração da Internet das Coisas (IoT) e da Internet dos Serviços (IoS) no processo de manufatura iniciou a

quarta revolução industrial (KAGERMANN et al., 2013). A Internet das Coisas permite que “coisas” e “objetos”, tais como RFID, sensores, atuadores, interajam uns com os outros, e cooperem com seus componentes “inteligentes” vizinhos para alcançarem objetivos (ATZORI; IERA; MORABITO, 2010).

Para (GUBBI et al., 2013) a Internet das Coisas é quando elementos físicos têm identidades virtuais e operam em espaços por meio de interfaces inteligentes para se conectar e comunicar dentro de contextos sociais e ambientais.

Outro elemento presente na Internet das Coisas, é a ideia de ubiquidade, que se refere à noção de algo que está presente em todos os lugares e em todos os momentos, persistente, sempre disponível e atuante (SANTAELLA et al., 2013). Além disso, a ubiquidade, em muitos aspectos supera as noções tradicionais de espaço físico e tempo, como no caso de atualização do digital em rede, em que muitos eventos ocorrem de modo simultâneo e em muitos lugares diferentes.

O significado de Internet das Coisas abrange a área de sensores e atuadores sem fio, de objetos conectáveis às redes que utilizam o protocolo TCP/IP (*Transmission Control Protocol/Internet Protocol*), assim como as tecnologias de semântica de dados, concebendo desta maneira, uma visão orientada às coisas, uma visão orientada à internet e uma visão orientada à semântica (ATZORI; IERA; MORABITO, 2010), conforme apresentado na Figura 2.

Atzori (ATZORI; IERA; MORABITO, 2010) afirma que: (i) a visão orientada às coisas objetiva demonstrar propostas que assegurem o melhor aproveitamento dos recursos dos dispositivos e sua comunicação; (ii) a visão orientada à semântica foca na representação, armazenamento, pesquisa e organização da informação gerada, procurando soluções para a modelagem das descrições que permitam um tratamento adequado para os dados



Figura 2: Internet das Coisas como resultado da convergência de diferentes visões. Fonte: (ATZORI; IERA; MORABITO, 2010)

produzidos pelos objetos; (iii) enquanto que a visão orientada à internet tem o intuito de conceber modelos e técnicas destinadas a interoperabilidade dos dispositivos em rede.

Na era da Internet das Coisas, os dispositivos físicos estão conectados à Internet, de modo que um grande volume de dados são coletados e transmitidos através da rede. A possibilidade de bilhões de pessoas se comunicarem por dispositivos móveis, com poder de processamento, capacidade de armazenamento e acesso ao conhecimento, é ilimitada (GANTZ; REINSEL, 2011). Como resultado, o volume de dados cresce rapidamente. Nesse contexto, surge o conceito de *Big Data* (MAYER-SCHÖNBERGER; CUKIER, 2013), o qual é abordado na próxima seção.

2.3 *Big Data*

Big Data é o termo que descreve um imenso volume de dados, incluindo dados estruturados, semi-estruturados e não estruturados. A definição principal de *Big Data* não vem dos dados em sua forma bruta, mas do processamento e análise destes dados. Em geral, *Big Data* serve para descrever uma grande quantidade de dados criados por diversas fontes de dados. Manyka (MANYIKA et al., 2011) afirma que *Big Data* refere-se aos dados maciços que não poderiam ser coletados, armazenados, gerenciados, compartilhados, analisados e computados por ferramentas de dados comuns dentro de um tempo tolerável. No entanto, para os usuários de dados, o valor dos dados é mais valioso do que a sua enorme quantidade (GANDOMI; HAIDER, 2015). Assim, *Big Data* também é interpretado como a capacidade de analisar e adquirir rapidamente o conhecimento e a informação de grandes volumes de dados.

As dimensões do *Big Data* podem ser classificadas em sete Vs segundo (CHEN; MAO; LIU, 2014)(UDDIN; GUPTA et al., 2014), sendo esses:

1. Volume: considera o montante de dados gerados e coletados, variando de Peta bytes (10^{15} bytes - PB) a Zeta bytes (10^{21} bytes - ZB) (GANTZ; REINSEL, 2011);
2. Velocidade: refere-se à velocidade de análise dos dados, significa que a geração de dados é rápida, e o processamento de dados exige alta pontualidade;
3. Variedade: indica a diversidade dos tipos de dados coletados, ou seja, significa que o tamanho, o conteúdo, o formato e as aplicações dos dados são diversificados. Por exemplo, os dados podem ser estruturados (dígitos, símbolos e tabelas); dados semi-estruturados (árvores, gráficos e documentos XML) ou podem ser dados não estruturados (registros, áudios, vídeos, documentos e imagens) (GANDOMI;

HAIDER, 2015);

4. Veracidade: quantifica ruído e confiabilidade do conjunto de dados coletados;
5. Validade: refere-se à exatidão dos dados em relação ao uso pretendido. Em outras palavras, os dados podem não ser válidos se não forem compreendidos corretamente. Ou seja, o mesmo conjunto de dados pode ser válido para um aplicativo ou uso e, em seguida, inválido para outros;
6. Volatilidade: indica a validade dos dados e por quanto tempo eles podem ser armazenados;
7. Valor: ao contrário dos outros V citados, esse V é o resultado desejado do processamento, sempre com o interesse em extrair o máximo valor de qualquer grande conjunto de dados em questão.

Portanto, com uma visão que *Big Data* pode fornecer, os fabricantes no setor industrial podem aumentar a qualidade e a produção, minimizando o desperdício baseado nas informações adquiridas pelas análises de dados. Eles podem resolver os problemas rapidamente e ter agilidade para tomar decisões de negócios.

2.4 Cyber Physical Systems

Um sistema ciber-físico (*Cyber-Physical Systems* ou CPS) é um sistema que combina e coordena elementos computacionais e físicos. O CPS integra a capacidade de atuação do mundo físico e a inteligência do mundo cibernético para adicionar novos recursos aos sistemas físicos do mundo real (LEE et al., 2008). Ou seja, sistemas ciber-físicos são definidos como sistemas físicos e de engenharia cujas operações são monitoradas, coordenadas, controladas e integradas por um núcleo de computação e comunicação. Esses

sistemas pretendem integrar objetos do mundo físico e sistemas de informação, realizando interconexões, bem como o compartilhamento de informação.

Um sistema ciber-físico agrega recursos de computação, comunicação e armazenamento para monitorar e controlar entidades do mundo físico, de forma confiável, segura, eficiente e em tempo real (SHA et al., 2008). Sistema ciber-físico é uma tecnologia emergente e capacitadora para enfrentar os desafios futuros em relação a automação industrial. Neste domínio, compreendem máquinas inteligentes, sistemas de armazenamento e instalações de produção capazes de realizar troca de informações de forma autônoma, desencadeando ações e controle uns dos outros de forma independente. Essa evolução tecnológica é inserida na quarta revolução industrial (JAZDI, 2014).

A Figura 3 (ROARKE, 2014) apresenta os elementos básicos que compõe um sistemas ciber-físico, que são os seguintes:

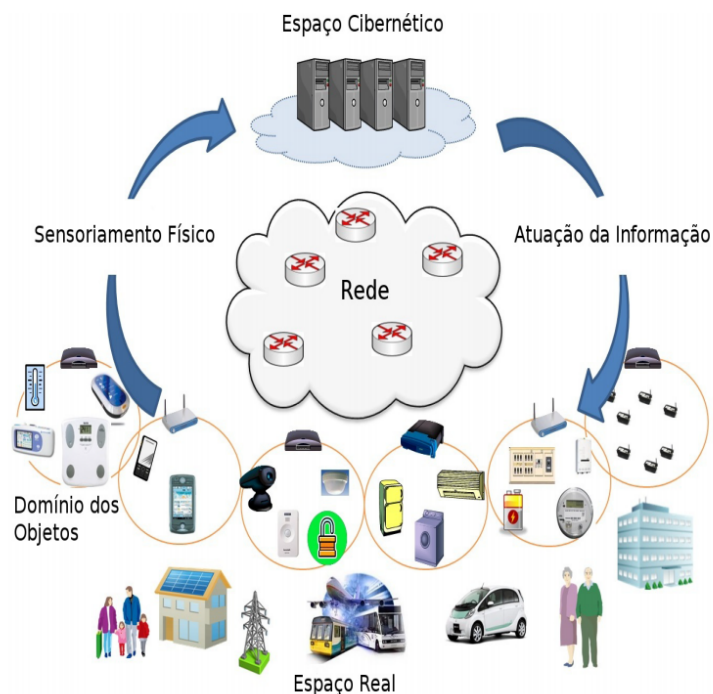


Figura 3: Elementos básicos de um sistema ciber-físico. Fonte: (ROARKE, 2014)

- Espaço real: possui objetos do ambiente físico, como carros, equipamentos, pessoas, casas, etc.

- Domínio dos objetos: constitui-se de sensores, atuadores (ou dispositivos dotados de sensores, atuadores) conectados em rede.
- Espaço cibernético/virtual: é formado pela parte computacional que recebe os dados do sensoriamento físico, faz o controle e promove a atuação no domínio dos objetos.

Gorecky (GORECKY et al., 2014) aborda a relação entre humanos e sistemas ciber-físicos e diz que essa relação acontece com o apoio de interfaces. O homem terá a função principal de determinar a estratégia de produção. Podem então surgir questões sobre a autonomia e o poder de decisão por parte do homem. Além disso, os trabalhadores não terão a necessidade de ter um local de trabalho fixo, pois eles terão acesso às informações via redes móveis.

Segundo Hermann (HERMANN; PENTEK; OTTO, 2016) a integração dos processos computacionais e físicos, dos computadores e das redes de monitoramento e controle de processos são características importantes dos sistemas ciber-físicos. E, a integração de sistemas embarcados e Internet das Coisas servem como base para os sistemas ciber-físicos. Sistemas embarcados podem realizar operações com outros sistemas em redes fechadas. Já a Internet das Coisas é a interligação dos objetos físicos por meio de redes de dados globais ou locais. Com a Internet das Coisas, os objetos podem interagir uns com os outros de forma autônoma.

Os sistemas ciber-físicos, junto com a Internet das Coisas, facilitam a visão e execução de “Fábricas Inteligentes” (ZUEHLKE, 2010) com as suas estruturas modulares. Eles permitem que processos físicos sejam monitorados, que cópias virtuais sejam criadas do mundo físico e que essas tomem decisões descentralizadas. Com a Internet das Coisas, os sistemas ciber-físicos comunicam e cooperam entre si e com os humanos, e através da

computação em nuvem, os serviços são oferecidos e utilizados pelos participantes de toda a fábrica.

2.4.1 Sistemas ciber-físicos e sistemas de automação industrial

Sistema ciber-físico é um dos conceitos principais para a Indústria 4.0, e que juntos, direcionam a indústria atual para obtenção de uma produção mais eficiente (LEE; BAGHERI; KAO, 2015).

Em uma planta industrial, sistemas de automação industrial são usados para controlar a planta. Eles recebem dados dos sensores da planta física, executam a parte de controle e enviam dados para os atuadores. Neste processo, conforme pode ser visto na Figura 4, podem ser identificados os componentes *cyber* e físicos de um sistema ciber-físico. No sistema de automação, tem-se a parte cibernética e, na planta industrial tem-se a parte física do sistema ciber-físico.

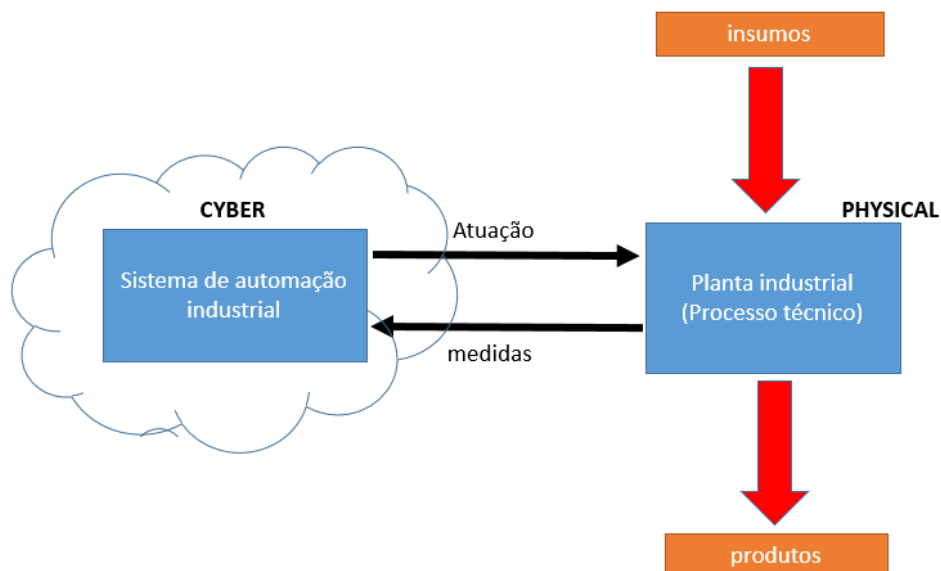


Figura 4: Processo Industrial. Fonte: Adaptado de (PEREIRA, 1996)

Portanto, um sistema de automação industrial pode ser abordado como um sistema ciber-físico, já que recebe dados de sensores, processa os valores e envia dados de atuação.

Sendo que, uma das diferenças entre eles, reside no fato de que em um sistema ciber-físico, o controle é realizado na camada cibernética, e em um sistema de automação, o controle não necessita dessa restrição.

2.4.2 Sistemas ciber-físicos e o Digital Twin

Em sistemas ciber-físicos, cada dispositivo físico tem sua representação virtual, culminando no conceito de digital twin (Figura 5). O digital twin pode monitorar e controlar sua entidade física, enquanto essa pode enviar dados para atualizar seu modelo virtual (digital twin)(LEE et al., 2008) (LEE et al., 2013).

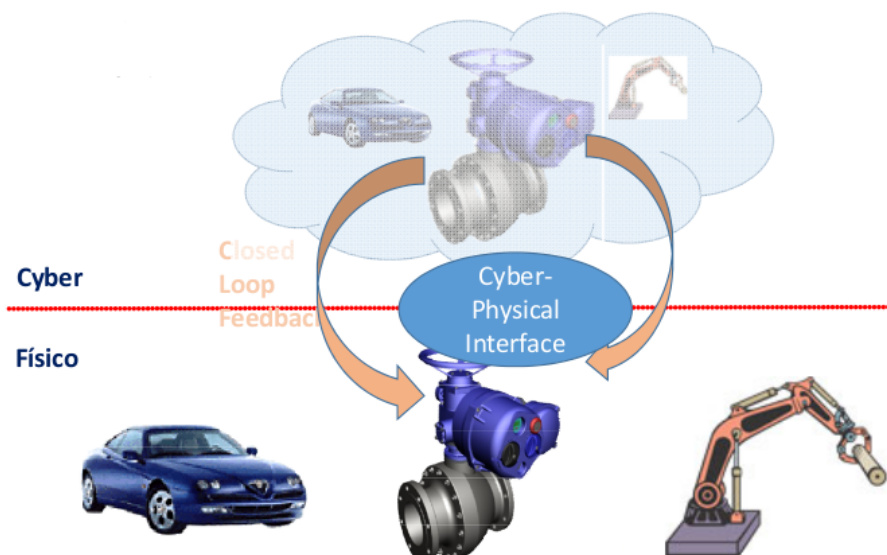


Figura 5: Digital twin no contexto de sistemas ciber-físicos. [Fonte: autor]

2.5 Ciclo de vida

O digital twin está diretamente ligado ao conceito de gerenciamento de ciclo de vida de produto, em inglês *Product Lifecycle Management* (PLM) (KIRITSIS, 2011). PLM é, no ramo da indústria, o processo de gerenciamento completo do ciclo de vida de um produto, desde a sua concepção, passando pelo desenvolvimento e manufatura, sua entrada

em serviço operacional, manutenção e, por fim, descarte.

A gestão do ciclo de vida de produtos, ou *Product Lifecycle Management (PLM)*, consiste em um conjunto de processos para gerenciar o ciclo de vida completo de um produto. O PLM integra pessoas, dados, processos e sistemas de negócios fornecendo informações para a empresa e sua rede de fornecedores, distribuidores e clientes. Atualmente, com as pressões ambientais e leis regulando o descarte de resíduos sólidos, o acompanhamento e registro do ciclo de vida dos produtos assume um papel importante na sustentabilidade da empresa.

Podemos categorizar o ciclo de vida de um produto em fases que compõem três sub-grupos:

- BOL (*Beginning of life*): incluindo a fase de projeto e manufatura;
- MOL (*Middle of life*): incluindo uso e manutenção;
- EOL (*End of life*): descarte, fim da vida, fase caracterizada por orientar o que fazer com o produto quando ele não possui mais utilidade.

Segundo Tao (TAO et al., 2017), as fases do ciclo de vida são as seguintes:

- Projeto: A equipe de desenvolvimento de produtos realiza o trabalho de projeto de produtos de forma colaborativa através da troca e compartilhamento de dados. Os dados envolvidos no projeto do produto incluem descrição da função e aparência do produto, configurações do produto, parâmetros de projeto e dados de teste, etc. Incluindo dados históricos de falhas de produtos similares, para melhorar o projeto de um novo produto.
- Manufatura: De acordo com as especificações de projeto, as matérias-primas ou componentes são processados ou montados em produtos, e os produtos são inspec-

onados através de testes de qualidade. Nesta fase, o processo dinâmico de execução da fabricação precisa ser monitorado e gerenciado. Portanto, dados de desempenho, parâmetros, condições e fatores do processo de produção são coletados e registrados para monitorar o processo de produção.

- **Uso:** Com base nas informações do manual do usuário, o cliente pode operar o produto normalmente. Durante a fase de uso, uma grande quantidade de dados é gerada, como dados do status do produto, dados do ambiente operacional, dados do comportamento do usuário. Esses dados podem ser usados não só para manutenção e reparo do produto, mas também para melhorar o projeto do produto.
- **Manutenção:** Esta etapa é responsável pela manutenção, serviço e reparo do produto. De acordo com os dados adquiridos dos produtos, soluções de manutenção e serviço adequadas são geradas. Como resultado, serviços eficientes são fornecidos aos usuários. Neste processo, dados e causas de falhas, dados de manutenção, qualidade dos componentes e dados de status são registrados e gerenciados para prever a vida útil do produto e outras falhas no produto.
- **Descarte:** Quando um produto é reciclado, o valor restante dos componentes individuais é analisado para determinar quando, como, onde e o que reciclar ou descartar, com base nos dados do produto e nos dados de manutenção armazenados. Para maximizar os benefícios de reciclagem, é necessário considerar o custo de reciclagem e desmontagem, o estado reutilizável, o valor e o tempo restante dos componentes.

A Figura 6 ilustra que o digital twin é um modelo que agrega dados e informações durante o seu ciclo de vida, ou seja, é um modelo que cresce ao longo de sua vida. Isto se torna importante para solucionar vários problemas do produto, e em especial, a coleta de

dados ao longo da vida do produto gera informações que podem servir para um descarte sustentável e, para fornecer melhorias nos modelos seguintes do mesmo produto.

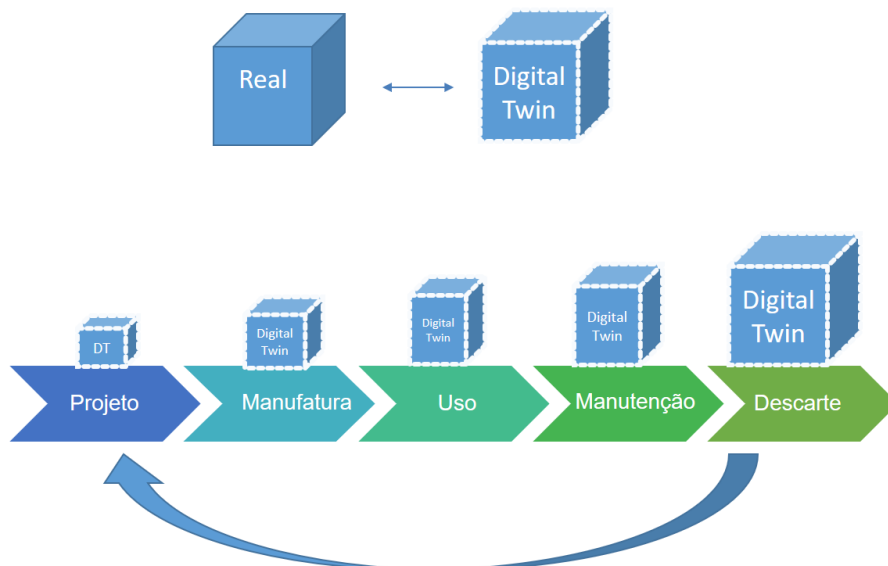


Figura 6: Ciclo de vida do Digital Twin. [Fonte: autor]

Conforme mencionado acima, dados, informações e conhecimento são criados em todas essas fases. Entre os dois primeiros estágios de BOL (projeto e manufatura), a criação de dados, informações e conhecimento é suportada por sistemas inteligentes como CAD/CAM/CAE e outros softwares de simulação.

Durante o MOL (fases de uso e manutenção) dados são gerenciados pelo *Product Data Management* (PDM). Este dados contribuem para realizar diagnósticos de saúde do produto e é útil para realizar previsões de falhas. Por fim, o descarte do produto (EOL) é caracterizado por vários cenários, tais como: reutilização do produto com remodelação, reutilização de componentes com desmontagem e remodelação, recuperação de materiais sem desmontagem, recuperação de materiais com desmontagem e, eliminação com ou sem incineração.

O objetivo é permitir que o gerenciamento de fluxo de informações vá além do cliente, fechando os laços de informações do ciclo de vida do produto e permitindo que as infor-

mações se transformem em conhecimento tanto para outras etapas do ciclo, como para novos produtos. O fechamento do ciclo de informações (veja a seta na Figura 6) poderá oferecer as seguintes vantagens (ABRAMOVICI, 2007):

1. Os produtores receberão dados completos sobre os modos de utilização e as condições de descarte e reciclagem dos seus produtos;
2. Os especialistas nos serviços de manutenção e reciclagem serão auxiliados no seu trabalho, tendo:
 - um relatório completo e sempre atualizado sobre o status do produto,
 - e assistência em tempo de execução através da internet
3. Os projetistas serão capazes de usar os conhecimentos para melhorar o projeto do produto em direção aos objetivos de qualidade do ciclo de vida;
4. Os recicladores/reutilizadores poderão obter informações precisas sobre “materiais de valor” que chegam através de rotas EOL.

2.6 Estratégias de projeto

Uma estratégia de projeto define em alto nível qual será a abordagem no desenvolvimento do modelo ou arquitetura, definindo princípios abrangentes que norteiam as tomadas de decisão ao longo do processo.

2.6.1 Engenharia baseada em modelo

A engenharia de sistemas baseados em modelos (*Model-based systems engineering* - MBSE) é uma metodologia de engenharia de sistemas que se concentra na criação e exploração de modelos de domínio como o principal meio de troca de informações entre

engenheiros, e não na troca de informações baseada em documentos (ESTEFAN et al., 2007).

Os modelos são usados durante o ciclo de desenvolvimento de um projeto e são colocados no centro do processo de desenvolvimento, podendo servir inclusive como documentação para troca de informação entre os desenvolvedores do projeto e para a realização de simulações baseadas em modelo (GIANNI; D'AMBROGIO; TOLK, 2014).

2.6.2 Abordagem de “baixo para cima” e “cima para baixo”

O princípio de projeto modular “de baixo para cima”, mais conhecida como abordagem *bottom-up*, é onde os elementos individuais de baixo nível do sistema são identificados primeiro. Estes elementos são então utilizados para formar subsistemas maiores através de um processo de composição e agregação, até que o sistema completo de nível superior seja formado.

Na abordagem de “cima para baixo” (*top-down*) é formulada uma visão geral do sistema, partindo de uma instância final para a inicial, como acontece na engenharia reversa. Cada nível vai sendo detalhado, do mais alto ao mais baixo, de forma a se chegar nas especificações dos níveis mais básicos do elemento abordado (SCHACH, 2007).

Em geral, as arquiteturas que empregam uma abordagem *bottom-up* tendem a ser mais flexíveis que as que utilizam uma abordagem *top-down*, permitindo a reutilização dos componentes de baixo nível em uma gama maior de aplicações.

2.6.3 Arquitetura baseada em serviços

A distribuição dos componentes de um sistema através de serviços conectados na web tem sido uma tendência recente em várias áreas da computação e automação. Isso se deve principalmente devido a flexibilidade na criação de novas aplicações que utilizem os ser-

viços e a total abstração quanto as especificações do dispositivo computacional (hardware, sistema operacional e a localização geográfica) que hospeda o serviço, possibilitando que ele implemente o conceito de “serviço na nuvem”. Para implementar esse serviço na nuvem, a seguir será visto o conceito de *Web Services*.

Web Services

Serviços web (*Web services*) são definidos como unidades modulares de aplicações lógicas que provêm funcionalidades para outras aplicações na rede (SRIVASTAVA; KOEHLER, 2003). Um serviço web é um método de comunicação que permite dois sistemas de software trocarem dados através da internet. Ele atua como uma camada de abstração, separando detalhes de programação (como linguagem de programação e plataforma) da forma como a aplicação é invocada (SNELL; TIDWELL; KULCHENKO, 2001). Portanto, programas em diferentes linguagens e plataformas podem se comunicar através do uso desta tecnologia.

Este tipo de tecnologia tem sido empregado em alguns trabalhos em aplicações industriais, como descrito em (MORA et al., 2012), motivado pelo fato de que um serviço web pode ser visto como um sistema de software projetado para suportar interações máquina-máquina sobre a rede de forma interoperável. Muitas empresas industriais usam múltiplos sistemas de software para gerenciamento, sendo que eles precisam trocar dados entre eles.

Os dois principais protocolos de serviços web são *Simple Object Access Protocol* (SOAP) e *Representational State Transfer* (REST). Esses protocolos serão caracterizados por:

- SOAP (*Simple Object Access Protocol*): Arquiteturas baseadas em SOAP trocam dados através de mensagens codificadas no formato *Extensible Markup Language* (XML) através do protocolo *Hypertext Transfer Protocol* (HTTP). Conjuntos de

serviços SOAP podem ser definidos através de arquivos *Web Service Definition Language* (WSDL), que são arquivos XML padronizados de acordo com uma gramática especificada pela W3C (SHI, 2006).

- REST (*Representational State Transfer*): REST foi introduzido como um estilo de arquitetura para sistemas distribuídos escalonáveis e sistemas de grande escala. Dado que a vasta maioria dos serviços web do tipo REST não são descritos usando WSDL, não é possível reusar clientes existentes que requerem esta descrição. REST é baseado em quatro princípios de arquitetura (PAUTASSO; ZIMMERMANN; LEYMANN, 2008):
 - i. Identificação de recursos através de URI: expõem recursos que identificam os alvos para interação com seus clientes.
 - ii. Interface uniforme: recursos podem ser manipulados usando quatro operações: PUT (criar), GET (ler), POST (atualizar), DELETE (apagar).
 - iii. Mensagens auto-descritivas: os recursos podem ser acessados através de vários formatos (ex. HTML, XML, plain text, etc).
 - iv. Interações por *hyperlinks* com estado: a interação com um recurso é sem estado.

2.7 AutomationML

O AutomationML (AML) é uma linguagem de modelagem que possui o formato de dados baseado em XML especialmente projetado para a troca de informações de engenharia de uma planta industrial. O formato pode ser usado para interconectar ferramentas de engenharia de diferentes disciplinas, desde a construção da planta industrial até o pro-

jeto mecânico e elétrico. O AutomationML não define um novo formato de arquivo, ele usa padrões já existentes e define regras para usá-los e combiná-los. Sendo assim, o arquivo principal criado pelo AutomationML é um arquivo no formato CAEX (IEC 62424)(SCHLEIPEN; DRATH; SAUER, 2008) sendo que outros formatos de arquivos podem ser agregados. Como por exemplo, o tipo de arquivo Collada(BARNES; FINCH, 2008), que é um formato de dados usado para descrever a geometria dos elementos e pode ser agregado ao CAEX. Também pode ser agregado o formato PLCOpenXML (PLCOPEN, 2013), que pode ser usado para guardar informações de comportamento (Figura 7). O CAEX abstrai conceitos de programação orientada a objetos, como classes e instâncias. Os objetos podem ser armazenados usando a estrutura de dados de árvores.

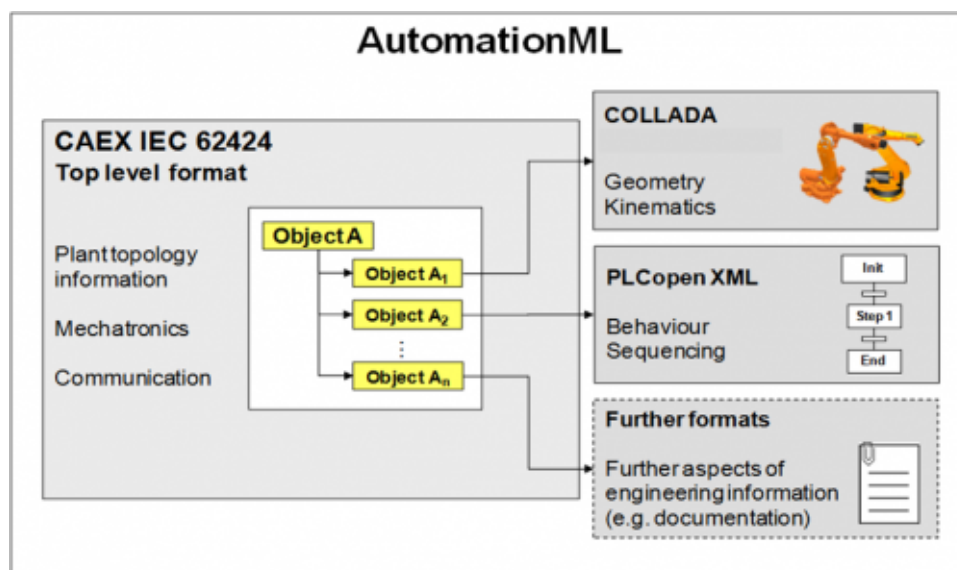


Figura 7: Estrutura do automationML. Fonte: (AUTOMATIONML, 2018)

2.7.1 Estrutura do AutomationML

O AutomationML possui alguns objetos pré-definidos e define diversas regras para a estruturação do modelo, sendo a maioria delas herdadas do formato CAEX. Cada modelo em AutomationML pode conter as seguintes bibliotecas:

- `InterfaceClassLibs`: é uma biblioteca usada para definir interfaces;

- `RoleClassLibs`: é usada para definições de função semântica dos objetos;
- `SystemUnitClassLibs`: é uma biblioteca que possui objetos genéricos e que podem ser reutilizáveis (instanciados);
- `InstanceHierarchy`: é onde pode-se criar instâncias do objetos definidos no `SystemUnitClassLibs`.

A Figura 8 descreve a estrutura básica de um arquivo AutomationML. O arquivo CAEX pode ser composto pelas três bibliotecas mencionadas acima e pela Instance Hierarchy. As setas desta figura representam uma relação “consiste de”.

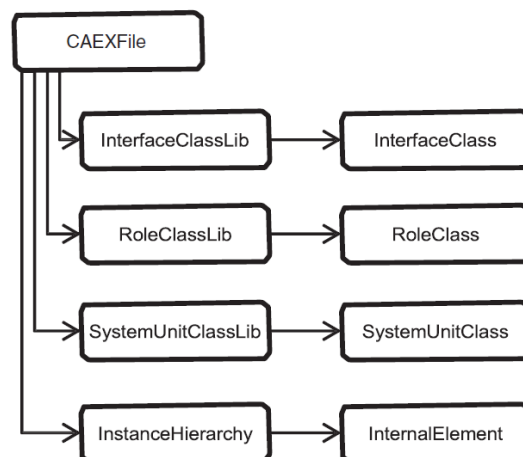


Figura 8: Estrutura do automationML. Fonte: (HENßEN; SCHLEIPEN, 2014)

Os principais elementos do AutomationML e todas as relações possíveis entre eles são mostrados na Figura 9 (HENßEN; SCHLEIPEN, 2014) e descritas a seguir:

- A organização do modelo em um tipo de estrutura de dados de árvore hierárquica descreve que cada elemento pode ter um filho do mesmo tipo (ver a seta `ChildElement` na figura).
- `SystemUnitClasses` e `RoleClasses` são definidos em uma estrutura de herança conforme ilustrado pela seta `RefBaseClassPath`.

- RoleClasses pode ser atribuído a SystemUnitClasses e *InternalElements*.
No último caso, existem duas maneiras diferentes de fazer a atribuição RoleClass usando SupportedRoleClasses ou RoleRequirement.
- SystemUnitClasses e *InternalElements* podem ter RoleClasses definidos.
No último caso, existem duas maneiras diferentes de fazer a atribuição: usando SupportedRoleClasses ou RoleRequirement.
- As setas restantes na Figura 9 descrevem as possíveis relações entre Internal Elements e SystemUnitClasses. Um InternalElement pode herdar de uma SystemUnitClass.
- No âmbito do AutomationML SystemUnitClasses são apenas *templates* de modelos e podem ser alterados após a instanciação.
- Uma System Unit Class pode ter subelementos em termos de Internal Elements. Esses subelementos (InternalElements) devem ser incluídos na instanciação.

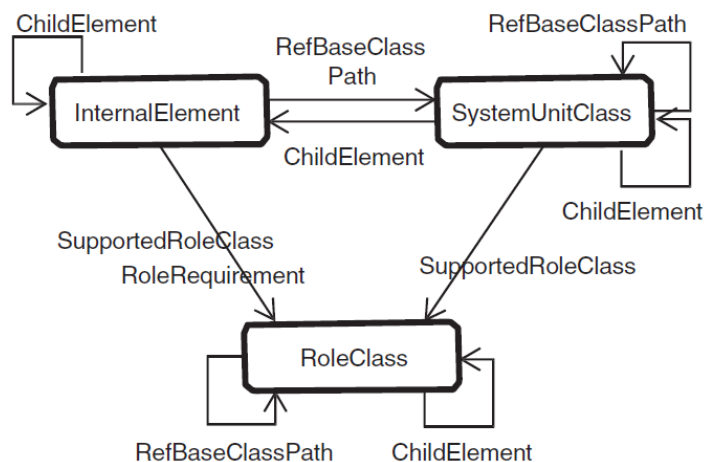


Figura 9: Principais elementos do AutomatioML. Fonte: (HENßEN; SCHLEIPEN, 2014)

A biblioteca `RoleClasses` descreve as funções de um objeto. Eles oferecem a possibilidade de especificar um objeto de maneira abstrata e independente do fabricante. A atribuição de uma `RoleClass` a um objeto (`SystemUnitClass` ou `InternalElement`) resulta na especificação de funções ou requisitos fundamentais para esse objeto.

A Figura 10 mostra como pode ser estruturada uma `RoleClass`, `SystemUnitClass` ou `InternalElement` (RC/SUC/IE). Cada uma dessas, pode conter atributos e interfaces (ver as setas `ChildElement`). As Interfaces são chamadas `ExternalInterface` e devem ser herdadas de uma `InterfaceClass`. Elas podem ser conectadas a outros elementos externos via `InternalLinks`. As `InterfaceClasses` são armazenadas hierarquicamente e devem ter relações de herança independentes.

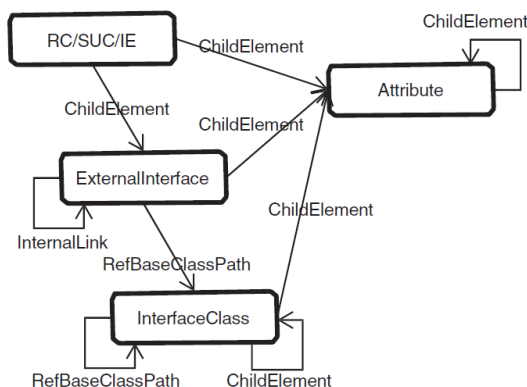


Figura 10: Detalhes dos principais objetos em AutomationML. Fonte: (HENßEN; SCHLEIPEN, 2014)

Para um melhor entendimento dessas estruturas e relações citadas acima, a seguir será apresentado um exemplo de modelagem utilizando o AutomationML.

2.7.2 Exemplo do AutomationML

Neste exemplo, suponha que tem-se os seguinte elementos: um PLC (*Programmable logic controller*), um módulo de entrada e saída, um motor, um sensor de luz e cabos para

conectar o motor e o sensor no módulo de E/S, ligados conforme ilustrado na Figura 11.

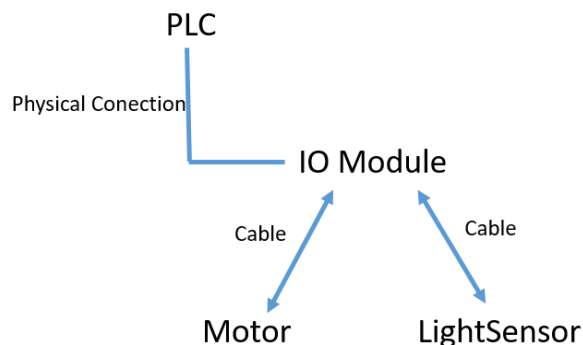


Figura 11: Esquema gráfico do exemplo usado para criar o modelo em AutomationML. [Fonte: autor]

Na Figura 12, a modelagem em AutomationML inicia-se pela definição das `SystemUnitClassLib`. Nela foram definidos os objetos **Motor**, **PLC**, **Cable**, **IOModule**, **LightSensor**. Para exemplificar o uso da biblioteca `RoleClassLib`, foi definido que o **LightSensor** tem uma `RoleClassLib` associada, que define que o **LightSensor** é um sensor, ou seja, a partir de agora, todos os objetos **LightSensor** tem a definição semântica do tipo sensor. Também pode ser visto na figura que em `RoleClassLib` existem várias `RoleLibs` que são pré-definidas no AutomationML, no entanto, o usuário pode definir suas `RoleLibs` conforme a sua necessidade.

Os objetos na SUC podem ter interfaces. Essas são definidas na `InterfaceClassLib`. O AML possui interfaces pré-definidas, mas o usuário também pode definir as suas interfaces. No exemplo, foi criada a interface **Plug**. O **Plug** é o conector de cada extremidade dos objetos do tipo **Cable**.

Após a criação dos objetos genéricos (SUCs) com suas *Roles* e *Interfaces* definidas, criam-se as instâncias dos objetos criados na SUCs no `InstanceHierachy` (IE). No IE, pode-se colocar quantas instâncias forem necessárias para a configuração que se deseja. No caso do exemplo ilustrado, existe uma instância do PLC, do motor, do módulo de E/S, do sensor e, duas instâncias de cabo. No IE, também associa-se os links entre as

interfaces. Por exemplo, um **Plug** do cabo 1 possui um link com o **Plug** do **IOModule**; a outra extremidade do **Plug** esta conectada no motor (as linhas pontilhadas azuis na figura ilustram os links). O mesmo serve para o outro cabo, no entanto, uma extremidade do cabo esta ligada no módulo e a outra extremidade no sensor de luz.

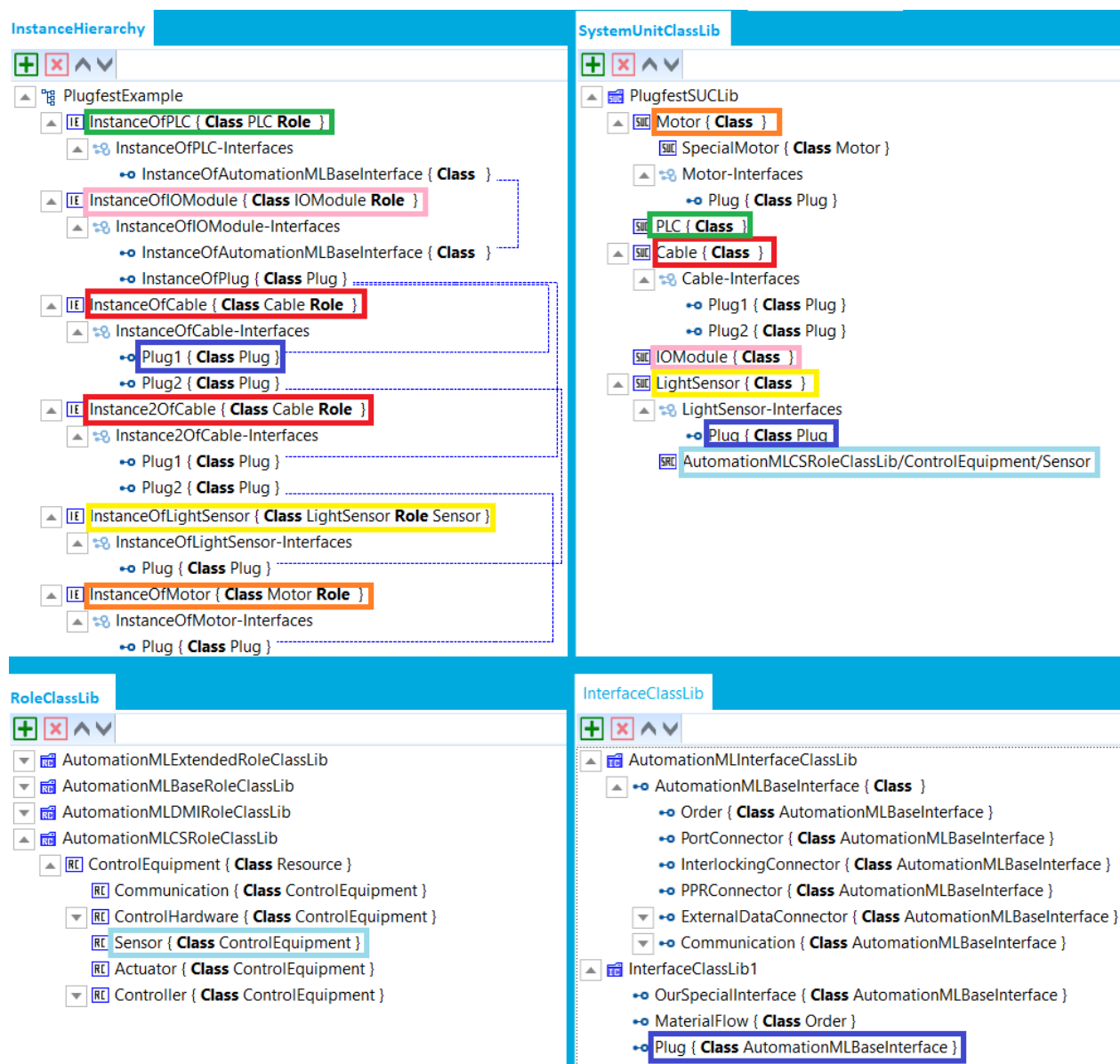


Figura 12: Exemplo da modelagem de um PLC, módulo de E/S, motor e um sensor em AML. [Fonte: autor]

2.8 Considerações finais

Este capítulo apresentou a fundamentação teórica para o entendimento e contextualização deste trabalho. Foi introduzido o conceito da Indústria 4.0, sistemas ciber-físicos, Internet das Coisas e *Big Data*. Em sistemas ciber-físicos, foi visto como os sistemas de automação se encaixam neste contexto, assim como, foi visto a relação entre digital twin e sistemas ciber-físicos. Além desses conceitos chaves, foi importante explicar sobre o ciclo de vida, sobre sistemas de manutenção inteligente e sobre estratégias de projeto. E, por fim, foi explicado o que são os Web Services e foi apresentada uma introdução sobre a linguagem de modelagem AutomationML usada neste trabalho.

O próximo capítulo apresentará as referências sobre o digital twin encontrada na literatura.

3 REVISÃO E DEFINIÇÃO DO DIGITAL TWIN

Este capítulo apresenta a revisão bibliográfica em relação ao digital twin. Inicialmente, será abordada a história do digital twin e as definições atribuídas a ele. Após, será mostrada a abordagem usada para a revisão da literatura e os destaques da revisão, seguido pela revisão do digital twin na indústria e suas aplicações, a sua relação com o ciclo de vida, e tópicos de estudo relevantes para o mesmo. Por fim, tem-se uma seção para descrever os trabalhos relevantes e relacionados ao estudo desta Tese.

3.1 Metodologia de Pesquisa

A metodologia de pesquisa utilizada foi uma revisão detalhada da literatura sobre o digital twin. Para isso, foi realizada uma pesquisa nas bases de dados *Scopus* e *Web of Science* procurando todas as publicações que tinham o termo “digital twin” no título, resumo ou nas palavras-chave. O tipo de publicação (artigos de revista, artigos de conferência e outros) não foi um critério de filtragem inicialmente e, o único idioma considerado foi o inglês. Verificou-se que as publicações de língua inglesa listadas no banco de dados Scopus até o momento¹, foram no total 158 e na base *Web of Science* foram 48 trabalhos publicados (na base *Web of Science* somente o título foi considerado). A Tabela 1

¹Pesquisa referente até a fevereiro de 2018

apresenta o número de publicações em cada ano, de cada base, referentes ao digital twin.

Tabela 1: Número de publicações referente ao termo digital twin nas bases de dados

Ano	Scopus	Web of Science
2004	2	-
2005	4	-
2010	1	-
2011	1	-
2012	10	-
2013	7	-
2014	2	2
2015	6	4
2016	23	13
2017	95	29
2018	7	-
Total	158	48

Apartir da análise inicial dessas publicações, o primeiro item explorado foi o conceito do termo digital twin e suas aplicações, que será introduzido na próxima seção. Depois, foi realizada uma segunda análise considerando o contexto deste trabalho: indústria 4.0 e sistemas ciber-físicos.

Com isso, verificou-se que as publicações de língua inglesa lançadas até 2018 e listadas no banco de dados Scopus, foram no total 151 trabalhos. Depois, uma terceira análise foi realizada, e somente publicações pertinentes ao tópicos deste trabalho (CPS e automação industrial) permaneceram. Com isso, 37 artigos foram selecionados como relevantes para o trabalho aqui proposto.

Primeiramente, foi analisado o propósito das publicações. As publicações entre 2012 e 2014, em sua maioria, são aplicadas no campo aeroespacial, afim de monitorar anomalias e realizar análises de saúde dos equipamentos. Após 2014, surgem pesquisas relacionadas a indústria, em particular, voltadas para a nova revolução industrial. As novas tecnologias, como CPS, abrem caminho para novos usos (além dos fins de diagnóstico e prognóstico de saúde dos equipamentos) e, trazem um novo significado do DT para o domínio dos sistemas de produção.

Durante a leitura de cada publicação, palavras de destaque (*Tags*) foram atribuídas a cada artigo e, com isso, foi possível categorizá-los por aplicação/uso do digital twin, como por exemplo: tópico de estudo (arquitetura, simulação, etc); tecnologias usadas e, termos relevantes encontrados. Com isso, a Tabela 2 foi gerada, onde cada linha contém a publicação em questão e cada coluna contém as *Tags* extraídas. A tabela 2 apresenta uma visão geral das publicações, suas aplicações e, tecnologias usadas para o digital twin de uma maneira ampla.

Com base nesta tabela, chega-se a algumas considerações que serão analisadas no restante deste capítulo. Para isso, a pesquisa foi classificada nos seguintes tópicos: história do digital twin, tópicos de estudo, aplicações na indústria (fábrica virtual, produto, simulação, predição), arquitetura e modelagem do digital twin.

Tabela 2: Principais *Tags* selecionadas para cada publicação.

Ano	Referência	Architecture	AutomationML	Big Data	BOL	Cloud	Communication	CPS	Database	Industrie 4.0	Manufacturing	PLM	Predictive	Product Lifecycle	Quality Assurance	Simulation	Tracking
2014	(GRIEVES, 2014)																
2015	(LEE, 2015)	x						x			x						
2015	(RIOS et al., 2015)											x					
2015	(ROSEN et al., 2015)							x			x			x		x	
2016	(GABOR et al., 2016)	x														x	
2016	(RIOS et al., 2016)													x			
2016	(REID; RHODES, 2016)							x									
2016	(WEYER et al., 2016)	x	x					x		x						x	
2017	(ALAM; SADDIK, 2017)	x				x		x									
2017	(ANGRISH et al., 2017)	x						x	x								
2017	(BORODULIN et al., 2017)					x				x	x						
2017	(BOTTANI et al., 2017)							x								x	
2017	(CIAVOTTA et al., 2017)							x		x							
2017	(FERGUSON; BENNETT; IVASHCHENKO, 2017)															x	
2017	(IGLESIAS et al., 2017)															x	
2017	(LEE; JIN; LIU, 2017)			x				x		x			x				
2017	(KNAPP et al., 2017)							x		x							
2017	(KONSTANTINOV et al., 2017)							x				x					
2017	(LI et al., 2017)					x		x									
2017	(MAGARGLE et al., 2017)												x			x	

Tabela 2 continuação da página anterior

Ano	Referência	Architecture	AutomationML	Big Data	BOL	Cloud	Communication	CPS	Database	Industrie 4.0	Manufacturing	PLM	Predictive	Product Lifecycle	Quality Assurance	Simulation	Tracking
2017	(MORENO et al., 2017)									x						x	x
2017	(SCHLEICH et al., 2017)											x			x		
2016	(SCHLUSE; ROSSMANN, 2016)															x	
2017	(SCHLUSE; ATORF; ROSSMANN, 2017)															x	
2017	(SODERBERG et al., 2017)				x										x	x	
2017	(TAMMARO et al., 2017)																x
2017	(TAO et al., 2017)			x				x			x			x		x	
2017	(TAVARES et al., 2017)									x						x	
2017	(UHLEMANN; LEHMANN; STEINHILPER, 2017)							x		x	x						
2017	(UM; WEYER; QUINT, 2017)		x													x	
2017	(YUN; PARK; KIM, 2017)	x					x	x								x	
2017	(ZHANG et al., 2017)				x	x	x	x	x	x	x			x		x	
2018	(QI; TAO, 2018)			x				x			x			x		x	
2018	(CAI et al., 2017)						x	x			x						
2018	(SCHLUSE et al., 2018)									x						x	
2018	(SIERLA et al., 2018)		x													x	
2018	(ZHUANG; LIU; XIONG, 2018)			x				x		x	x						x

3.2 Histórico

O termo digital twin foi definido por Dr. Michael Grieves na Universidade de Michigan em 2002, no contexto do *Product Lifecycle Management*. Após algum tempo, em seu artigo (GRIEVES, 2005), ele apresentou o conceito de digital twin como uma representação virtual do produto que foi fabricado e sua idéia inicial era comparar o produto produzido com o que foi projetado, criando um *loop* entre a fase de projeto e a fase de execução e manufatura do produto.

No entanto, o digital twin não foi usado somente para comparação na fase de projeto e manufatura, ele foi definido como uma representação virtual de um produto físico que contém toda e qualquer tipo de informação sobre este, sendo regularmente atualizado com informações vindas do meio físico. Para isso, ele deve estar conectado com a sua parte física de alguma maneira, permitindo a coleta e transferência de dados do físico para o digital. Este conceito é mostrado na Figura 13 que ilustra que o digital twin contém três partes principais:

- a) as entidades físicas, no mundo físico,
- b) os modelos virtuais, no mundo virtual e
- c) dados conectados que unem os dois mundos.

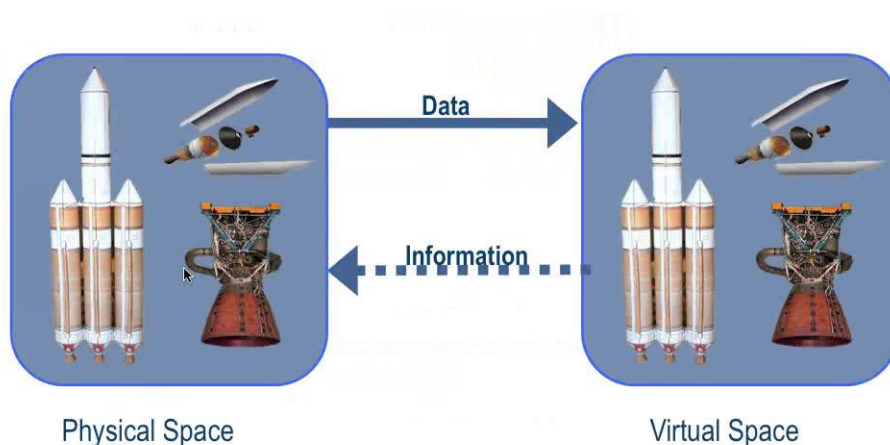


Figura 13: Conceito do twin model. Fonte:(GRIEVES, 2005)

Em 2010 surgiu outra definição do digital twin nas pesquisas realizadas pela NASA. Sua definição era: “uma simulação integrada multi-física, multi-escala e probabilística de um veículo ou sistema que usa os melhores modelos físicos disponíveis, atualizações de sensores, histórico da frota, etc., para espelhar a vida do dispositivo físico. É ultra-realista e pode considerar um ou mais sistemas de veículo importantes e interdependentes”. Esta definição apareceu pela primeira vez como um rascunho em (SHAFTO et al., 2010) e então, foi lançado um documento oficial intitulado de *Roteiro de Modelagem, Simulação, Tecnologia da Informação e Processamento da NASA em 2010* (SHAFTO et al., 2012) . A partir desse momento, pesquisadores aeroespaciais começaram a se referir ao roteiro como um documento de referência para definir o digital twin, como foi afirmado em (GLAESSGEN; STARGEL, 2012).

Fica evidente, que o escopo da definição original do digital twin era refletir a vida útil dos veículos aéreos com uma série de sub-modelos integrados que refletiam diferentes aspectos e sistemas. Para isso, considerava dados dos sensores e mantinha um histórico dos mesmos e, realizava as interações do mundo digital com o mundo real. Em trabalhos de pesquisa subsequentes, surgiram outros aspectos, como a visão do ciclo de vida (TUEGEL, 2012), o controle de requisitos para a missão do veículo (GOCKEL et al., 2012)(TUEGEL, 2012) e, o uso do digital twin para atividades de prognóstico e diagnóstico (REIFSNIDER; MAJUMDAR, 2013). Essas características permaneceram como características fundamentais do conceito em obras futuras.

Em 2014, Grieves redefiniu o digital twin como uma maneira distribuída e descentralizada de gerenciar informações de produtos em cada nível do ciclo de vida (GRIEVES, 2014). Ainda, o digital twin também passa a ser considerado como a virtualização de entidades físicas que ele representa.

Já em 2015, com o trabalho de Ríos (RIOS et al., 2015), a definição de digital twin compreendeu um “Produto” genérico, abrindo o caminho para o uso de tal conceito em outros setores, e não apenas para veículos aéreos, mesmo que seu trabalho ainda estivesse inserido em pesquisas sobre estruturas de aeronaves.

Outros trabalhos surgiram em outros setores: ao lado da pesquisa no setor aeroespacial, em 2013, apareceu o primeiro trabalho de pesquisa sobre o digital twin no setor de manufatura proposto por Lee (LEE et al., 2013). Este trabalho considerava que o digital twin seria a contrapartida virtual dos recursos de produção, e não apenas do produto, criando um debate sobre o papel do digital twin em ambientes de manufatura avançados, como por exemplo, no setor da Indústria 4.0 com seu núcleo tecnológico, análise de dados e plataformas de computação em nuvem (LEE et al., 2013).

Em Negri (NEGRI; FUMAGALLI; MACCHI, 2017), um resumo dos principais conceitos para o digital twin pode ser encontrado. A Tabela 3 apresenta este resumo adaptado, com outras definições que surgiram após essa publicação.

Tabela 3: Definições do digital twin encontradas na literatura. Adaptado de (NEGRI; FUMAGALLI; MACCHI, 2017)

No	Ref	Ano	Definição do digital twin
1	(GRIEVES, 2005)	2002	É definido como uma maneira distribuída e descentralizada de gerenciar informações do produto em cada nível do seu ciclo de vida
2	(SHAFTO et al., 2010)	2010 2012	Uma simulação integrada multi-física, multi-escala e probabilística integrada de um veículo ou sistema que usa os melhores modelos físicos disponíveis, atualizações de sensores, histórico da frota, etc., para espelhar a vida de seus gêmeos físicos. O digital twin é ultra-realista e pode considerar um ou mais sistemas de veículos importantes e interdependentes
3	(TUEGEL, 2012)	2012	Um modelo do início ao fim de vida da estrutura de uma aeronave com a habilidade de cumprir os requisitos da missão, incluindo submodelos eletrônicos, controles de vôo, sistema de propulsão e outros subsistemas
4	(GOCKEL et al., 2012)	2012	Modelo de computador ultra-realista, do início ao fim de vida de uma estrutura de aeronave que é usado para avaliar a capacidade da aeronave de cumprir os requisitos da missão
5	(LEE et al., 2013)	2013	Modelo acoplado da máquina real que opera na plataforma de computação em nuvem e simula as condições de saúde usando um conhecimento integrado de algoritmos analíticos orientados por dados, junto com outros conhecimentos físicos disponíveis
6	(REIFSNIDER; MAJUMDAR, 2013)	2013	Modelos de alta fidelidade dos materiais e estruturas que controlam a vida de um veículo
7	(MAJUMDAR; HAIDER; REIFSNIDER, 2013)	2013	Modelo estrutural que incluirá dados quantitativos de características de materiais com alta sensibilidade
8	(ROSEN et al., 2015)	2015	Modelos muito realistas do estado atual do processo e seu comportamento com o meio no mundo real
9	(RIOS et al., 2015)22	2015	Modelo digital do produto físico
10	(BIELEFELDT; HOCHHALTER; HARTL, 2015)	2015	Modelos computacionais multi-físicos ultra-realistas associados a cada aeronave única e combinados com histórias de aeronaves conhecidas
11	(BAZILEVS et al., 2015)	2015	Modelo estrutural de alta fidelidade que incorpora danos por fadiga e apresenta uma parte digital bastante completa do sistema estrutural de interesse
12	(SCHLUSE; ROSSMANN, 2016)	2016	Substitutos virtuais de objetos do mundo real que consistem em representações virtuais e capacidades de comunicação criando objetos submissos atuando como nós inteligentes dentro da internet de coisas e serviços
13	(CANEDO, 2016)	2016	Representação digital de um objeto do mundo real com foco no próprio objeto
14	(GABOR et al., 2016)	2016	Simulação do próprio objeto físico para prever os futuros estados do sistema
15	(SCHROEDER et al., 2016)	2016	Representação virtual de um produto real no contexto dos sistemas ciber-físicos
16	(KRAFT, 2016)	2016	Uma simulação integrada multi-física, multi-escala, probabilística de um sistema construído, habilitado por uma <i>Digital Thread</i> , que usa os melhores modelos disponíveis, informações de sensores e dados de entrada para espelhar e prever atividades e o desempenho ao longo da vida de seu correspondente gêmeo físico
17	(BAJAJ; COLE; ZWEMER, 2016)	2016	Um modelo de sistema unificado que pode coordenar modelos de arquitetura, mecânica, elétrica, software, verificação e outros modelos específicos ao longo do ciclo de vida do sistema, federando modelos em várias ferramentas de fornecedores e repositórios controlados por configuração
18	(SCHLEICH et al., 2017)	2017	A visão entre a relação bidirecional entre o artefato físico e o conjunto dos seus modelos virtuais
19	(ALAM; SADDIK, 2017)	2017	Cópia exata de um sistema físico que representa fielmente todas as suas funcionalidades

3.3 Aplicações do digital twin

No início deste capítulo foi comentado que nas publicações iniciais do digital twin, o termo era usado para comparar o que foi planejado com o que foi produzido em relação aos produtos manufaturados. E, que entre 2012 e 2013, as aplicações destacavam-se no setor aeroespacial. No entanto, considerando somente as publicações no setor industrial, destacam-se as seguintes afirmações:

1. Os tipos de uso do digital twin encontrados na literatura são para realizar o monitoramento dos dispositivos físicos e realizar a predição de falhas (LEE; JIN; LIU, 2017) (MAGARGLE et al., 2017) (CAI et al., 2017). Além disso, destaca-se o uso para realizar simulações (a coluna ‘Simulação’ é a mais marcada na Tabela 2)
2. Em geral os setores industriais destacados foram: a indústria automotiva (WEYER et al., 2016) e o setor aeroespacial. E, alguns casos específicos como: *Automated Guided Vehicles* (AGV)(BOTTANI et al., 2017), manufatura aditiva (KNAPP et al., 2017);
3. Alguns artigos apresentam uma arquitetura para o digital twin, como por exemplo: (ALAM; SADDIK, 2017) (ANGRISH et al., 2017) (GABOR et al., 2016) (WEYER et al., 2016);
4. Alguns artigos afirmam a importância do digital twin durante o ciclo de vida, no entanto a maioria deles refere-se a fase inicial do ciclo de vida, com o propósito de garantir a qualidade do que foi produzido com o que foi projetado (ver na seção 3.4);
5. Também é apresentado o uso do digital twin para a virtualização do chão de fábrica (*shop-floor*) (TAO et al., 2017);

6. Em relação a modelos, somente (WEYER et al., 2016) aborda modelagem, no entanto, os autores apenas apresentam os desafios de modelagem e simulação no contexto de *Smart Factories*,
7. Em relação a Gerenciamento de dados: alguns citam a relação entre o termo *Big Data* e digital twin (QI; TAO, 2018). Rios (RIOS et al., 2016) usa um software comercial da Dassault para estruturar os dados do digital twin de aeronaves.

3.4 Digital twin no ciclo de vida do produto

Durante a revisão da literatura sobre o digital twin, ficou claro que muitos trabalhos destacam a importância do digital twin no ciclo de vida. Dentre os mais importantes, destacam-se:

- Rosen (ROSEN et al., 2015) acredita que o digital twin é uma das tecnologias para a realização da Indústria 4.0. Ele afirma que o digital twin permite que a informação criada em cada etapa do ciclo de vida do produto deve estar disponível para etapas subsequentes. O artigo contextualiza o digital twin no contexto da indústria 4.0 e CPS e, afirma que a modularidade e autonomia são tópicos importantes. Isso significa que o sistema de produção e as unidades de produção reagem de forma autônoma a novas ordens, modificando as prioridades de produção durante a operação. Em tal situação, é necessário um conhecimento abrangente sobre o estado atual do sistema de produção e sobre suas próprias capacidades. O primeiro tópico (modularidade) é abordado principalmente por meio da coleta, armazenamento e processamento de todos os dados disponíveis (sensores) no sistema de produção, juntamente com as condições de operação. No entanto, nem todas as quantidades desejáveis podem ser medidas diretamente e as previsões para o comportamento

futuro com base apenas em dados de operação são delicadas, especialmente se o sistema de produção flexível for considerado. A combinação dos dados da vida real com os modelos de simulação do projeto permite, do outro lado, fornecer boas previsões com base nos dados realistas. Isso aborda as oportunidades de usar a simulação para auxiliar os sistemas a oferecer suporte a operadores e planejadores durante a operação normal, bem como para manutenção e serviço por meio de previsão baseada em simulação. O conceito de digital twin permite exatamente tal procedimento, pois todos os modelos e todos os dados estão disponíveis em um ambiente consistente e bem alinhado.

- Tao (TAO et al., 2017) apresenta alguns problemas relacionados aos dados de um produto durante todo o seu ciclo de vida. Ainda, afirma que falta convergência entre o espaço virtual e físico, e que os dados no ciclo de vida do produto são isolados, fragmentados e estagnados, o que é inútil para as empresas industriais. Além disso, os dados, muitas vezes, ficam duplicados em diferentes fases do ciclo. O autor cita que o digital twin tem o potencial para resolver os problemas em PLM e o artigo enfatiza para o uso do digital twin para aplicações em projeto, manufatura, uso e manutenção do produto. Em projeto, o uso do digital twin colabora na concepção, detalhes e virtualização do produto. Na manufatura o digital twin da fábrica ajuda na organização e otimização da produção. E, na fase de uso e manutenção, o digital twin, baseado em dados reais, virtuais e dados conectados, tem o potencial de melhorar o nível de inteligência do produto, reduzir a ocorrência de falhas e inatividade do produto, melhorar as atividades de manutenção e melhorar a eficiência da produção.
- Rios (RIOS et al., 2015) apresenta o ciclo de vida de aeronaves e também destaca a

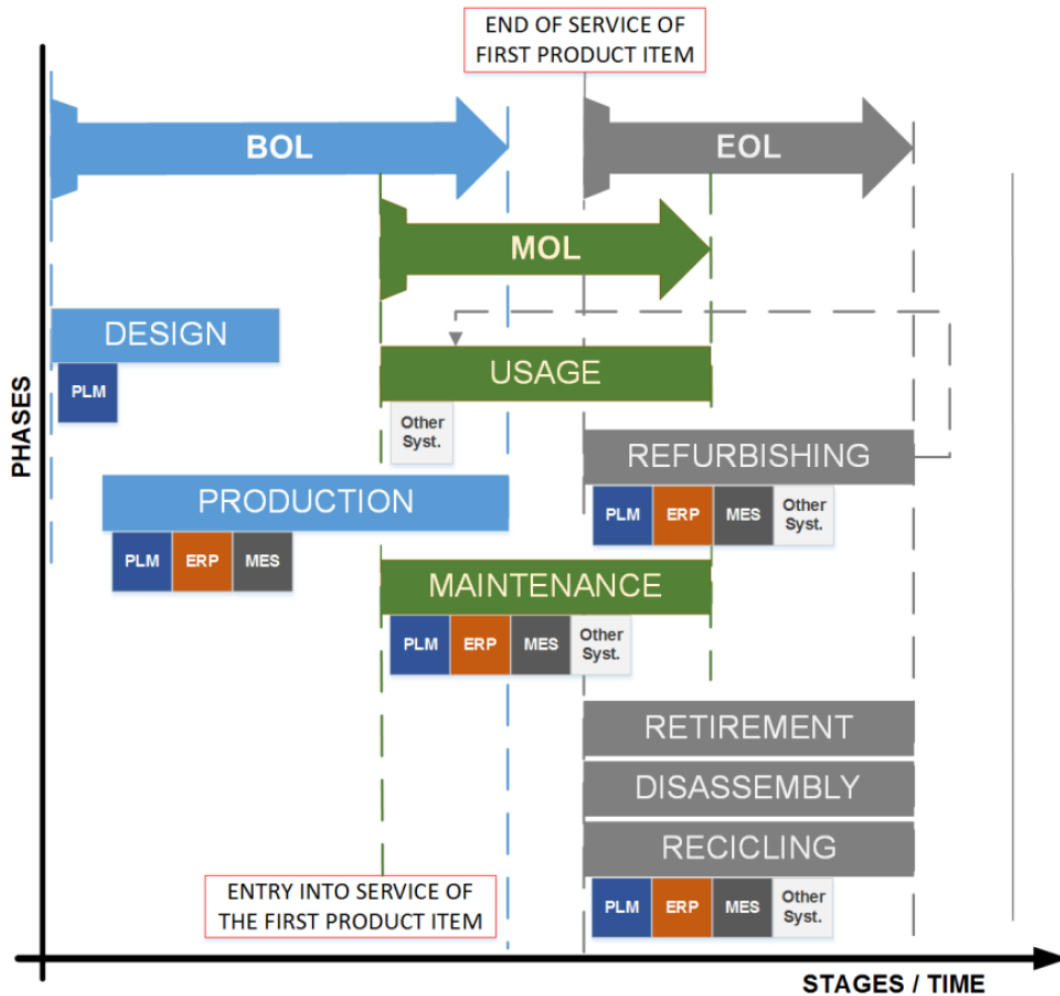


Figura 14: Fases do ciclo de vida e principais sistemas de software apresentado em (RIOS et al., 2015). Fonte: (RIOS et al., 2015)

importância do uso do digital twin no gerenciamento das informações das aeronaves. O artigo comenta sobre tópicos que afetam a criação do digital twin: identificação do produto, ciclo de vida, informações do produto, configurações do produto e modelos do produto. Os autores citam os principais sistemas de suporte para o ciclo de vida e a Figura 14 mostra a alocação destes sistemas ao longo das fases do ciclo de vida. A aplicação usada no artigo foi para produtos no setor aeroespacial.

- (SCHLEICH et al., 2017) não aborda diretamente o ciclo de vida, mas a pesquisa é aplicada para o gerenciamento de variações geométricas entre as fases de projeto e manufatura do ciclo de vida (BOL). É proposto um modelo de referência baseado

em *Skin Model Shapes* (que é o digital twin do produto nas fases de projeto e manufatura). O *Skin Model Shapes* é baseado em esquemas de representação geométrica discreta. Propriedades como escalabilidade, interoperabilidade, expansibilidade e fidelidade são apresentadas para o modelo. Assim como operações de conversão, composição, decomposição e avaliação podem ser realizadas no modelo.

- Em (KONSTANTINOV et al., 2017) é apresentada uma ferramenta, chamada Vue-One, para integrar e dar suporte a todas as fases do ciclo de vida. O estudo mostrou o uso da ferramenta na fase de construção de uma máquina usando um modelo comum. As ferramentas de engenharia tem sido fundamentais para integrar usuários e elementos do sistema. Trabalhos futuros devem ser em explorar o modelo nas outras fases do ciclo de vida.
- (ZHANG et al., 2017) diz que um dos desafios do digital twin é obter um modelo que combine o gerenciamento do ciclo de vida, sistemas de execução de manufatura (MES) e, sistemas de gerenciamento de operações (*Operations Management Suite-OMS*).

3.5 Tópicos de estudo

Durante a análise da literatura, Ríos (RIOS et al., 2015) apresentou os requisitos ou tópicos de estudo do digital twin. A seguir, será apresentado cada um deles e uma classificação de publicações em cada tópico:

- **Identificação:** o produto necessita de uma identificação global que conecte o produto físico a sua representação digital. Algumas tecnologias que podem ser usadas para isso são: RFID tags e Electronic Product Code (EPC) (BROCK, 2001). Eles

forneem uma identidade única para cada objeto físico em qualquer lugar do mundo e por todo o seu ciclo de vida.

- **Gerenciamento de dados:** dados e informações do produto são gerados e crescem ao longo do seu ciclo de vida (BOL, MOL and EOL - seção 2.5). O tamanho de armazenamento dos dados podem atingir grandes proporções e com isso, surgem as pesquisas relacionadas a *Big Data Management*. A partir da análise desses dados, surgem alguns benefícios, como predição de falhas em equipamentos, previsão do tempo de vida, etc.
- **Modelagem do produto:** tipos diferentes de modelos são criados durante o ciclo de vida do produto. E, muitas vezes, estes modelos são interoperáveis entre eles, levando a problemas. Os modelos necessitam de uma maneira de falar a mesmas linguagem.
- **Interfaces Homem Máquina:** tendo tanta informação para lidar e, em diferentes repositórios, faz-se necessário uma maneira de mostrar a informação certa para o usuário certo. O digital twin pode fornecer informação para todos os tipos de usuários envolvidos em todas as fases do seu ciclo de vida. Com isso, é necessário o desenvolvimento de Interfaces Homem Máquina (IHM) para o digital twin.
- **Comunicação:** é necessário ter um meio de comunicação para armazenar os dados e extrair informações para as interfaces IHM. Deve-se considerar o fato de que dados podem ser armazenados em diferentes repositórios e em qualquer lugar do mundo, considerando principalmente a tecnologia da internet das coisas. Portanto, essa comunicação precisa ser segura e extrair dados de qualquer lugar e a qualquer hora.

Além destes tópicos, deve-se considerar que o digital twin possui informações estáticas e dinâmicas. As informações estáticas podem ser: dimensões geométricas, lista de materiais, lista de processos de fabricação, etc. As informações dinâmicas são aquelas que mudam ao longo de sua vida, como o valor de um sensor, por exemplo.

Além disso, o digital twin pode ter inteligência. Por exemplo, um produto inteligente pode reter informação sobre si mesmo e ser capaz de tomar decisões sobre o que fazer no futuro. Como citado em (HRIBERNIK; WUEST; THOBEN, 2013), um produto inteligente tem as seguintes características:

- Necessita uma identificação global;
- É capaz de se comunicar no ambiente;
- Pode retrain e guardar informação sobre si mesmo;
- É capaz de participar na tomada de decisões que são relevantes para o seu futuro.

3.6 Trabalhos relacionados

O propósito desta seção é abordar os trabalhos relacionados e relevantes para esta Tese. Cada publicação selecionada será resumida e comentada.

(CAI et al., 2017)

Este artigo apresenta técnicas para construir digital twins de máquinas usando ferramentas de integração de dados de sensores e fusão de informações de usinagem (Figura 15). A técnica apresentada pode ser usada para integrar dados de fabricação e sensores em máquinas virtuais para melhorar sua precisão e capacidade. Os sensores são usados para registrar os principais dados de usinagem e ferramentas analíticas são usadas para

extrair os perfis e características de uma máquina. Resultados experimentais preliminares mostraram que os dados coletados foram úteis para monitorar a condição de operação e demonstrar funcionalidades e características de fabricação da máquina sob teste. Esforços futuros de pesquisa serão focados na acessibilidade de dados com a nuvem. A técnica apresentada que usa dados e informações sensoriais pode ser inferida como sendo um dos principais blocos de construção para o desenvolvimento da fabricação cibernética.

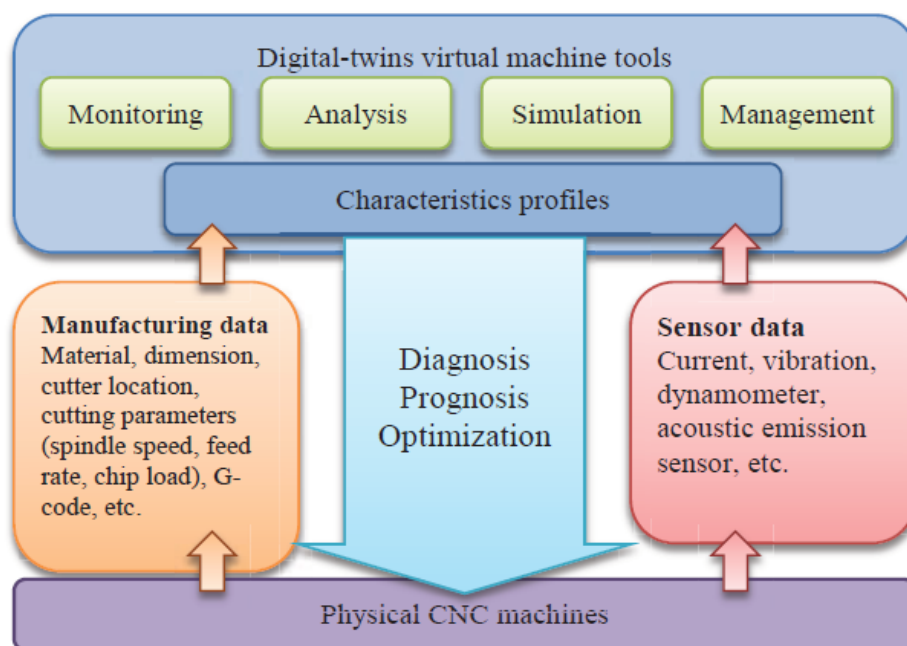


Figura 15: Esquema de construção da ferramenta virtual para digital twins usando integração de sensores e fusão de informações de manufatura. Fonte: (CAI et al., 2017)

O artigo possui o foco em aquisição e análise de dados para ajudar na construção dos modelos do digital twin, extraindo características pertinentes das máquinas e equipamentos. Os sensores são usados para registrar os principais dados de usinagem e ferramentas analíticas são usadas para extrair os perfis e características de uma máquina. Para isso, o autor mostra um diagrama UML para o modelo de dados. Os dados são armazenados em um banco de dados PostgreSQL na máquina local. Para a extração de características usa um método de aprendizado de partículas (*Particle Learning method*) desenvolvido em trabalhos anteriores do próprio autor.

(UM; WEYER; QUINT, 2017)

O artigo usa o AutomationML pra criar um modelo para módulos de produção que compõe uma *SmartFactory* e realizar simulações. Este modelo é usado para simular as configurações possíveis de produção da fábrica para alcançar flexibilidade de produção e customização. Descreve como é a comunicação para que um novo módulo possa ser inserido na planta e, como um módulo pode se recuperar quando é desconectado e/ou interrompido. Para isso, cada módulo possui estados possíveis (*sleep, isolated, configured, loaded*).

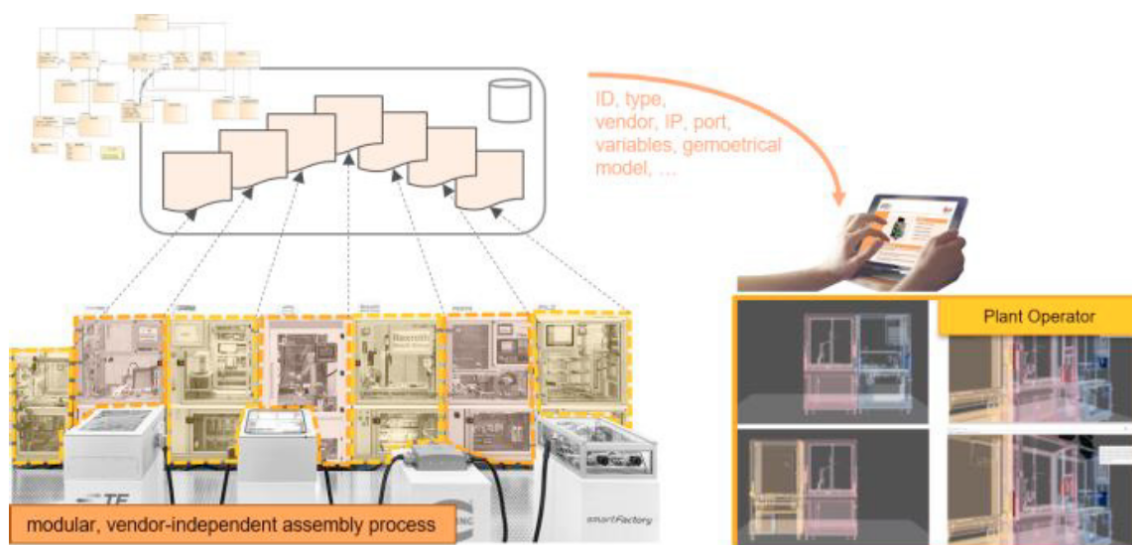


Figura 16: Sincronização da dados do módulos com o uso de representação digital. Fonte: (UM; WEYER; QUINT, 2017)

O estudo de caso foi implementado usando o *DyVisual*, um ambiente para modelagem gráfica. A partir do modelo descrito em AutomationML, um ambiente gráfico é criado para os usuários terem acesso às informações do digital twin como por exemplo: sua identificação, tipo, endereço IP e porta de comunicação, variáveis, modelo geométrico, etc. Assim, usuários podem ser informados sobre o estado das máquinas, configurações e quais máquinas estão conectadas no momento. Um exemplo ilustrativo do trabalho é visto na Figura 16. Trabalhos futuros serão realizados para integrar o modelo em uma

ferramenta de simulação da planta e na aplicação de sensores para manutenção preditiva. O autor não cita como o gerenciamento de dados é realizado, nem fornece detalhes de identificação, somente que existe um ID para os módulos.

(SIERLA *et al.*, 2018)

Esta publicação (SIERLA *et al.*, 2018) é a mais recente sobre digital twins e, tem entre suas palavras chaves os termos digital twin e AutomationML. A proposta é usar um modelo em AutomationML para o planejamento automático de montagem (*assembly*) de produtos, baseado na sua descrição. O digital twin é derivado da descrição do produto e tem o papel de orquestrar os recursos de produção em uma célula de manufatura, realizar simulações de montagem e, possibilitar o acesso a partes do modelo (Figura 17).

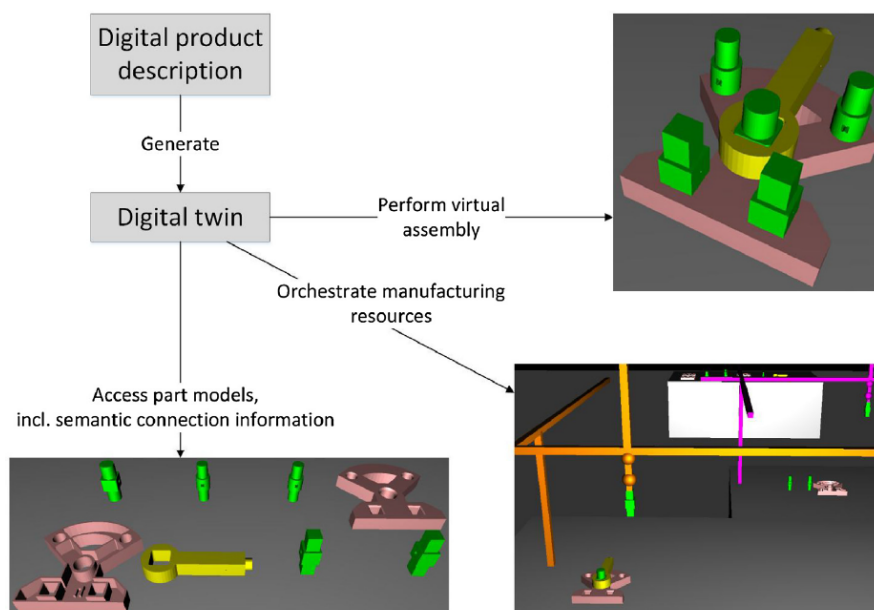


Figura 17: Idéia geral do trabalho de Sierla. Fonte: (SIERLA *et al.*, 2018)

Os autores apresentam um *framework* conceitual usando diagramas em UML. A implementação é feita em um ambiente 3D de simulação usando o AutomationML para a descrição do produto. Em relação a comunicação de dados entre o físico e o digital twin, conclui-se que essa não existe, pois a aplicação é para simulações, sem conexões com o

mundo real.

(ALAM; SADDIK, 2017)

Alam (ALAM; SADDIK, 2017) propõe uma arquitetura de referência e afirma a importância da ‘nuvem’ para sistemas ciber-físicos. O artigo contribui com uma descrição analítica para modelar uma arquitetura de referência para sistemas ciber-físicos baseados em nuvem (C2PS), onde cada elemento físico acompanha um elemento hospedado na nuvem, que é o digital twin.

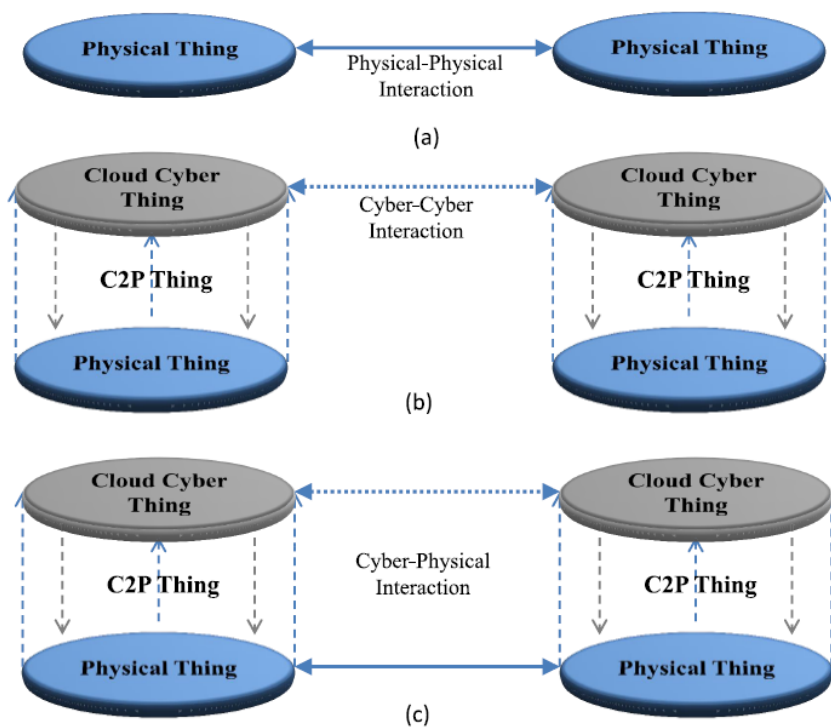


Figura 18: Tipos de comunicação, proposto em (ALAM; SADDIK, 2017).

O ponto forte do artigo é a caracterização dos tipos de computação e as interações existentes na arquitetura. Os tipos de comunicação podem ser: entre os elementos físicos, entre os elementos virtuais e híbrida (físico-virtual ou virtual-físico), conforme visto na Figura 18. A computação refere-se ao processamento, que pode ser realizado no dispositivo físico ou no ciber. Para isso, implementa inteligência através do uso de Redes

Bayesianas e Lógica Fuzzy e usa essa inteligência para decidir seu modo de operação dependendo do contexto do sistema e, então, a rede decide qual modo de comunicação selecionar (físico-físico, ciber-ciber, físico-ciber).

O artigo também propõe uma composição hierárquica dos digital twins. Sendo assim, um elemento ciber pode ser composto por outros digital twins. Ele define que um elemento de nível superior funciona como mestre dos seus elementos de nível inferior. Para implementar este tipo de composição, os autores usam uma composição hierárquica de máquinas finitas de estados. Portanto, a modelagem da arquitetura é realizada usando uma máquina de estados baseada em lógica fuzzy. Em relação a interfaces homem computador, o artigo não comenta nada a respeito.

(SCHLUSE *et al.*, 2018)

Um novo conceito é apresentado em (SCHLUSE *et al.*, 2018) chamado “*Experimentable digital twin*” (EDT). Ele é definido como um digital twin que contém modelos de dados (geometria, estrutura, etc), sua funcionalidade (processamento de dados, comportamento, etc) e suas interfaces de comunicação. Ele integra todos os conhecimentos resultantes das atividades de modelagem em engenharia (Modelo Digital) e dos dados de trabalho capturados durante a operação do mundo real (Digital Shadow).

Esse artigo, além de propor esse novo elemento estruturante para processos de engenharia baseados em simulação, o EDT, também propõe uma nova abordagem à tecnologia de simulações que traz vida e esses EDTs. Essa tecnologia é chamada por “*Virtual Testbeds*” e pode ser usada em diversos cenários de aplicação que o autor denomina “*Simulation based X*”, abrangidos por um processo de desenvolvimento integrado chamado “Engenharia de Sistemas Baseado em Simulação” (veja a Figura 19).

A modelagem é feita baseada em MBSE (*Model-based Systems Engineering*) pois

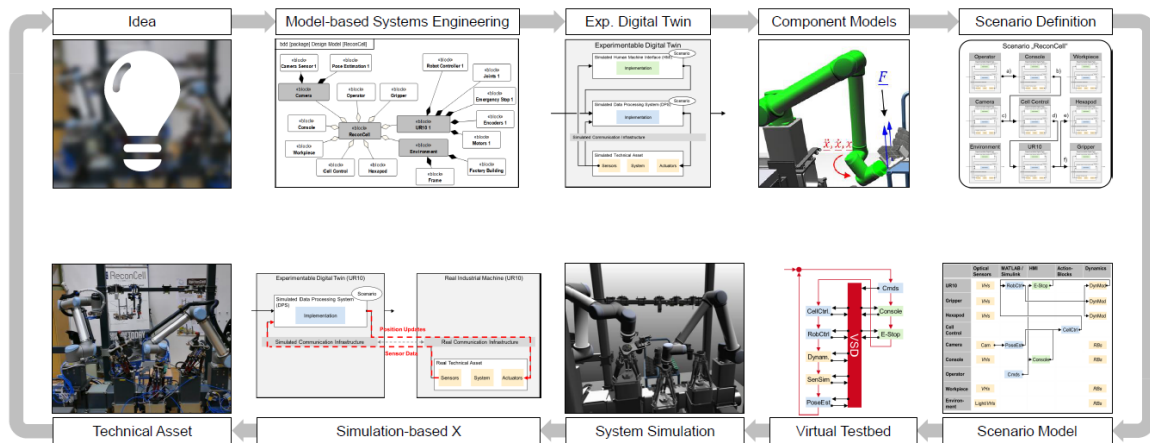


Figura 19: Ciclo de vida que engloba a Engenharia de Sistemas Baseada em Simulação usando EDTs. Fonte: (SCHLUSE et al., 2018)

fornece uma abordagem sistemática para analisar e modelar sistemas complexos. O resultado do processo MBSE, neste caso, são diagramas em SysML, que são o ponto de partida para as atividades de modelagem do ponto de vista da simulação. Para o gerenciamento de dados, usa o VSD (*Versatile Simulation Database*) que é um banco de dados. Ele é capaz de armazenar qualquer estrutura de dados que possa ser modelada por diagramas de classes UML. Em relação a comunicação, o autor cita que existem infra-estruturas de comunicação entre o real e o digital, mas não cita nenhuma em específico.

(ZHUANG; LIU; XIONG, 2018)

Propõe um framework para o digital twin da linha de montagem no chão de fábrica. Esse framework faz o gerenciamento de produção inteligente e o controle para montagem de produtos complexos. A Figura 20 ilustra graficamente o framework proposto.

O artigo aborda sobre os seguintes assuntos para a implementação do framework: (1) aquisição, organização e gerenciamento em tempo real dos dados de montagem do chão-de-fábrica, (2) construção do digital twin da linha de montagem, (3) serviços de predição orientada pela grande quantidade de dados coletados pelo digital twin, (4) serviço de gerenciamento e controle de produção de chão-de-fábrica digital de montagem com base

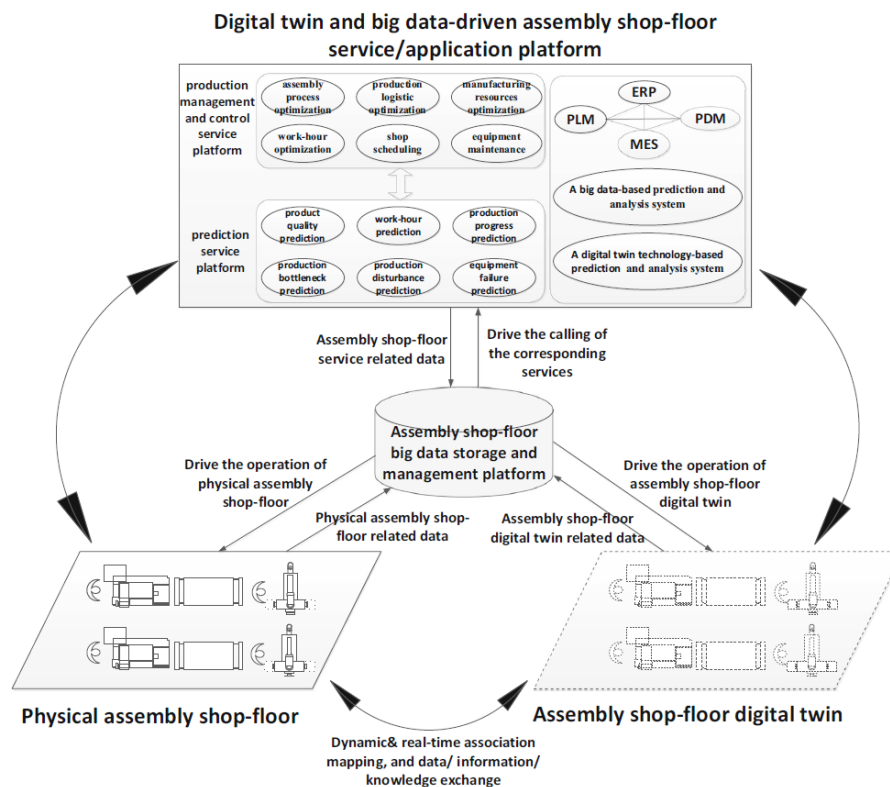


Figura 20: Framework para o digital twin de linhas de montagem no chão de fábrica.
Fonte: (ZHUANG; LIU; XIONG, 2018)

no digital twin.

No entanto, este é um trabalho teórico, onde os autores apresentam a aplicação, os problemas e discutem tecnologias para a implementação do digital twin. Por exemplo, o artigo cita que a identificação pode ser realizada usando RFID tags e que, a comunicação poderá acontecer com o uso da rede industrial, redes wireless, redes móveis, Bluetooth, etc. O artigo apresenta algumas interfaces para o gerenciamento de dados implementadas pelo autor, mas não cita nenhuma interface específica. Sobre a modelagem do digital twin, os autores discursam sobre uma lista de modelos que podem ser usados para criar o modelo do digital twin, dentre esses modelos, estão: modelo geométrico, modelo físico, modelo do chão de fábrica, modelo da linha de produção, modelo das estações de montagem, modelos de recursos de manufatura (montagem, inspeção, medição, teste e logística). Além dos modelos dos produtos e modelo do ambiente.

3.7 Conclusões

Este capítulo apresentou a revisão bibliográfica em relação ao digital twin. Inicialmente, foi comentado o procedimento de pesquisa. Depois, foi abordada a história do digital twin e as definições atribuídas a ele, assim como suas principais aplicações foram apresentadas. Em meio a todas as definições apresentadas, esta Tese refere-se ao digital twin como uma representação virtual de um produto real no contexto de sistemas ciberfísicos. O digital twin possui um modelo de dados que pode armazenar dados durante o ciclo de vida do produto, fornecer serviços de análise dos dados, possuir meios de comunicação com a sua parte física, que atualiza o digital twin com dados e esse se comunica com o físico afim de enviar dados de atuação quando for necessário.

Neste Capítulo, também foi destacado os tópicos de estudo do digital twin encontrados na literatura (identificação, gerenciamento de dados, modelos, HMI e comunicação). Após, os trabalhos relevantes para esta Tese foram analisados afim de encontrar em cada um, uma solução usada para tratar os tópicos apresentados anteriormente. Por fim, apresenta-se a Tabela 4 com um resumo dessa análise e permitindo a comparação dos trabalhos citados. Após uma revisão das publicações, percebe-se que existe uma lacuna a ser preenchida no sentido de que nenhum trabalho considera todos os tópicos de estudo apresentados, sendo este, portanto, um dos objetivos desta Tese.

Tabela 4: Publicações e tópicos de estudo.

Referência	Identificação e rastreamento	Gerenciamento de dados	Modelos	HMI	Aplicação	Topologia	Comunicação
(RIOS et al., 2016)	Uma aeronave já possui um número de série de fabricação (MSN) ou um número de cauda (TN) para identificar cada aeronave	framework Dassault Systemès V6 fornece um banco de dados comum para armazenar e recuperar dados de industrialização	Aircraft digital models: modelos de sistemas, modelo multifísico, modelo geométrico	Não cita	Setor aeroespacial-aeronaves	Fala que é viável ter uma relação de pai e filho para cada unidade do produto	Não cita
(UM; WEYER; QUINT, 2017)	ID	Não cita	As classes do modelo de dados são mapeadas e descritas dentro do AutomationML SystemUnit-ClassLib	O DyVisual consiste em um conjunto de bibliotecas dinâmicas, como controle de teclado, animação geométrica e atualização de dados, que são detectados e importados automaticamente.	Criação de um ambiente para simulação de módulos plug-and-play de linhas de fábrica	Não cita	A comunicação entre os módulos é feita usando endereço IP e porta
(CAI et al., 2017)	Não cita	Tanto os dados de fabricação como os dados do sensor são armazenados em um banco de dados PostgreSQL em um PC local conectado à Internet	UML	Os usuários podem acessar os dados no banco de dados através da internet com o nome de usuário e a senha válidos.	Criar o DT de máquinas usando técnicas de integração de dados de sensores	Não cita	Um cabo de adaptador serial RS232 é conectado entre o gateway DB-25 do controlador da máquina CNC e a porta USB do computador para coletar dados de fabricação
(SIERLA et al., 2018)	Usa a ID que o próprio AutomationML cria para cada componente	Não cita	UML + AutomationML	Usa Java 3D simulation environment JMonkeyEngine3 para implementar o framework de simulação	Simulação de montagem (assembly) de produtos com foco em o produto ter o controle para orquestrar sua montagem	DT desconectada porque o DT é de um produto que não existe ainda	Não existe comunicação entre o físico e real porque o físico ainda não existe

Tabela 4 continuação da página anterior

Referência	Identificação e rastreamento	Gerenciamento de dados	Modelos	HMI	Aplicação	Topologia	Comunicação
(ALAM; SADDIK, 2017)	Toda coisa física ou cibernética é identificada por uma identificação única (ID) (i.e. IPv6, Universal Product Code (UPC), Electronic Product Code (EPC), etc.)	Cada coisa física e seu correspondente cibernético gerenciam uma banco de dados	Máquina de estados baseado em lógica fuzzy	não cita	Propõe um modelo de arquitetura de referência para dispor o DT na nuvem. Aplicação serve para qualquer “coisa” física (“Things”)	Um digital twin se comunica com outros seguindo uma topologia ou relacionamento	M2M social network
(SCHLUSE et al., 2018)	Não cita	Versatile Simulation Database - O VSD é um banco de dados que é capaz de armazenar qualquer estrutura de dados que possa ser modelada por diagramas de classes UML.	Diagramas em SysML, que são o ponto de partida para as atividades de modelagem do ponto de vista da simulação.	Simulado HMI	Produção industrial, área de construção e gerenciamento de instalações, setor automotivo	Não cita	Pode existir qualquer tipo de infraestrutura de comunicação.
(ZHUANG; LIU; XI-ONG, 2018)	RFID tags e códigos de barra	Comenta sobre o uso de Big Data	Modelo geométrico, modelo físico (modelo do chão de fábrica, modelo da linha de produção, modelo das estações de montagem, modelos de recursos de manufatura), modelos do produto e modelo do ambiente	usa interfaces para coletar e organizar os dados	Satellite assembly shop-floor: uso do DT para implementar uma linha de montagem inteligente	Não cita	Industrial LAN, wireless and mobile networks, Bluetooth, e RFID sensor networks

4 ARQUITETURA E MODELO DE REFERÊNCIA PARA O DIGITAL TWIN

Neste capítulo será proposta uma nova arquitetura e um novo modelo para o digital twin. Na definição da arquitetura foram especificados os principais componentes e atores presentes em aplicações que utilizarão o digital twin, assim como a forma como eles podem se relacionar e trocar informações entre si. Baseado nesta especificação, é proposto um modelo genérico do digital twin que incorpora as características da arquitetura. Tanto a arquitetura quanto o modelo foram propostos para serem genéricos e flexíveis, podendo ser aplicados em diversos cenários de forma que nenhuma plataforma de implementação ou aplicação específica foi definida neste momento. A análise dos possíveis cenários de aplicação levaram à proposta de possíveis topologias de conectividade que podem ser usadas para categorização do digital twin.

Inicialmente, na Seção 4.1, são apresentadas as principais estratégias de projeto que fundamentam algumas das decisões tomadas ao longo do processo de modelagem. Em seguida, na Seção 4.2 é apresentada a arquitetura de referência para o digital twin. Na Seção 4.3 é proposto um modelo de digital twin que se adéqua a tal arquitetura. Na Seção 4.4 é apresentada uma análise das topologias de conectividade do digital twin. Por fim, as conclusões do capítulo são apresentadas na Seção 4.5.

4.1 Estratégias de projeto adotadas

Uma estratégia de projeto define em alto nível qual será a abordagem no desenvolvimento do modelo ou arquitetura, definindo princípios abrangentes que norteiam as tomadas de decisão ao longo do processo. As informações levantadas nos capítulos anteriores serviram como base para a definição dos principais paradigmas de projeto aqui adotados, seja devido ao sucesso obtido ou devido a uma lacuna observada que não foi atendida.

Neste trabalho, foi adotada a estratégia de engenharia baseada em modelo porque dada a relevância do tópico de estudo desta Tese e a evidente capacidade de agregação de valor que um modelo rico em detalhes pode trazer ao digital twin, decidiu-se adotar uma metodologia de projeto que também coloque o modelo, tanto do dispositivo físico quanto do próprio digital twin, em foco.

Um dos objetivos desta Tese é que as propostas aqui presentes sejam tão genéricas e flexíveis quanto possível, sendo que neste capítulo, nenhuma plataforma de implementação ou aplicação específica é definida. Visando este objetivo, adota-se um princípio de projeto modular “de baixo para cima”, mais conhecida como abordagem *bottom-up*, onde os elementos individuais de baixo nível do sistema são identificados primeiro. Estes elementos são então utilizados para formar subsistemas maiores através de um processo de composição e agregação, até que o sistema completo de nível superior seja formado.

A distribuição dos componentes de um sistema através de serviços conectados na web tem sido uma tendência recente em várias áreas da computação e automação. Isto se deve principalmente devido a flexibilidade na criação de novas aplicações que utilizem os serviços e a total abstração quanto as especificações do dispositivo computacional (hardware, sistema operacional e a localização geográfica) que hospeda o serviço, possibilitando que ele implemente o conceito de “serviço na nuvem”. Visto que estes quesitos são impor-

tantes dentro do contexto de Indústria 4.0 e que arquiteturas baseadas em serviços têm demonstrado bons resultados nesta área, deu-se prioridade a arquiteturas com esta abordagem.

4.2 Arquitetura de referência

Baseando-se nas estratégias de projeto e nos conceitos levantados ao longo dos capítulos anteriores, chegou-se na arquitetura apresentada a seguir. Em (RIOS et al., 2015), as principais características que um digital twin deve apresentar foram descritas e serão adotadas neste trabalho como requisitos de projeto, devendo portanto ser suportadas pela arquitetura proposta. Esses requisitos já foram apresentados e analisados na Seção 3.5.

A Figura 21 apresenta uma visão geral da arquitetura de referência para o digital twin aqui proposta. Os componentes em azul são aqueles que fazem parte do digital twin de fato, enquanto os componentes verde são externos, não sendo considerados parte do digital twin mas mantendo uma interação direta com este. Os componentes pontilhados são opcionais, podendo estar presentes ou não de acordo com a aplicação, enquanto os componentes com borda contínua são considerados obrigatórios e sempre deverão estar presentes em qualquer sistema que se baseie nesta arquitetura de referência.

A seguir é feita uma descrição dos componentes obrigatórios que devem estar presentes na arquitetura:

- Modelo do dispositivo físico: modelo abrangente podendo conter diversas informações relevantes do dispositivo físico, tais como especificações do seu projeto construtivo, propriedades técnicas, identificação única, descrição dos seus componentes, estado atual de suas propriedades dinâmicas, especificação de operação, etc. A complexidade deste modelo e quais as informações estarão contidas nele são de-

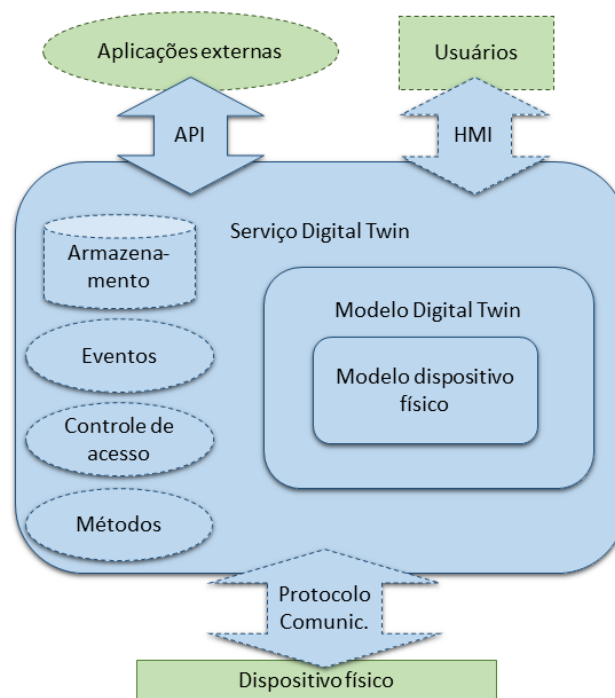


Figura 21: Visão geral da arquitetura de referência.

pendentes da aplicação, podendo variar de uma simples estrutura de dados contendo algumas especificações técnicas até um modelo com vários arquivos contendo representação em 3D, modelo dinâmico, esquemáticos de projeto, etc.

- Modelo do digital twin: modelo abrangente do digital twin contendo todas as informações necessárias para sua implantação e execução. Na arquitetura proposta o modelo do digital twin contém o modelo do dispositivo físico, podendo assim utilizar tais informações de forma agregada. Ou seja, o modelo do digital twin pode estender ou se basear em informações do modelo físico para se configurar de forma apropriada e fornecer as suas funcionalidades. No entanto, deve-se observar que o contrário não se aplica: o modelo do dispositivo físico é independente do modelo do digital twin, permitindo assim que o dispositivo físico possa ser modelado e projetado sem consideração quanto a sua eventual utilização em um digital twin. Na Seção 4.3 será proposto um modelo genérico que atende as necessidades aqui

apresentadas.

- Serviço digital twin: é o serviço que permite o compartilhamento dos dados e acesso às funcionalidades providas pelo digital twin. É através deste serviço que usuários e aplicações externas interagem com o digital twin, sendo que o primeiro o faz através de Interfaces Homem-Máquina (HMI) e o segundo através de Interfaces de Programação de Aplicação (API). Este serviço também implementa a lógica de execução do digital twin, que integra todos os componentes internos (modelos, banco de dados, controle de segurança, protocolos de comunicação, etc) a fim de implementar as funcionalidades do digital twin. De forma geral, assume-se que este serviço é um processo sendo executado em um dispositivo computacional com acesso a internet.

Além destes componentes, a arquitetura de referência prevê a possível existência de outros componentes representados como opcionais:

- Armazenamento de dados: componente utilizado para manter o histórico de operação do dispositivo físico e do digital twin. Idealmente, todas alterações devem ser registradas, no entanto podem existir limitações quanto a capacidade de armazenamento e processamento que dependem da aplicação.
- Gerenciador de eventos: componente responsável por gerenciar o disparo de eventos e possível notificação da ocorrência de tais eventos para outros componentes, aplicações ou usuários. Esta característica permite que o digital twin se torne um componente ativo, gerando informação de forma assíncrona para outros componentes do sistema e não somente consumindo informação ou respondendo a solicitações de forma passiva.

- **Controle de acessos:** define regras específicas de controle de acesso aos recursos do digital twin, incluindo informações e funcionalidades. Este controle deve permitir especificar restrições de acesso a determinados atributos, HMIs, APIs e métodos a determinados usuários ou grupos de usuários, sendo necessário portanto alguma forma de autenticação de usuários.
- **Métodos de acesso a funcionalidades:** um digital twin pode ter funcionalidades que podem ser usadas por outras aplicações ou usuários. Essas funcionalidades são modeladas como métodos, que são um conjunto de instruções executados pelo digital twin. Métodos podem ser usados para controlar o dispositivo, executar simulações e programas de diagnósticos, entre outras programas executáveis.
- **Interfaces de comunicação:** interfaces que fazem a comunicação com o dispositivo físico, sendo assim responsáveis pela atualização das propriedades do digital twin quando o dispositivo físico sofrer alguma alteração no seu estado e vice-versa. De forma geral, pode-se dizer que estas interfaces fazem a ponte entre o físico e o cyber.

4.3 Modelo do digital twin

A Figura 22 mostra o modelo proposto para o conceito de digital twin usando a notação do diagrama de classe UML. O modelo proposto reflete em grande parte os componentes presentes na arquitetura de referência apresentada na Seção anterior, de forma que serve como uma descrição completa para um digital twin dentro desta arquitetura, contendo todas as informações necessárias para a sua implantação e execução.

A principal característica do digital twin é que ele deve necessariamente, ter uma associação a pelo menos um modelo de dispositivo físico (modelado através da multiplicidade

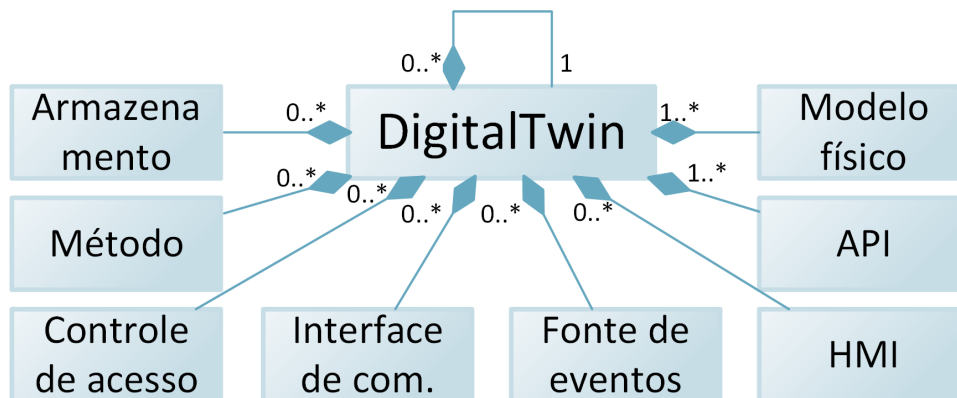


Figura 22: Diagrama UML do modelo de referência para o digital twin.

1..*)), que é uma representação virtual interna para o dispositivo físico. Embora muitos trabalhos anteriores na área assumam que a representação virtual inclui um modelo tridimensional do dispositivo, isso não é um requisito obrigatório quando se considera uma abordagem ampla do conceito de digital twin. Uma vez que se busca flexibilidade e generalidade neste momento, nenhum tipo de restrição ou requerimento quanto ao tipo de informação presente neste modelo é aplicada.

Durante a execução, o modelo do dispositivo físico deve ser mantido em sincronia com o estado real e atual do dispositivo físico, de modo que qualquer alteração no domínio físico deve ser devidamente refletida nas propriedades de seu modelo. O contrário também pode ser permitido, ou seja, uma alteração no modelo deve ser refletida sobre o dispositivo físico através de uma ação de controle. Mais detalhes sobre a implantação e execução do digital twin baseadas neste modelo serão apresentados ao longo dos próximos capítulos.

O modelo proposto também especifica que um digital twin deve necessariamente ter pelo menos uma interface de programação de aplicativo (API). Entende-se que aplicações e sistemas externos devem acessar as funcionalidades do digital twin através de um protocolo bem definido, sendo a API um conjunto de especificações que permite tal acesso externo. Por exemplo, a API pode permitir que aplicativos externos consultem o estado

atual do modelo virtual, acessem o registro histórico de um sensor, ou acionem o início de uma simulação.

No modelo proposto, o modelo físico e a API são os únicos dois componentes que são necessários de fato, o que significa que o digital twin mínimo deve ter um modelo do dispositivo físico e uma API. Opcionalmente, o modelo do digital twin pode ter outros componentes que agregam funcionalidades, tais como armazenamento de dados, geração de eventos, execução de métodos, controle de acesso a seus recursos e fornecimento de interfaces homem-máquina (HMIs) para os usuários. É argumentado que esses componentes suportam a implementação da maioria das funcionalidades de um digital twin. Esses componentes estão diretamente relacionados com os componentes da arquitetura de referência apresentada na Seção anterior.

4.4 Análise das topologias de conectividade do digital twin

Nos sistemas de manufatura e automação, os componentes (controladores, máquinas, supervisórios, etc) e usuários do sistema podem estar conectados de diversas formas. Ao se fazer uma análise em alto nível destas formas, pode-se observar o surgimento de certos padrões. De forma análoga, um sistema com digital twin pode apresentar diferentes formas de organização. Nesta Seção é apresentada uma análise que além de propor uma nomenclatura para as topologias observadas atualmente na literatura, propõem novas topologias suportadas pela arquitetura e modelo de referência.

Inicialmente, para a realização da análise identificou-se os principais componentes do sistema como sendo:

- Dispositivo, sendo este atrelado ao digital twin, podendo ser uma máquina, produto, ferramenta, robô, veículo, etc.

- Digital twin, onde assume-se que ele segue a arquitetura de referência apresentada na Seção 4.2.
- Aplicações externas, que utilizam ou são utilizadas pelo digital twin, podendo ser equipamentos externos, softwares ou até mesmo outros digital twins.
- Usuários, que necessitam utilizar as funcionalidades do digital twin.

Dados esses componentes, foram identificadas algumas possíveis topologias de conexão. A seguir é apresentada a descrição das topologias propostas.

4.4.1 Digital twin desconectado

O digital twin desconectado não possui uma conexão direta com o dispositivo, seja porque este não possui capacidade de comunicação ou porque não é relevante ou exequível implementar tal conexão. Um exemplo típico do primeiro caso é quando o dispositivo é um produto, tal como um pneu, ou uma ferramenta sem nenhuma capacidade de processamento computacional e por conseguinte de comunicação, tal como uma furadeira. Não obstante, ainda se faz necessário que o digital twin obtenha informações sobre o dispositivo, sendo que neste caso, tais informações devem ser providas de forma indireta por outros atores ao longo do ciclo de vida do dispositivo, seja através da API ou da HMI. Ou seja, as informações sobre o dispositivo são provenientes de componentes externos ou usuários, e não do dispositivo físico.

A Figura 23 ilustra esta topologia. A seta pontilhada entre o dispositivo e as aplicações externas e usuários representa uma aquisição indireta, que não se dá por vias de protocolo de comunicação.

Um exemplo desta topologia pode ser quando o digital twin for utilizado como uma representação virtual de produtos ao longo do seu ciclo de vida. Quando o produto passa

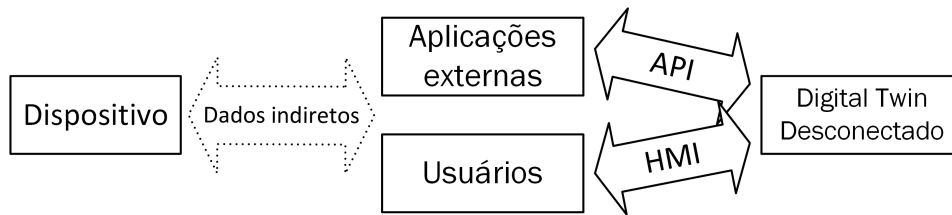


Figura 23: Digital twin desconectado.

por uma máquina, ele é identificado, por exemplo através de um código QR (*Quick Response Code*), e o seu estado no digital twin é atualizado com as informações passadas pela máquina. Nesse caso não é o dispositivo que está provendo a informação, mas a máquina que, através de uma interação com o dispositivo, alimenta os dados para o digital twin.

Esta topologia também é observada em digital twins com foco no início do ciclo de vida, onde o dispositivo pode não estar *online* ou nem mesmo existir. Por exemplo, em (SIERLA et al., 2018) é proposto um digital twin para simulação do processo de manufatura de um dispositivo, sendo que nenhuma informação tem origem no próprio dispositivo.

Tendo como base o modelo de referência da Figura 22, pode-se dizer que o modelo do digital twin desconectado é uma derivação do modelo de referência onde existe uma restrição de que a multiplicidade da associação para os componentes de interface de comunicação, que originalmente é $0..*$, deve passar a ser necessariamente zero.

4.4.2 Digital twin embarcado

O digital twin embarcado está implementado no próprio dispositivo, compartilhando alguns dos recursos computacionais. Embarcar o digital twin no dispositivo pode facilitar a sincronização de informações entre mundo físico (estado real do dispositivo) e o mundo *cyber* (representação do dispositivo no digital twin), visto que ambas informações se encontram no mesmo dispositivo computacional. No entanto, a implantação de algumas funcionalidades do digital twin podem se tornar inviáveis devido às limitações computa-

cionais geralmente presentes em dispositivos embarcados, tais como baixa capacidade de armazenamento e/ou pouco poder de processamento.

A Figura 24 ilustra esta topologia, onde pode-se observar que o dispositivo e o digital twin estão dentro da mesma caixa, representando o mesmo dispositivo computacional, e a comunicação é interna.

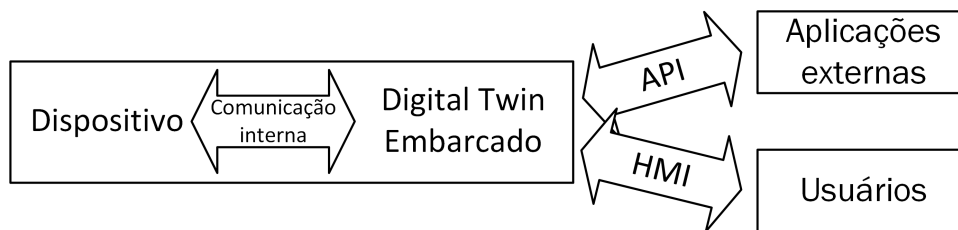


Figura 24: Digital twin embarcado.

Diversos equipamentos atualmente implementam algumas funcionalidades de alto nível (tal como manutenção preditiva, HMI, etc) no próprio dispositivo. Um exemplo é a válvula utilizada no estudo de caso deste trabalho, que possui um *watchdog* interno para antecipação de possíveis falhas. No entanto, geralmente estas funcionalidades não estão caracterizadas de forma que possam ser chamadas de digital twin, sendo que não foi encontrado nenhum trabalho que explicitamente implemente o conceito de digital twin embarcado.

Assim como na topologia anterior, o modelo do digital twin embarcado também não terá interfaces de comunicação uma vez que elas não se fazem necessárias, portanto a multiplicidade da associação para os componentes de interface de comunicação deve passar a ser necessariamente zero.

4.4.3 Digital twin conectado

Neste caso, o digital twin conectado está implementado em um recurso computacional diferente do dispositivo, mas assume-se que existe ao menos um canal de comunicação

entre o dois. E, toda a sincronização entre a representação física e *cyber* é enviada através deste canal. A Figura 25 ilustra esta topologia.



Figura 25: Digital twin conectado.

No âmbito da automação industrial, muitos equipamentos implementam algum tipo de comunicação usando protocolos industriais, possibilitando assim, ao menos em princípio, a implantação do digital twin no modo conectado.

A restrição imposta por esta topologia no modelo de referência é que a multiplicidade na associação de interfaces de comunicação deve ser 1..*.

4.4.4 Digital twin multi-dispositivo

Nesta topologia o digital twin está conectado a mais de um dispositivo. Tradicionalmente, o digital twin se refere a um único dispositivo, sendo que mesmo quando o dispositivo em questão é um sistema com vários componentes, existe uma única representação *cyber* do sistema. Em contrapartida, o digital twin multi-dispositivo pode conter representações independentes para múltiplos dispositivos.

No modelo de referência apresentado na Seção 4.3, esta possibilidade está presente na multiplicidade 1..* da associação entre digital twin e modelo do dispositivo. Este tipo de topologia tem a vantagem de possibilitar o compartilhamento de componentes do digital twin para múltiplos dispositivos, tal como banco de dados, serviço web, etc. Ou seja, apesar de se ter múltiplas representações *cyber*, elas estão hospedadas no mesmo recurso computacional e portanto podem ser utilizadas de forma compartilhada.

A Figura 26 ilustra esta topologia, onde n é um número arbitrário de dispositivos, cada um com a sua conexão com o digital twin e com o seu modelo de dispositivo.

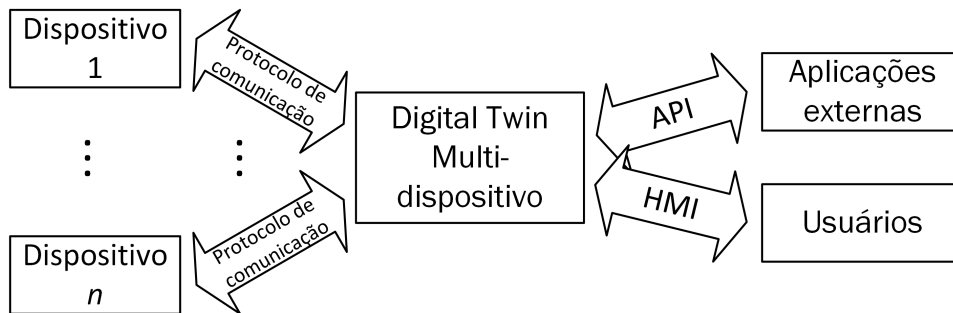


Figura 26: Digital twin multi-dispositivo.

A restrição imposta por esta topologia ao modelo de referência é que a associação entre digital twin e modelo físico deverá ser um para vários, assim como a associação entre digital twin e interfaces de comunicação.

4.4.5 Digital twin agregado

Nesta topologia o digital twin é composto por outros digital twins. Da mesma forma que um dispositivo físico pode ser uma composição de partes, é proposto que um digital twin possa ser criado e estendido através da agregação de outros digital twins. Pode ser visto como um digital twin hierárquico, onde o pai agrega as funcionalidades dos filhos, possivelmente estendendo-as. Esta abordagem permite por exemplo, que digital twins embarcados com poucas funcionalidades devido a limitações computacionais possam ser agregadas e expandidas em um digital twin com mais recursos computacionais. Além disto, existe a possibilidade de se manter uma relação um-para-um entre físico e o *cyber* em diferentes níveis do sistema, onde o digital twin agregado reflete os sub-componentes da sua parte física. Por exemplo, o digital twin de uma linha de produção em um sistema de manufatura pode ser uma agregação dos digital twins de suas máquinas e produtos; a agregação de linhas de produção pode gerar o digital twin da fábrica como um todo.

Em um digital twin agregado haverá necessariamente comunicação entre o digital twin pai e seus filhos. Esta comunicação pode ser feita utilizando-se diversos protocolos, no entanto, é sugerido que ela seja feita exclusivamente através da API, pois possibilita a padronização na troca de dados e possibilita a abstração dos detalhes de implementação de cada componente. Mais detalhes operacionais sobre esta topologia serão explorados na Seção 6.2, onde será proposto uma metodologia para sua implementação.

A Figura 27 ilustra esta topologia, onde n é um número arbitrário de digital twins sendo agregados. A topologia de cada um destes digital twins é arbitrária, portanto a forma como eles se conectam aos seus respectivos dispositivos não é relevante e não é mostrada na Figura.

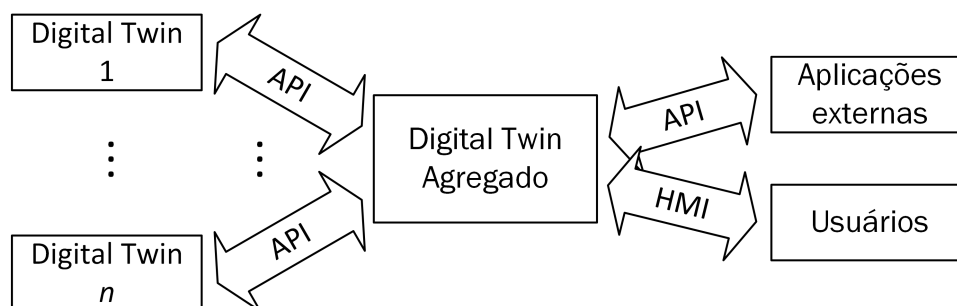


Figura 27: Digital twin agregado.

No modelo de referência, a possibilidade de agregação está modelada na associação reflexiva presente no próprio digital twin, sendo que para caracterização desta topologia é necessário que a multiplicidade seja de um para vários.

4.5 Conclusão

Neste Capítulo, foi proposta uma arquitetura de referência para o digital twin, seguida de um modelo genérico e flexível que atende as necessidades da arquitetura. Destaca-se a importância de se ter um modelo, tanto do dispositivo quanto do próprio digital twin, que

passam a ter papel central na arquitetura.

Foram levantadas as diferentes topologias que os componentes de um sistema com digital twin podem apresentar, o que originou uma proposta para categorização do digital twin baseado nesta topologia. Cinco categorias foram descritas: desconectado, embarcado, conectado, multi-dispositivo e agregado. Deve-se observar no entanto, que podem existir casos de topologias mistas, que exibem as características de duas ou mais topologias (por exemplo, um digital twin é agregado e conectado ao mesmo tempo). O estudo destas possibilidades e de suas características foge do foco desta Tese e é proposto como trabalho futuro.

5 MODELAGEM COM AUTOMATIONML

A arquitetura de referência proposta no capítulo anterior foi apresentada de forma genérica, sem imposição de nenhuma tecnologia subjacente. Na prática, ela pode ser instanciada de diferentes maneiras utilizando-se diversas tecnologias para implantação de seus componentes. Por exemplo, ela poderia ser implementada usando protocolos de comunicação industrial tais como OPC-UA e MTConnect que já oferecem suporte à alguns conceitos presentes na proposta. No OPC-UA por exemplo, o modelo de informação que representa pontos de dados e a relação entre eles pode ser visto como equivalente ao modelo do dispositivo físico. Já o MTConnect especifica um modelo de dispositivo em XML que pode ter adaptadores de comunicação semelhantes às interfaces de comunicação presente na arquitetura proposta. Ambas tecnologias especificam uma API para acesso a dados através de uma arquitetura baseada em serviços. No entanto, outros conceitos como HMI e controle de acesso não estão presentes, sendo que teriam que ser estendidos para suportar totalmente a arquitetura, o que pode ser problemático visto que se tratam de protocolos com especificações rígidas e altamente regulamentadas. Desta forma buscou-se por tecnologias alternativas que sejam capazes de proporcionar mais flexibilidade e que suportem a implantação de todos os aspectos previstos na arquitetura.

Outras tecnologias que podem ser citadas como possíveis bases para implantação são

frameworks para internet das coisas (por exemplo, Fiware¹) e sistemas SCADA. Uma análise aprofundada e a comparação entre estas tecnologias será foco de trabalhos futuros.

Nesta tese, optou-se pelo uso de duas tecnologias principais em conjunto: AutomationML para a modelagem e Webservice para a disponibilização dos serviços via internet. A escolha destas tecnologias se deu principalmente devido a grande flexibilidade que elas proporcionam, tanto na modelagem quanto na implantação do sistema. Mais detalhes sobre a motivação da escolha destas tecnologias será apresentada ao longo deste capítulo.

Este capítulo descreve uma metodologia para a modelagem e implantação do digital twin utilizando estas tecnologias. O diagrama de fluxo mostrado na Figura 28 resume a metodologia proposta. Na Seção 5.1 é descrito o primeiro passo do processo, que é a modelagem do dispositivo físico. A Seção 5.2 aborda o segundo e terceiro passos, cujo foco passa a ser o modelo do digital twin. A Seção 5.3 traz uma análise dos padrões de modelagem observados de acordo com a topologia do digital twin. O último passo da Figura 28 é abordado com mais detalhes no próximo capítulo.

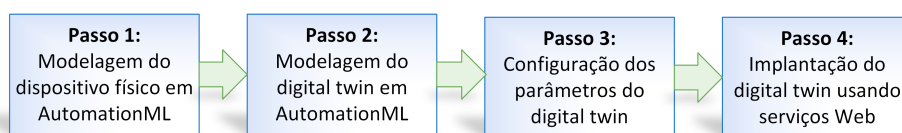


Figura 28: Visão geral da modelagem e implantação do digital twin.

5.1 Modelagem do Dispositivo Físico

O primeiro passo na criação de um digital twin é a modelagem do dispositivo físico. Neste modelo, devem estar especificados todos os atributos relevantes que serão usados para formar uma representação virtual do dispositivo.

Assim como em qualquer processo de modelagem, o modelo pode ter diferentes níveis

¹<https://www.fiware.org/>

de detalhamento, podendo variar entre ser muito simples até muito complexo ou mesmo ser um modelo com múltiplas granularidades de detalhamento. O nível de detalhamento a ser adotado depende das características da aplicação em que o digital twin será empregado e portanto, não é definido nesta Seção uma vez que se busca uma metodologia genérica, que possa ser aplicada em diversos casos. No entanto, justamente a fim de que se garanta esta generalidade e flexibilidade, deve-se definir uma linguagem para a descrição do modelo que tenha estas características.

Existem várias linguagens de modelagem propostas para a área de automação e sistemas industriais. PackML (HAGGE; WAGNER, 2008) (*Packaging Machine Language*) por exemplo é um padrão de modelagem para máquinas de manufatura discreta definida pela OMAC (*Organization for Machine Automation and Control*) cujo principal objetivo é proporcionar uma maior integração entre equipamentos e softwares de manufatura ao longo de uma linha de produção. A linguagem define terminologias, estruturas de dados e um modelo de estados para descrição das características operacionais e sequência de controle das máquinas. Apesar dos seus princípios se alinharem com os objetivos aqui almejados, observa-se que esta linguagem possui um escopo de aplicação limitado, não se estendendo por exemplo para aplicações na indústria de processos.

SysML (*Systems Modeling Language*) (FRIEDENTHAL; MOORE; STEINER, 2014) é uma linguagem de modelagem genérica para a engenharia de sistemas ou sistemas-de-sistemas. Seu objetivo principal é fornecer padrões de modelagem para auxiliar no processo de desenvolvimento e execução de um projeto, contando com suporte para a fase de análise, design, verificação e validação de sistemas. SysML é uma extensão da linguagem de modelagem UML (*Unified Modeling Language*).

Neste trabalho, foi escolhida a linguagem AutomationML, que já foi apresentada na

Seção 2.7. Destaca-se as seguintes principais vantagens do AutomationML que motivaram a sua escolha:

- Flexibilidade: pode ser aplicada em uma ampla gama de casos e com diferentes níveis de detalhamento do modelo;
- Portabilidade: as informações do modelo são armazenadas no padrão XML, que é facilmente compartilhado e interpretado entre dispositivos heterogêneos;
- Recursos orientados a objetos: um modelo pode ser derivado e instanciado, o que proporciona a capacidade de reaproveitamento e especialização de modelos;
- Orientação à automação: como o próprio nome sugere, AutomationML é desenvolvida para aplicações na área de automação industrial em geral, alinhando-se portanto, com a proposta desta Tese.

Nesta tese, é proposto que o modelo do dispositivo em AutomationML seja diretamente usado como a representação *cyber* do dispositivo. Quando um digital twin está sendo executado, a hierarquia da instância do modelo AutomationML é carregada na memória do dispositivo computacional que hospeda o digital twin e usada como um modelo dinâmico de execução, sendo mantido em sincronia com o estado do dispositivo físico. Por exemplo, quando um sensor é ativado no domínio físico, os valores são comunicados ao digital twin através de uma interface de comunicação, que então altera o atributo apropriado do sensor na hierarquia da instância. Alterações estruturais na hierarquia da instância também podem ser feitas em tempo de execução, permitindo por exemplo, que o modelo de uma máquina seja expandido de forma dinâmica ao se adicionar uma ferramenta a ela.

O conhecimento sobre o tipo de classe e o papel semântico de cada elemento interno

disponível em modelos AutomationML também é muito útil em algumas aplicações. Este tipo de informação (que pode ser vista como meta-informação) permite que sejam conhecidas não só as propriedades do dispositivo, mas também qual é o seu tipo (classe) e papel (*role*). Estas informações são especialmente importantes no âmbito da Indústria 4.0, onde sistemas heterogêneos precisam se adaptar e responder a novas demandas em cenários não vistos de acordo com as suas características. Neste contexto, uma aplicação inteligente, por exemplo, pode ter conhecimento através do modelo sobre o tipo de cada componente, e, se necessário, adaptar o seu comportamento de acordo com estas informações.

5.2 Modelagem do digital twin

Nesta etapa, que compreendem os passos 2 e 3 da Figura 28, é descrita uma metodologia para a modelagem do digital twin. O objetivo é criar um modelo que contenha todas as informações sobre seus componentes, permitindo que estes sejam criados e executados baseando-se unicamente neste modelo, sem a necessidade de configuração externa. Para isto, é necessário que o modelo do digital twin inclua não somente a descrição dos componentes, mas também o relacionamento entre eles quando há troca de informação.

Como mencionado anteriormente, o AutomationML também foi usado para modelar o digital twin. Assume-se aqui que o modelo do dispositivo físico já está modelado em AutomationML, podendo assim ser utilizado na modelagem do digital twin se necessário.

É proposta a execução de cinco etapas principais que vão da modelagem genérica através de *roles* e classes (SUCs) até a instanciação concreta e configuração dos componentes. As cinco etapas são descritas a seguir, mas deve-se já ressaltar que, uma vez que uma biblioteca de componentes foi criada (passos 1 a 3), esta pode ser reutilizada em outros modelos, então somente os passos 4 e 5 são necessários para modelar um novo digital

twin.

5.2.1 Definição dos papéis (*roles*)

As classes de papéis (ou *Role Classes* (RC), como referido no AutomationML) servem para definir a função semântica dos elementos de um modelo. Visto que o modelo de referência descrito na Seção 4.3 está sendo usado como base para a modelagem em AutomationML, as *role classes* aqui propostas para o digital twin replicam o modelo de referência e são: comunicação, armazenamento, métodos, eventos, HMI e controle de acesso. Estas *roles classes* foram criadas dentro de uma biblioteca conforme ilustradas na Figura 29, podendo assim ser importadas e reutilizadas em outros projetos do AutomationML.

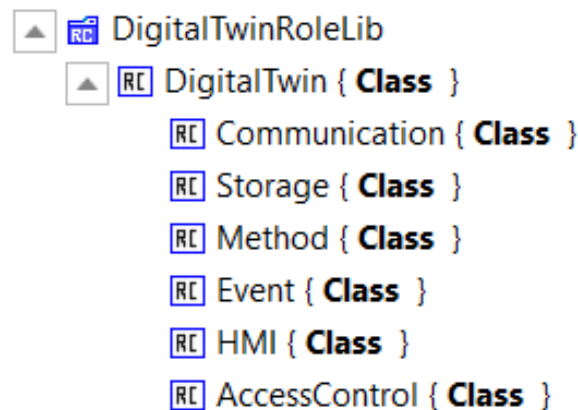


Figura 29: Biblioteca de papéis do digital twin.

Uma parte do código XML da biblioteca contendo os papéis é mostrada abaixo:

```

<RoleClassLib Name="DigitalTwinRoleLib">
  <Version>1.0.0</Version>
  <RoleClass Name="DigitalTwin">
    <RoleClass Name="Communication">
      <Description>Communications provide a link between the
        ↔ digital twin execution and the device.
      </Description>
    </RoleClass>
    <RoleClass Name="Storage">
      ...
    </RoleClass>
  
```

```
</RoleClassLib>
```

5.2.2 Definição das Interfaces

As interfaces no AutomationML servem para modelar explicitamente a interação entre elementos. Foi criada apenas uma interface: `AttributeInterface`, que expõe um atributo definido no modelo do dispositivo a outros componentes, permitindo assim que este atributo possa ser alterado ou lido durante a execução do digital twin.

Em outras palavras, esta interface modela o fluxo de informação entre um atributo do modelo físico para algum componente do digital twin, permitindo que se modele a dinâmica entre eles durante a execução. Por exemplo, criando um link entre uma interface de comunicação do digital twin para um `AttributeInterface`, é modelado que o atributo será alterado pelo canal de comunicação.

O código em XML da definição desta interface é exibido abaixo, onde pode-se perceber que a interface `AttributeInterface` é derivada da classe `AutomationMLBaseInterface`, que é a classe base das interfaces do AutomationML:

```
<InterfaceClassLib Name="DigitalTwinInterfacesLib">
  <Version>1.0.0</Version>
  <InterfaceClass Name="AttributeInterface" RefBaseClassPath="
    ↪ AutomationMLInterfaceClassLib/AutomationMLBaseInterface">
    <Attribute Name="attributeName" AttributeDataType="xs:string
      ↪ ">
      <Description>Name of an attribute in the InternalElement.
        ↪ This interface exposes the attribute externally so
        ↪ it can be changed. </Description>
      <Value> </Value>
    </Attribute>
  </InterfaceClass>
</InterfaceClassLib>
```

5.2.3 Criação das classes de unidades do sistema (*System Unit Classes* - SUC)

As *System Unit Classes* (SUCs) modelam componentes específicos que podem ser reutilizados através de instanciação. Portanto, esta etapa da modelagem tem como objetivo criar elementos genéricos que podem ser instanciados posteriormente na construção do digital twin. A Figura 30 mostra uma visão geral dos componentes criados (os detalhes foram mantidos escondidos para melhor visualização).

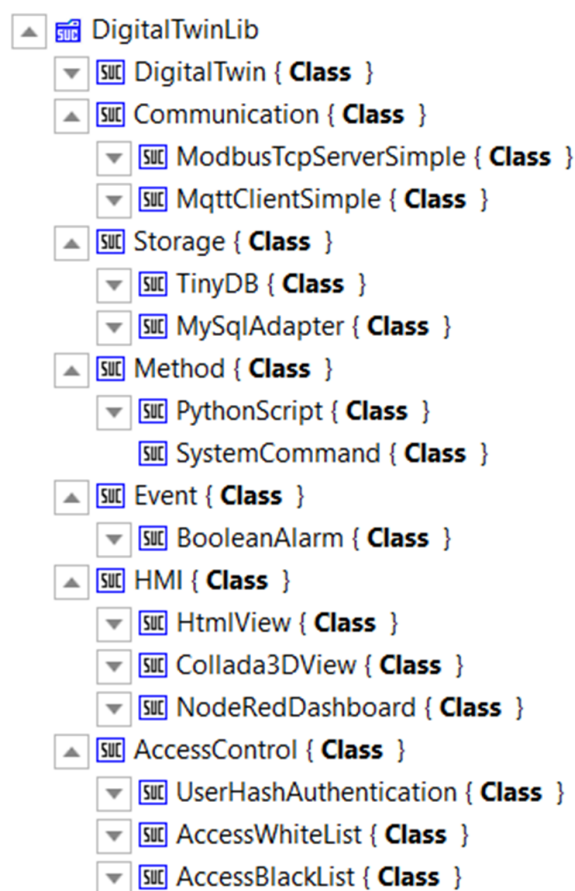


Figura 30: digital twin System Unit Classes.

Cada SUC suporta um único papel definido na etapa 1, que especifica sua função semântica no digital twin. Cada componente modelado como uma SUC deve conter todas as informações necessárias para implantar este componente na prática. Abaixo por exemplo, é mostrado o código em XML da SUC principal `DigitalTwin`, que contém dois atributos importantes que precisam ser especificados: o endereço IP e a porta TCP do seu

servidor web. A escolha deste protocolo deve-se ao fato que o digital twin se comunica externamente via web services que utiliza o protocolo HTTP/TCP.

```
<SystemUnitClassLib Name="DigitalTwinLib">
  <Description>Library of components to model a DigitalTwin.</
    ↳ Description>
  <Version>0.2</Version>
  <Copyright>Greyce N. Schroeder</Copyright>
  <SystemUnitClass Name="DigitalTwin">
    <Description>A DigitalTwin component.</Description>
    <Attribute Name="twinAddress" AttributeDataType="xs:string">
      <Description>IP address of the digital twin</Description>
      <DefaultValue>127.0.0.1</DefaultValue>
    </Attribute>
    <Attribute Name="twinPort" AttributeDataType="xs:unsignedInt
      ↳ ">
      <Description>Tcp port of digital twin</Description>
      <DefaultValue>8080</DefaultValue>
      <Value>8080</Value>
    </Attribute>
  </SystemUnitClass>
  ...
</SystemUnitClassLib>
```

A seguir é apresentada uma breve descrição das SUCs criadas, agrupadas de acordo com o seu papel. Mais detalhes sobre a utilização destas SUCs na prática serão apresentados nos estudos de caso do capítulo 7.

- **Storage:** classes que especificam a forma de armazenamento dos dados do digital twin. No exemplo criou-se duas formas de criar o Banco de Dados. A primeira opção é usar o `TinyDB` que é banco de dados simples que guarda os dados em um arquivo local no formato JSON, sendo recomendado para aplicações simples. A segunda opção é usar `MySqlAdapter`, que especifica um adaptador para o banco de dados MySQL, que é um banco de dados relacional com suporte a funcionalidades de maior escala. No entanto, cabe salientar que outras opções podem ser adicionadas.

- **Communication:** classes que especificam a forma de comunicação de dados entre o digital twin e o dispositivo físico. As opções criadas são: `ModbusTcpServerSimple`, que define um servidor para o protocolo Modbus, ou `MqttClientSimple`, que define um cliente para o protocolo MQTT.
- **Method:** classes que definem métodos que podem ser executados pelo digital twin. Foram criadas três opções: 1) `PythonScript`, que especifica um script na linguagem Python a ser executado; 2) `SystemCommand`, que especifica um comando a ser executado no terminal do sistema operacional; 3) `MatlabScript`, que especifica um script na linguagem Matlab a ser executado (é necessário que o Matlab esteja instalado no servidor que hospeda o digital twin).
- **Event:** classes que definem possíveis eventos disparados pelo digital twin. Apenas uma opção foi definida no momento: `BooleanAlarm`, que permite que expressões booleanas sejam especificadas, sendo que o evento é disparado sempre que a expressão se torna verdadeira. Esta classe possui interfaces que permitem a ligação entre as variáveis da expressão com os atributos do dispositivo físico (através de uma `AttributeInterface`).
- **HMI:** classes que definem as interfaces de visualização. Três opções foram criadas: 1) `HtmlView`, que disponibiliza o acesso a páginas em Html que permitem a navegação pelos dados do dispositivo físico; 2) `Collada3DView`, que permite a visualização geométrica do dispositivo em 3D associada a um modelo no padrão Collada; 3) `NodeRedDashboard`, que especifica um *dashboard* de visualização utilizando a ferramenta Node-Red (deve estar instalada no servidor).
- **AccessControl:** classes que permitem especificar restrições de acesso aos recur-

tos e informações. A classe `UserHashAuthentication` contém a lista de usuários e o *hash* de sua senha, permitindo a autenticação de usuários; a classe `AccessWhiteList` define uma lista de recursos que podem ser acessados por um determinado usuário, sendo que os outros recursos serão bloqueados; já a classe `AccessBlackList` faz o contrário, define uma lista de recursos que não pode ser acessado por um usuário, sendo que o resto estará liberado.

5.2.4 Criação de uma hierarquia de instâncias (*Instance Hierarchy*)

Uma vez que todos os componentes foram modelados como SUCs, o modelo do digital twin pode ser criado como uma composição modular e hierárquica das instâncias desses SUCs. A Figura 31 mostra um exemplo de um modelo de digital twin para um robô que contém uma comunicação via protocolo Mqtt e um banco de dados MySQL.

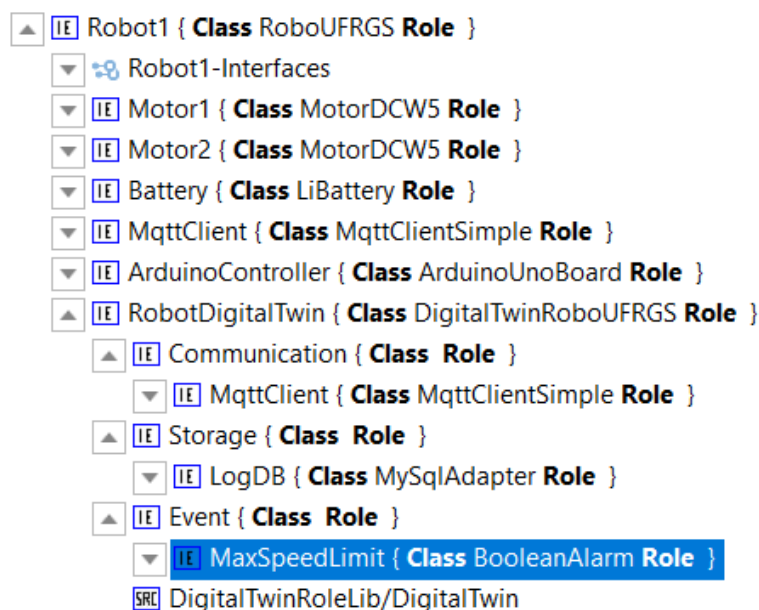


Figura 31: Exemplo da modelagem do robô para criar o seu digital twin.

Regras de modelagem são definidas para esta etapa, a fim de garantir-se a padronização estrutural do modelo:

1. Define-se que a instância do digital twin deve ser da classe `DigitalTwin` ou

derivada desta. No exemplo anterior, o modelo do digital twin é da classe `DigitalTwinRoboUFRGS` que é derivada da classe `DigitalTwin`.

2. Define-se que a instância do digital twin deve ser um filho direto de seu respectivo dispositivo físico. Ou seja, o digital twin passa a ser um componente do próprio dispositivo.
3. Para o caso do digital twin agregado, os subcomponentes do dispositivo podem conter os seus próprios digital twins que devem seguir as regras 1 e 2. Foi definido neste caso, que o digital twin destes subcomponentes serão automaticamente agregados no digital twin pai.
4. Para fins de organização e melhor compreensão, os componentes do digital twin ilustrados na Figura 31 e ao longo deste documento estão inseridos dentro de elementos com o mesmo nome do seu papel. No entanto, este padrão não é um requisito obrigatório pois não adiciona nenhuma funcionalidade extra, podendo ser suprimido se desejado.

Abaixo é mostrado parte do código XML da hierarquia, onde pode-se observar o atributo `position_x` do robô, sua interface externa e o modelo do digital twin. É importante observar também que cada elemento desta hierarquia tem um ID único automaticamente gerado pelo editor do AutomationML, o que atende o requisito de identificação dos componentes do digital twin.

```
<InstanceHierarchy Name="InstanceHierarchy1">
  <Version>1.0.0</Version>
  <InternalElement Name="Robot1" ID="b0ef0dde-5d2f-4811-8206-1
    ↪ f663cff3e15" RefBaseSystemUnitPath="UFRGSlib/RoboUFRGS">
    <Attribute Name="position_x" AttributeDataType="xs:integer">
      <Description>Posio x atual estimada</Description>
    </Attribute>
  ...
```

```

<ExternalInterface Name="position_x" RefBaseClassPath="
  ↳ DigitalTwinInterfacesLib/AttributeInterface" ID="9
  ↳ d996372-da5c-447d-a138-63f21041d01f" />
...
<InternalElement Name="RobotDigitalTwin"
  ↳ RefBaseSystemUnitPath="UFRGSLib/DigitalTwinRoboUFRGS"
  ↳ ID="b423f754-9438-407b-a956-94e3a03c1cb9">
...
</InternalElement>
</InstanceHierarchy>

```

5.2.5 Configuração dos atributos do digital twin

As SUCs do digital twin podem ter diversos atributos, tais como parâmetros de configuração e operação, que precisam ser definidos de acordo com a aplicação após a sua instanciação. Por exemplo, a própria SUC `DigitalTwin` tem dois atributos iniciais a serem configurados: o endereço IP e a porta TCP que especificam o endereço do servidor web (Figura 32). Da mesma forma, o componente `ModbusServerSimple` deve ter seu endereço definido e o mapeamento do espaço de memória do protocolo Modbus deve ser feito através da conexão com interfaces do tipo `AttributeInterface`. Outros sub-componentes também podem ter atributos que precisam ser configurados ou interfaces que devem ser vinculadas para indicar o fluxo de informações.

Em suma, a Figura 33 ilustra através de um fluxograma a sequência da modelagem do digital twin: o primeiro passo é modelar o dispositivo físico no `AutomationML`, representado no fluxograma como um sub-processo por ter a sua própria complexidade. Este modelo pode ter componentes que já possuem seus próprios digital twins, que deverão então ser modelados (esse processo gerará um digital twin agregado). Após, o modelo do digital twin deve ser adicionado no modelo do dispositivo físico (como um elemento interno) e, então os componentes que serão utilizados devem ser adicionados (armazenamento, comunicação, etc). Por fim, links entre o modelo do digital twin e o modelo do

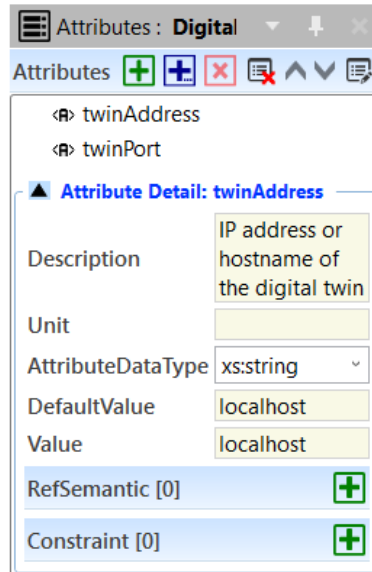


Figura 32: Atributos ‘Endereço IP’ (twinAddress) e ‘porta’ (twinPort) para a configuração do servidor WEB.

dispositivo físico devem ser criados.

5.3 Padrões de modelagem para as topologias

Cada uma das cinco topologias apresentadas na Seção 4.4 tem características específicas que se refletem em padrões de modelagem do digital twin. A seguir é feita uma análise destes padrões, que são exemplificados através de modelos ilustrativos. Estes padrões podem servir como uma referência a ser seguida durante a modelagem ou para identificar o tipo de digital twin para um dado modelo.

5.3.1 Digital twin desconectado

Nesta topologia não existe uma conexão direta entre o dispositivo físico e seu digital twin. Logo, o modelo do digital twin não deve conter nenhum componente com o papel *Communication*. A Figura 34 ilustra este caso. Não há restrições para os outros componentes.

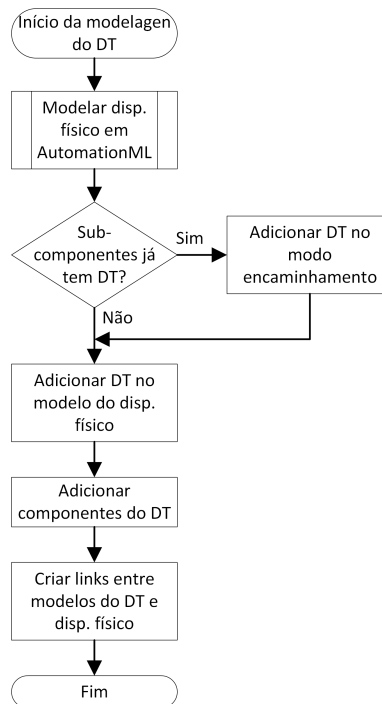


Figura 33: Fluxograma geral de modelagem do digital twin.

5.3.2 Digital twin conectado

Nesta topologia o digital twin está conectado ao seu respectivo dispositivo através de um protocolo de comunicação. Logo, o modelo do digital twin deverá necessariamente conter ao menos um componente com papel *Communication*, podendo haver mais de um se necessário. A Figura 35 ilustra este caso. Também deve haver um ou mais *links*

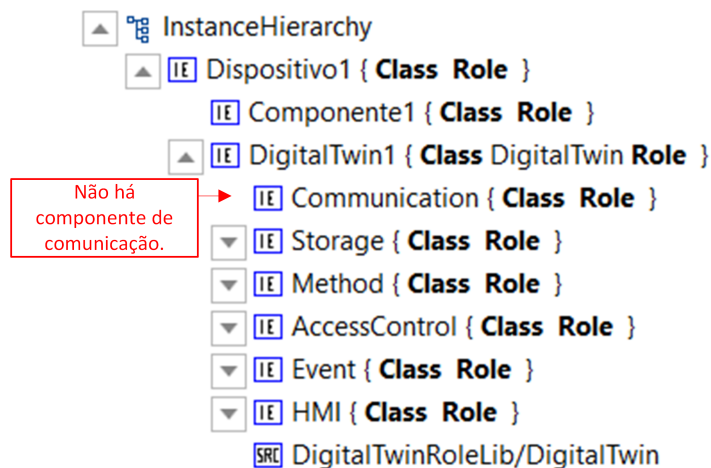


Figura 34: Exemplo de modelo de digital twin desconectado.

entre uma interface `AttributeInterface` do dispositivo físico até uma interface do componente de comunicação, indicando que o respectivo atributo está sendo comunicado sobre o canal de comunicação.

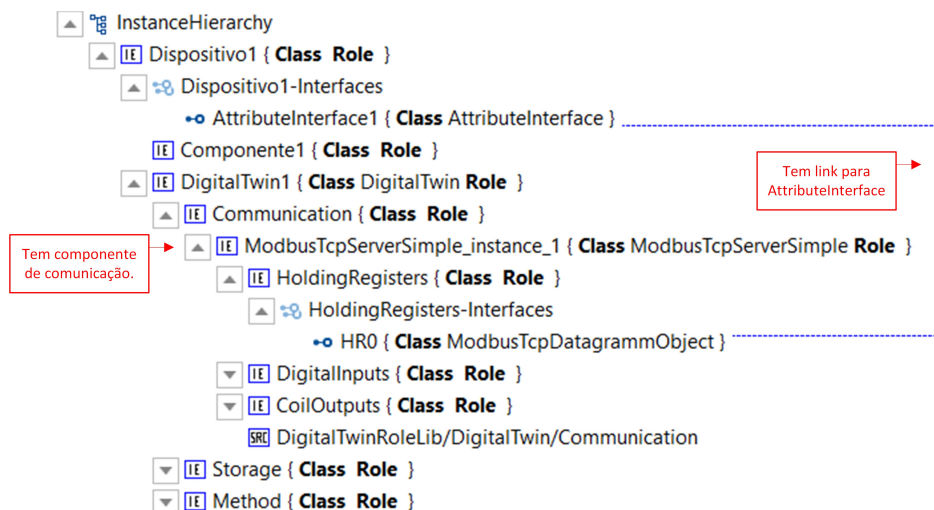


Figura 35: Exemplo de modelo de digital twin conectado.

5.3.3 Digital twin embarcado

Nesta topologia o digital twin está sendo executado no mesmo aparato computacional do seu respectivo dispositivo. Assim, as informações do dispositivo podem ser enviadas diretamente ao digital twin (através de memória compartilhada ou leitura das entradas, por exemplo), sem a necessidade de comunicação através de protocolos. Logo, não haverá componente com papel `Communication` no digital twin. Além disto, o digital twin pode ter acesso direto às interfaces internas do dispositivo (em casos que o sistema operacional não restrinja o acesso), tal como uma leitura de um sensor, permitindo que seus componentes recebam informações diretamente destas interfaces. O exemplo ilustrado na Figura 36 demonstra tal acesso direto a uma interface do tipo `SignalInterface`, que no AutomationML é geralmente utilizada para representar um sinal elétrico analógico.

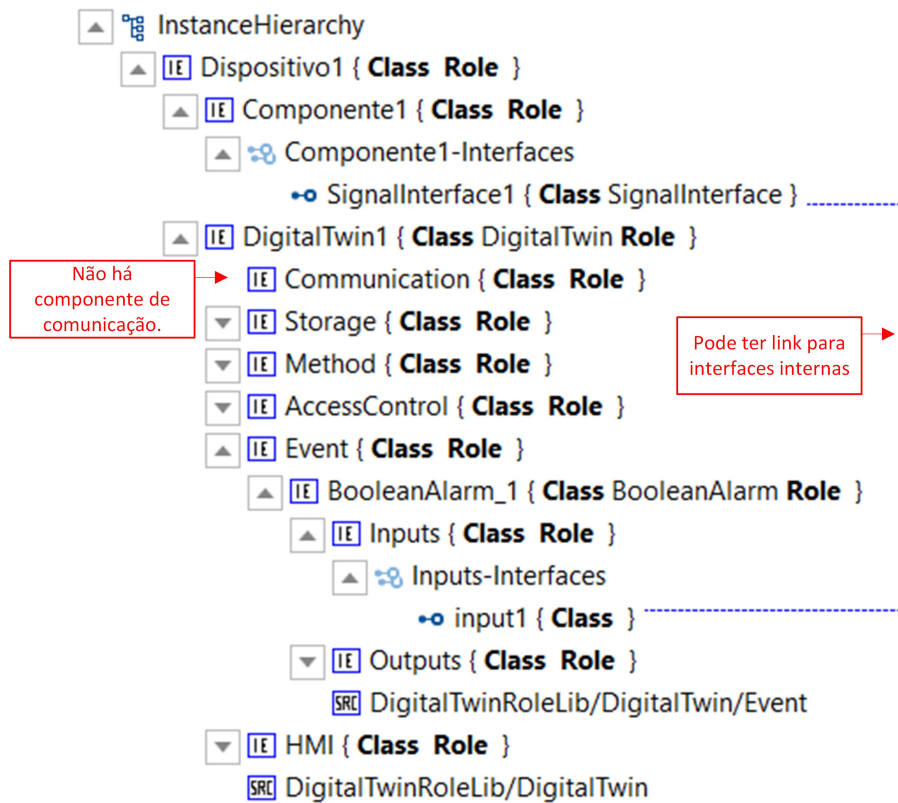


Figura 36: Exemplo de modelo de digital twin embarcado.

5.3.4 Digital twin multi-dispositivo

Nesta topologia um digital twin está conectado a mais de um dispositivo. Para modelagem desta topologia deve-se criar um componente lógico que agrupa os dispositivos e então, adiciona-se o digital twin ao grupo. Além disto, deve haver uma comunicação individual entre cada dispositivo para o digital twin; ou seja, deve existir ao menos um componente com papel `Communication` e ao menos um link para uma interface `AttributeInterface` de cada um dos dispositivos físicos. A Figura 37 ilustra este padrão.

5.3.5 Digital twin agregado

Nesta topologia o digital twin pode ser composto por outros digital twins. Para modelagem desta topologia deve-se primeiramente criar um componente lógico que agrupa os

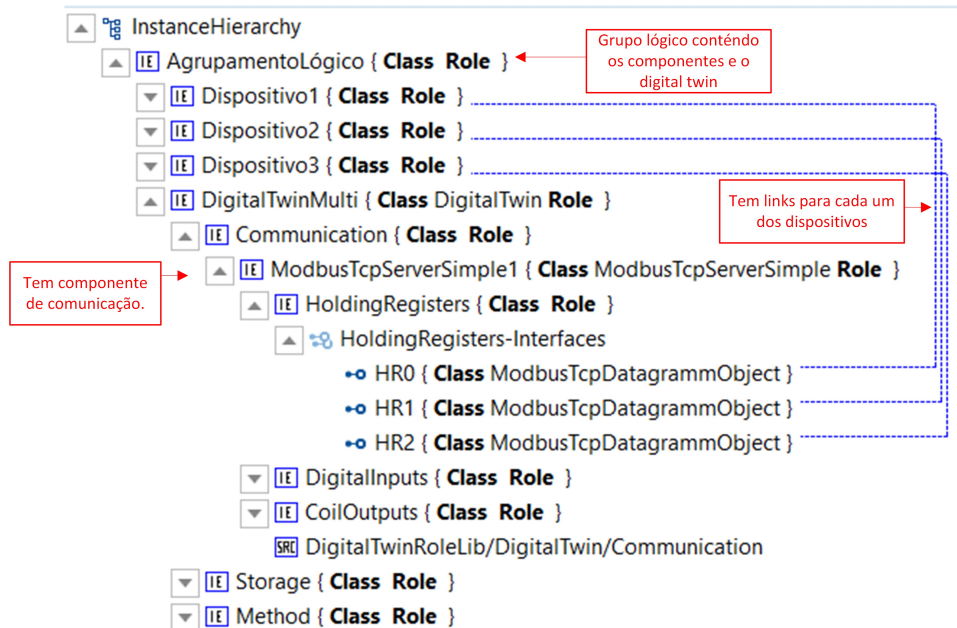


Figura 37: Exemplo de modelo de digital twin multi-dispositivo.

dispositivos a serem agregados, sendo que estes dispositivos devem ter um componente da classe `DigitalTwin`. Isto indica que este dispositivo já tem o seu digital twin. Por fim adiciona-se o digital twin ao grupo, que implicitamente agrega os digital twins dentro deste grupo. A Figura 38 ilustra este padrão.

5.4 Conclusão

Este Capítulo apresentou uma metodologia para modelagem do digital twin usando AutomationML que segue a arquitetura e modelo de referência dos capítulos anteriores.

As funcionalidades da linguagem AutomationML possibilitaram a modelagem em diferentes níveis de abstração, o que permite a generalização de conceitos. Foram criadas bibliotecas para *role classes*, *system unit classes* e interfaces, permitindo a reutilização dos componentes para a modelagem de novos digital twins de forma eficiente.

Foi demonstrado que a metodologia proposta pode ser usada para modelagem de qualquer uma das topologias identificadas no capítulo anterior, o que corrobora a alegação de

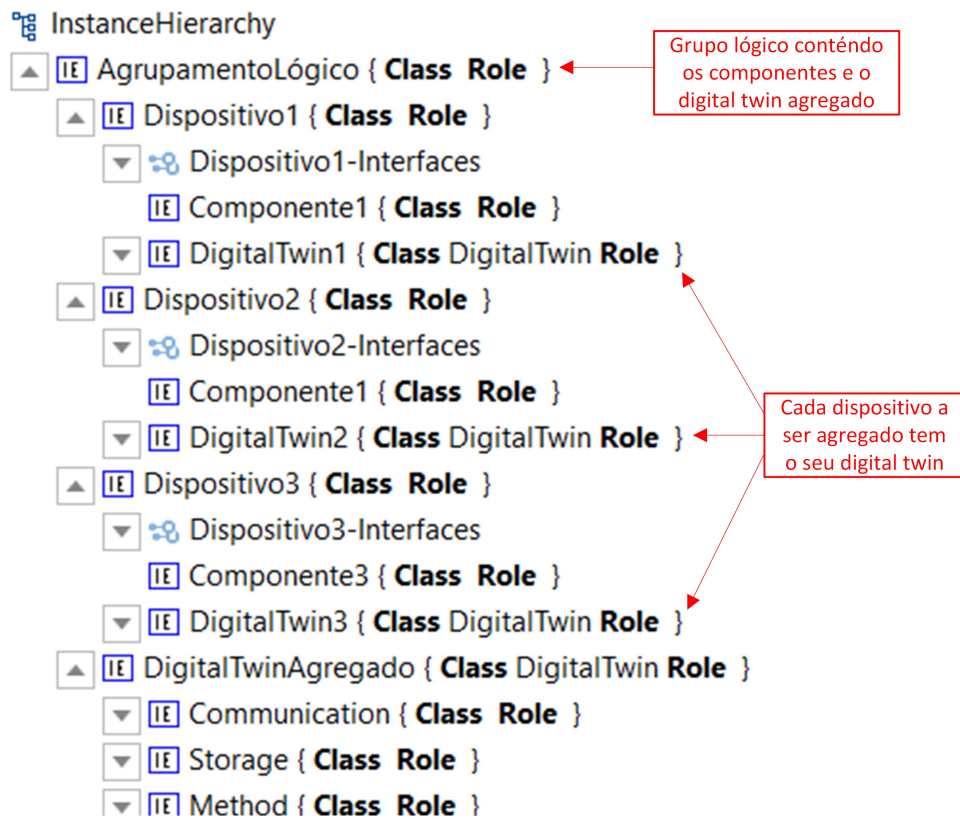


Figura 38: Exemplo de modelo de digital twin agregado.

que a metodologia é flexível e genérica.

6 IMPLANTAÇÃO AUTOMATIZADA BASEADA EM MODELO DO DIGITAL TWIN

Uma vez que o dispositivo físico e o digital twin foram modelados em Automati-onML, todas as informações necessárias para implantar e executar o digital twin estarão disponíveis em um único arquivo. Portanto, o processo de implantação do digital twin, que consiste na criação e execução dos diversos componentes do digital twin, pode ser baseado unicamente neste arquivo.

A grande maioria dos trabalhos na literatura não segue um modelo para a implantação do digital twin e o faz (presumivelmente) de forma manual, o que pode ser complexo e ineficiente na prática, principalmente em aplicações de grande escala. Uma exceção a este padrão é o trabalho de (HIRMER et al., 2016), onde é proposta a implantação do digital twin para ambientes inteligentes (*smart environments*) de forma automática a partir da descrição dos sensores e atuadores presentes no ambiente. No entanto, além de ter um escopo de aplicação restrito, a proposta também adota uma visão limitada do digital twin, cujas funcionalidades incluem apenas acesso aos dados dos dispositivo e armazenamento do histórico.

Neste capítulo é apresentada uma sistematização e automatização do processo de implantação do digital twin baseada na modelagem descrita no capítulo anterior. Inicial-

mente, na Seção 6.1 é apresentada uma visão geral da metodologia de implantação proposta. Na Seção 6.2 é proposto um software que recebe um arquivo AML contendo o modelo do digital twin e implementa todos os seus componentes de forma automatizada. O resultado da implantação é um serviço web com todas as funcionalidades em execução conforme definidas no modelo. A Seção 6.3 descreve com detalhes a API REST implementada. Por fim, a Seção 6.4 apresenta as conclusões do capítulo.

6.1 Metodologia geral de implantação

A Figura 39 mostra o fluxograma em alto nível de uma proposta para implantação do digital twin. A entrada é o arquivo em formato AML contendo o modelo do dispositivo e do digital twin e ao fim do processo tem-se um servidor web implementando o digital twin. Os seguintes passos são executados:

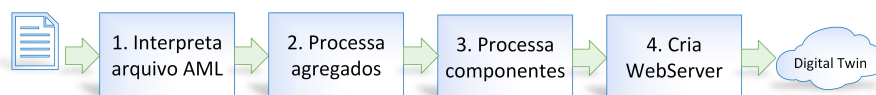


Figura 39: Fluxograma de implantação em alto nível.

1. Interpreta-se (*parse*) o arquivo em AutomationML, permitindo que os dados do modelo sejam recuperados e processados de forma eficiente quando necessários. Visto que a hierarquia de instância do AutomationML representa os elementos do modelo em forma de árvore, sugere-se o uso de uma estrutura de dados em árvore para sua representação na memória.
2. Procura-se os sub-componentes no modelo do dispositivo que já possuem seus próprios digital twins, sendo estes processados como agregados se presentes. A identificação dos digital twins agregados é importante pois durante a execução do digital

twin, todas as requisições de dados relativas a estes sub-componentes devem ser encaminhadas aos seus respectivos digital twins, e não processadas localmente.

3. Procura-se os componentes no modelo do digital twin, sendo estes criados e configurados de acordo com as especificações do modelo. O tipo do componente é identificado pela sua classe SUC, sendo cada tipo processado de acordo. O componente pode ter uma ligação à uma interface `AttributeInterface` do dispositivo, que deve ser identificada e utilizada na configuração do componente de forma apropriada.
4. Cria-se o servidor web que implementa e fornece as funcionalidades do digital twin via solicitações HTTP. Ao receber uma solicitação, o servidor deve coordenar a execução dos componentes internos do digital twin, portanto, a lógica central de operação do digital twin é implementada neste servidor web. As funcionalidades são providas via API Restfull ou, quando especificado por algum componente, via páginas HTML.

A Figura 40 expande os detalhes deste processo através de um fluxograma multifuncional mostrando a relação entre a linha principal de implantação e os modelos do dispositivo e do digital twin.

6.2 Software para automatização da implantação

Seguindo a metodologia descrita na Seção anterior, foi criado um software que automatiza o processo de implantação, dando origem ao conceito aqui proposto de digital twin baseado em modelo (*Model Based Digital Twin*). O software foi escrito na linguagem Python, sendo o principal componente a classe `DigitalTwin`, que recebe os

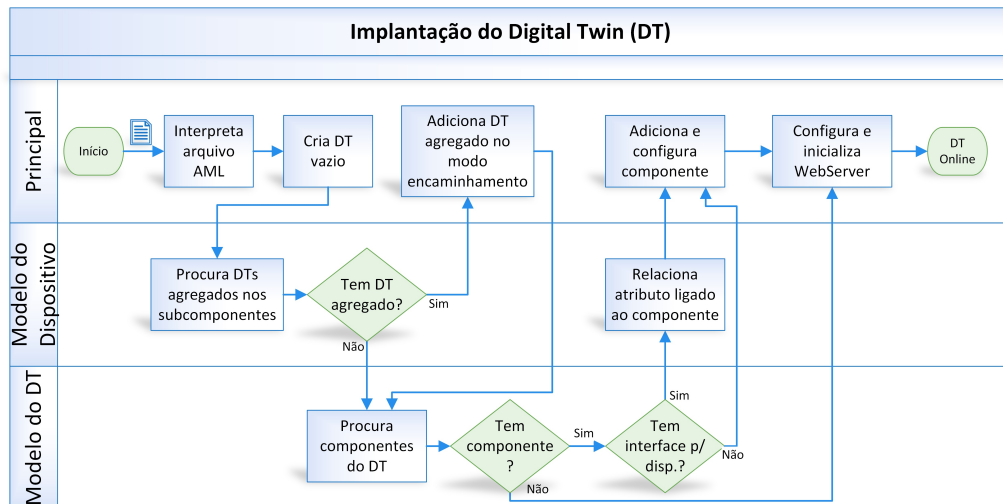


Figura 40: Fluxograma multifuncional de implantação.

modelos do dispositivo e digital twin e cria todos os componentes, incluindo o servidor web. Abaixo é mostrado parte de seu código contendo a inicialização (vazia) das suas variáveis membro, sendo que após esta inicialização, os componentes são criados conforme o modelo.

```
def __init__(self, digTwinElement, deviceElement):
    self.deviceName = deviceElement.get("Name")
    #Dados de configuracao do servidor digital twin
    self.ipAddress = digTwinElement.find("./Attribute/[@Name='
        ↳ twinAddress']").find("Value").text
    self.tcpPort = int(digTwinElement.find("./Attribute/[@Name='
        ↳ twinPort']").find("Value").text)

    #Componentes do digital twin
    self.webSrv = Bottle() #WebServer
    self.comms = [] #Comunicacao
    self.alarms = [] #Alarmes
    self.methods = [] #Metodos
    self.accessUsers = {} #Usuarios
    self.db=None #Banco de dados
    self.forwardMap = {} #DT agregados
    ...
```

A lógica principal de execução do digital twin que coordena a interação entre os componentes foi implementada dentro desta classe. Assim, por exemplo, quando um pacote de dados é recebido do dispositivo físico via algum protocolo de comunicação, sinali-

zando que um sensor esta ativo e enviando dados, as seguintes etapas são realizadas:

1. Verificação se a fonte dos dados (no caso o dispositivo físico) pode alterar a propriedade na sua representação digital do dispositivo físico, ou seja, pode alterar um atributo que representa o sensor, com base nos componentes do DT de controle de acesso.
2. Atualização da representação virtual (DT) para refletir o novo estado físico.
3. Registro da mudança no banco de dados (quando este componente está presente).
4. Verificação de eventos relacionados a esta alteração e disparo do mesmo (quando necessário).

Parte do código que implementa esta lógica pode ser visto abaixo, implementada na função membro `changeAttribute`.

```
def changeAttribute(self, elem, attribElem, value, user):
    #1) Check if user has access
    accessOK = self.canChange(elem, attribElem, user)
    if(accessOK):
        #2) Changes value
        attribElem.find("Value").text = value
        print("Attribute ", attribElem.get("Name"), "changed to", str
              ↪ (value))
        #3) Log change
        if self.db is not None:
            r = {'time' : time.asctime(time.gmtime()), 'id' : elem.get(
                  ↪ "ID"), 'attrib':attribElem.get("Name"), 'value' :
                  ↪ value }
            self.db.insert(r)
        #4) check alarms
        self.checkAllAlarms()
    else:
        print("Access denied.")
```

Além disso o software também gerencia a execução de leitura dos dados (com controle de acesso), execução de métodos, e disponibilização de HMIs, sempre de forma a seguir as especificações do modelo.

Diversas bibliotecas e frameworks foram usados para implementação deste software.

A Tabela 5 lista os principais, juntamente com uma pequena descrição das suas características e onde foi usado.

Componente	Biblioteca/ Framework	Descrição
Interpretador XML	ElementTree ¹	Biblioteca que faz parte do Python para auxílio no manuseio de arquivos XML, facilitando a leitura, navegação, edição e criação destes. Usado para interpretar o arquivo AML.
Servidor web	Bottle ²	Implementa um servidor web seguindo o padrão WSGI (<i>Web Server Gateway Interface</i>) do Python. Fornece suporte à autenticação de usuários, definição de rotas dinâmicas e uso de templates.
Servidor Modbus	uModbus ³	Implementa um servidor Modbus seguindo a especificação V1.1b3. Suporta o modo TCP e RTU. Usado na implementação da SUC <code>ModbusServerSimple</code>
Banco de dados	TinyDB ⁴	Fornece funcionalidades para armazenamento de dados em formato JSON, ideal para aplicações pequenas. Usado na implementação da SUC <code>TinyDB</code>
Renderizador WebGL	three.js ⁵	Facilita o manuseio de objetos 3D e a sua renderização usando a tecnologia WebGL. Utilizado para implementação da SUC <code>Collada3DView</code> .
Gerenciador requisições HTTP	Requests ⁶	Ferramentas para criar e enviar requisições HTTP. Usado para encaminhamento de mensagens aos digital twins agregados.

Tabela 5: Bibliotecas e frameworks utilizados na implementação do software para automatização da implantação do digital twin

6.3 Implementação da API do digital twin

O Servidor Web fornece acesso às informações do modelo físico, dados históricos e métodos através de uma API RESTful que padroniza a interação com aplicativos externos. A Tabela 6 exemplifica, baseando-se no modelo de um robô, algumas solicitações HTTP definidas pela API. Após processadas pelo servidor, as informações solicitadas são

retornadas (quando encontradas) no formato JSON. Quando a solicitação é para um digital twin agregado, o servidor web encaminha a solicitação para o agregado e retorna o resultado para o solicitante, de forma que seja transparente do ponto de vista externo.

Requisição HTTP	Descrição
GET */Motor1	Solicita todos os dados atuais do componente <i>Robot/Motor1</i>
GET *?Attribute=speed	Solicita somente o valor atual do atributo <i>speed</i>
GET *?History=50	Solicita as últimas 50 alterações no componente <i>Robot</i>
GET *?Attribute=speed&History=6	Solicita as últimas 6 alterações para o atributo <i>speed</i>
GET *?Search=[@Name='Motor1']	Solicita os endereços (URL) de todos os componentes que satisfazem o critério de pesquisa (Name='Motor1') usando a sintaxe XPath
POST *?Attribute=speed&Value=20	Altera o valor atual do atributo <i>speed</i> para 20

Tabela 6: Exemplos de uma requisição Restfull API. O Prefixo * representa `http://<ipaddress>:<port>/api/Robot`

6.4 Conclusão

Neste capítulo, um software que automatiza a implantação do digital twin baseado em modelo foi proposto. Foi apresentada uma visão geral da sua lógica de funcionamento e das ferramentas utilizadas para a implementação de seus componentes, onde destaca-se a implementação das SUCs, que agregam as funcionalidades ao digital twin. A implementação da API do digital twin foi descrita, permitindo a padronização das operações de leitura e alteração dos dados via solicitações HTTP.

A agilidade na implantação de um sistema é crucial no contexto da Indústria 4.0, sendo que a automatização deste processo pode permitir grandes ganhos de eficiência, principalmente na implantação de sistemas complexos e que necessitem constantemente

ser reconfigurados para se adaptar a novas exigências do mercado.

7 ESTUDOS DE CASO

Este capítulo apresenta a validação da arquitetura, modelagem e metodologia de implantação propostas nesta Tese com dois estudos de caso. O primeiro, com um equipamento de sistema de automação industrial: uma válvula industrial e, o segundo, com um robô usado na educação universitária na disciplina de Introdução a Engenharia de Controle e Automação. Em ambos os casos será feita a modelagem completa do digital twin em AutomationML assim como a sua implantação. Baseando-se neste processo e nos resultados observados será feita uma análise sobre a validade da proposta.

7.1 Estudo de Caso 1: Válvula Industrial

O dispositivo físico usado neste estudo de caso foi uma válvula industrial modelo CSR6 produzida pela empresa Coester Automação. A Coester desenvolve atuadores para a indústria de petróleo e gás, sendo o atuador utilizado pela empresa Petrobras. A válvula abre e fecha permitindo que algo passe através dela, sendo que este trabalho pode ser realizado de forma automatizada, sem intervenção manual, com acionamento a distância via protocolo de comunicação industrial. Além disso, a válvula permite o monitoramento de seu estado de funcionamento, oferecendo leitura do torque durante operação, temperatura, nível da bateria e o valor da tensão de alimentação.

A Figura 41 ilustra o conjunto válvula-atuador usado no estudo de caso. Essa figura mostra alguns dos componentes pertencentes a válvula como por exemplo, o volante manual, o atuador elétrico, a caixa de engrenagens, etc. Segundo o manual da válvula (CO-ESTER, 2018), o conjunto válvula-atuador é composto pelos seguintes elementos: tubo protetor da haste, volante manual, painel de comando local, placa controladora, sensor de posição, transdutor de torque, sensor térmico do motor, alimentação trifásica, engrenagens satélites e, motor.

Os atuadores elétricos para válvulas, têm como características básicas:

- Movimentar válvulas e outros elementos como comportas, dampers, etc., através do acionamento feito por motores elétricos, trifásicos ou monofásicos, de corrente alternada ou corrente contínua.
- Comandar com segurança os movimentos para o conjunto Atuador/Válvula, bem como para a instalação em geral, através de módulos de controle de posição (mecanismos contadores de voltas) e de sensores limitadores do torque de saída e, quando aplicável, modular o posicionamento contínuo do conjunto, através de rastreamento, normalmente com sinais de controle entre 4 e 20 mA.
- Permitir o acionamento manual do conjunto, quando da impossibilidade de fazê-lo eletricamente, seja pela falta momentânea de alimentação, seja no comissionamento da obra.

7.1.1 Objetivo

Neste estudo de caso tem-se como objetivo criar um digital twin para a válvula seguindo a arquitetura padrão de modelagem e metodologia de implantação propostas ao longo desta tese. Outro objetivo deste estudo de caso é implantar e analisar o digital

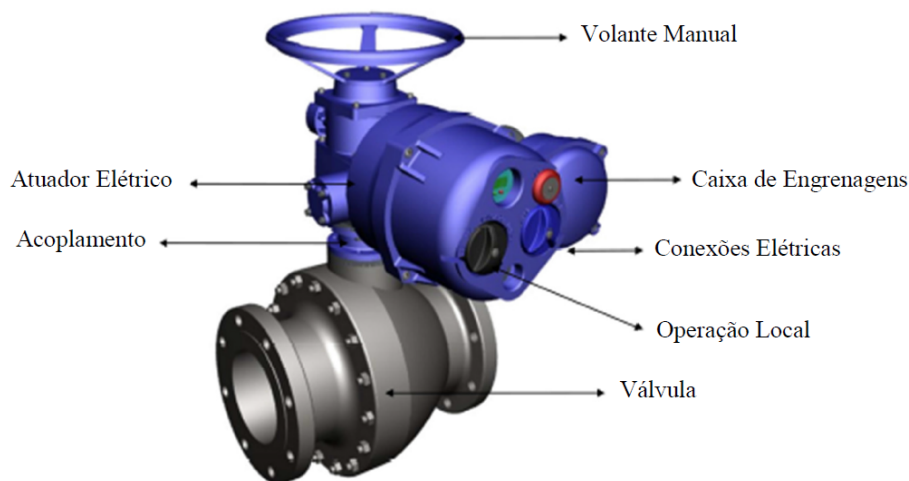


Figura 41: Conjunto Válvula-Atuador

twin sob diferentes topologias através da criação de quatro cenários diferentes que implementam o digital twin conectado, multi-dispositivo, agregado e utilizando o framework Fiware. Uma comparação entre estes quatro cenários pode destacar as vantagens e desvantagens de cada topologia, bem como apontar melhorias para futuras arquiteturas.

Como objetivo para o digital twin especificamente, foi definido que ele deve conter as seguintes funcionalidades:

1. Comunicação via protocolo Modbus com a válvula, recebendo informações do seu estado de operação, valor do torque durante acionamento, temperatura e nível da bateria.
2. Manutenção de um log (histórico) de operação da válvula, permitindo acesso a estes dados quando forem requisitados.
3. Acionamento de um sistema de manutenção inteligente (Watchdog) que monitora o padrão de operação da válvula e gera um índice de confiança relacionado à possibilidade da válvula apresentar falhas futuramente.
4. Acionamento de um simulador de comunicação Modbus que envia dados aleatórios

da válvula para fins de teste.

5. Autenticação de usuários, permitindo o acesso ao digital twin somente para os usuários devidamente cadastrados.
6. Disparo de alarme de torque elevado quando o nível aferido passar de $15N.m$ e disparo de alarme quando o índice de confiança calculado pelo Watchdog for inferior que 70%.
7. Navegação pelos dados do modelo da válvula através de um navegador web, possibilitando a visualização de seus componentes e atributos.
8. Visualização de uma representação tri-dimensional (3D) da válvula via navegador web, permitindo a interação com o objeto 3D.
9. Visualização dos dados de operação (torque, temperatura, nível da bateria e nível de confiança) através de um *dashboard* via navegador web.

Neste estudo de caso, os dados para o sensor de torque foram simulados através da reprodução de um conjunto de dados previamente adquiridos de operações do mundo real. Estes dados foram reproduzidos em um cliente Modbus ligado em rede, simulando assim a abertura e fechamento da válvula. Essa abordagem permite a comparação entre experimentos nas mesmas condições de funcionamento.

Primeiramente, é apresentada nas próximas sub-seções a implementação dos componentes que dão suporte às funcionalidades do digital twin descritas acima e são utilizados nos experimentos, sendo seguido da apresentação dos quatro cenários propostos.

7.1.2 Modelagem do dispositivo físico no AutomationML

Como proposto no Capítulo 5, o primeiro passo para o início da implantação do digital twin é a modelagem do dispositivo físico no AutomationML. A Figura 42 ilustra parcialmente o modelo criado. A principal fonte de dados utilizada na criação do modelo foi o manual de instalação e operação da válvula (COESTER, 2018), que inclui diversas informações técnicas relevantes.

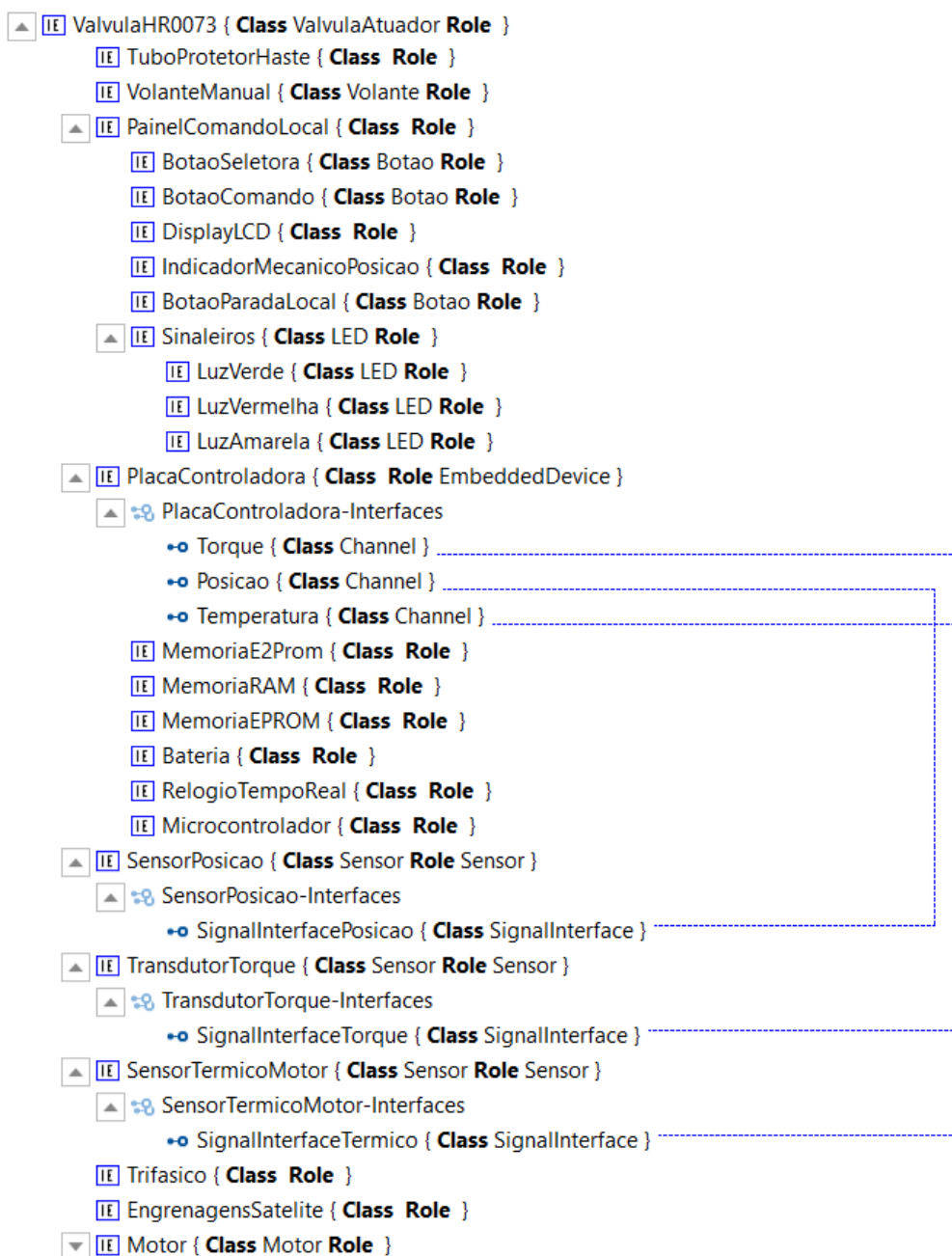


Figura 42: Modelo da válvula-atuador Coester CSR6 em AutomationML

Deve-se observar que muitos dos componentes deste modelo são instâncias de classes SUCs previamente criadas. Na Figura 42 o volante manual, por exemplo, é uma instância da classe `Volante` e, os sensores presentes (torque, temperatura e posição) são da classe `Sensor`. As botoeiras: `BotaoComando`, `BotaoSeletora` e `BotaoParadaLocal` são da classe `Botao`. E as luzes de sinalização (vermelha, verde e amarela) são instancias da classe `LED`. O fato de os elementos no modelo terem uma classe atribuída é importante porque contribui para a agregação de semântica ao modelo, além de facilitar a modelagem por proporcionar a reutilização de componentes.

Observa-se também que a placa controladora possui uma interface de comunicação com os sensores do atuador. Essa interface serve para a placa controladora receber os valores dos sensores de torque, posição e temperatura. Para isso, no entanto, nota-se que os sensores também possuem interfaces que são do tipo `SignalInterface` (significa que a comunicação é feita enviando os sinais elétricos).

7.1.3 Modelagem da Geometria 3D

O modelo tri-dimensional (3D) da válvula-atuador em formato CAD foi fornecido pela empresa Coester Automação. No entanto, a linguagem AutomationML não possui suporte nativo a este formato, sendo suportado atualmente somente modelos 3D no formato Collada. Seria possível a adição de suporte a modelos CAD ao AutomationML, visto que ela é uma linguagem aberta e flexível, mas preferiu-se fazer a conversão do modelo CAD para formato Collada usando a ferramenta SolidWorks eDrawings¹.

O arquivo Collada pode então ser vinculado ao modelo da válvula e do digital twin usando um componente `Collada3DView` no modelo do digital twin. O componente `Collada3DView` especifica que o digital twin deve fornecer uma página HTML com a

¹<http://www.edrawingsviewer.com/>

visualização em 3D do arquivo dado como entrada. A página em HTML foi implementada usando a tecnologia WebGL, onde a renderização do objeto 3D é feita diretamente no cliente, permitindo a visualização e interação com o objeto via navegador web de forma fluída. Como suporte a este processo foi utilizado o framework em javascript ThreeJS (<https://threejs.org/>). A Figura 43 demonstra o resultado obtido.

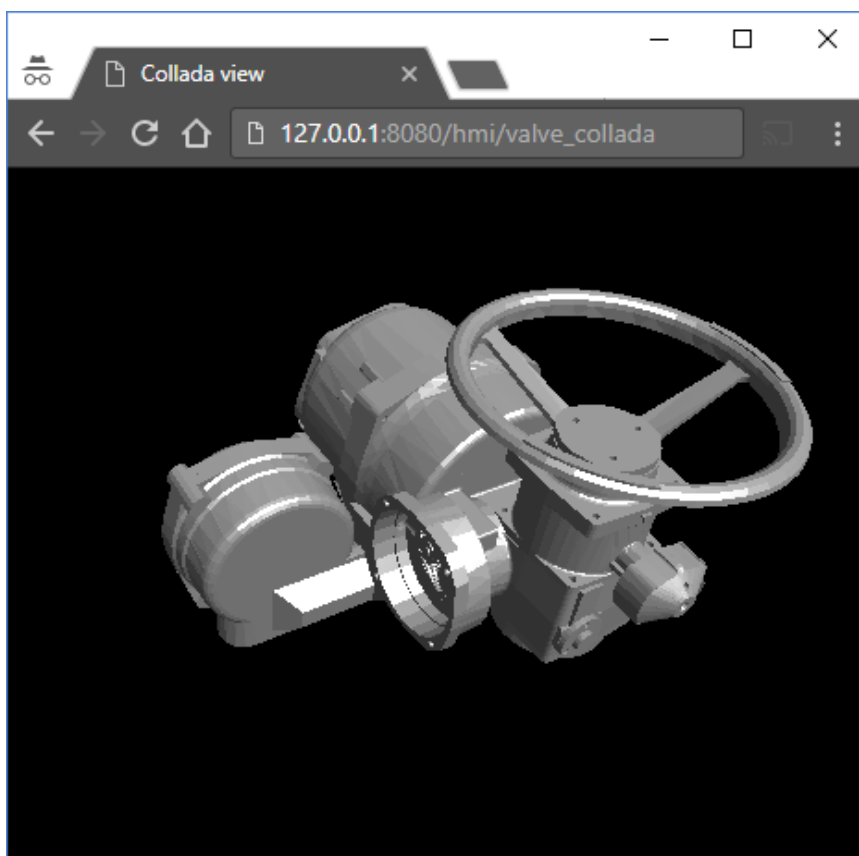


Figura 43: Visualização 3D da válvula via digital twin.

7.1.4 WatchDog

O *watchdog* é um tipo de sistema de manutenção inteligente (*Intelligent Maintenance Systems - IMS*), ou sistemas preditivos, e tem por objetivo prever quando falhas podem ocorrer nos equipamentos industriais, geralmente usando a estratégia de manutenção baseada em condições ou CBM (*Condition Based Maintenance*). A manutenção baseada em condições visa reduzir o número de falhas de equipamentos monitorando a condição do

equipamento. Assim, permitem que ações corretivas sejam realizadas antes que a falha ocorra.

Para isso, estes sistemas usam ferramentas computacionais e sensores embarcados nos equipamentos a serem monitorados. Estes sensores fornecem sinais que permitem o monitoramento do desempenho do equipamento ao longo do tempo. Baseado no desempenho previsto, os agentes de diagnóstico (componentes de hardware e software embarcados) predizem o potencial de falha antes que esta ocorra, promovendo assim a manutenção preditiva (ESPÍNDOLA, 2011).

O *watchdog* para a válvula foi implementado com base no trabalho de (DJURDJANOVIC; LEE; NI, 2003). Foi realizada uma adaptação do algoritmo proposto para que este tipo de aplicação pudesse ser chamada como método do digital twin. Nesta adaptação, o *watchdog* foi implementado com uma rede neural multi-camadas que recebe um vetor contendo 21 medidas de torque durante a última operação de abertura ou fechamento da válvula e produz um nível de confiança entre 0 e 1 indicando a condição de desgaste - 0 significando condição ruim e 1 significando condição boa.

O implementação e treinamento da rede neural foram realizados utilizando-se a *toolbox* de redes neurais do software Matlab. As principais características da rede são as seguintes:

- Possui 21 neurônios de entrada com função de transferência linear.
- Possui uma camada intermediária com 40 neurônios e função de transferência sigmoidal.
- Possui 1 neurônio de saída com função de transferência linear.
- Emprega o algoritmo de gradiente descendente para treinamento dos pesos.

A rede foi treinada em um conjunto de dados de 2000 amostras de operação (abertura ou fechamento da válvula) adquiridas em situações do mundo real fornecidas pela empresa Coester Automação, onde 50% delas são de válvulas em boa condição e as outras 50% são de válvulas que demonstram sinais de falha. Cada operação contém 21 leituras do torque adquiridas ao longo da operação. Após o treinamento, a rede alcançou uma precisão de 99,2% em um conjunto de dados de teste.

A índice de confiança final da válvula é calculado através da média ponderada das 20 últimas operações. Seja $c(i)$ a saída da rede neural após processar os dados de torque para a operação i . O índice de confiança IC da válvula é calculado através da seguinte equação:

$$IC = \frac{\sum_{i=N-20}^N w(i)c(i)}{\sum_{i=N-20}^N w(i)}, \quad (1)$$

onde N é o número total de operações e $w(i)$ é o peso da operação i definido por uma função exponencial que dá importância maior para operações mais recentes conforme a equação:

$$w(i) = e^{0.1(i-N)} \quad (2)$$

Para integração do watchdog com o digital twin, foi desenvolvido um *script* no Matlab que adquire os dados das últimas 20 operações da válvula via digital twin, executa o algoritmo watchdog e reporta o valor de confiança. Uma vez acionado, este *script* repete a execução do watchdog a cada hora e se comunica com o digital twin exclusivamente através da sua API.

O *script* com o watchdog pode então ser vinculado a um componente `MatlabScript`

no modelo do digital twin quando necessário, permitindo seu acionamento remotamente via uma mensagem HTTP POST para a API do digital twin.

7.1.5 Dashboard

Uma aplicação *dashboard* para a web foi desenvolvida manualmente, como mostrado na Figura 44 para monitorar a condição das válvulas. A aplicação foi escrita usando a ferramenta de programação gráfica Node-Red ², que permite conectar dispositivos de hardware, APIs e serviços on-line através de fluxos de informação usando diversos componentes pré-definidos.

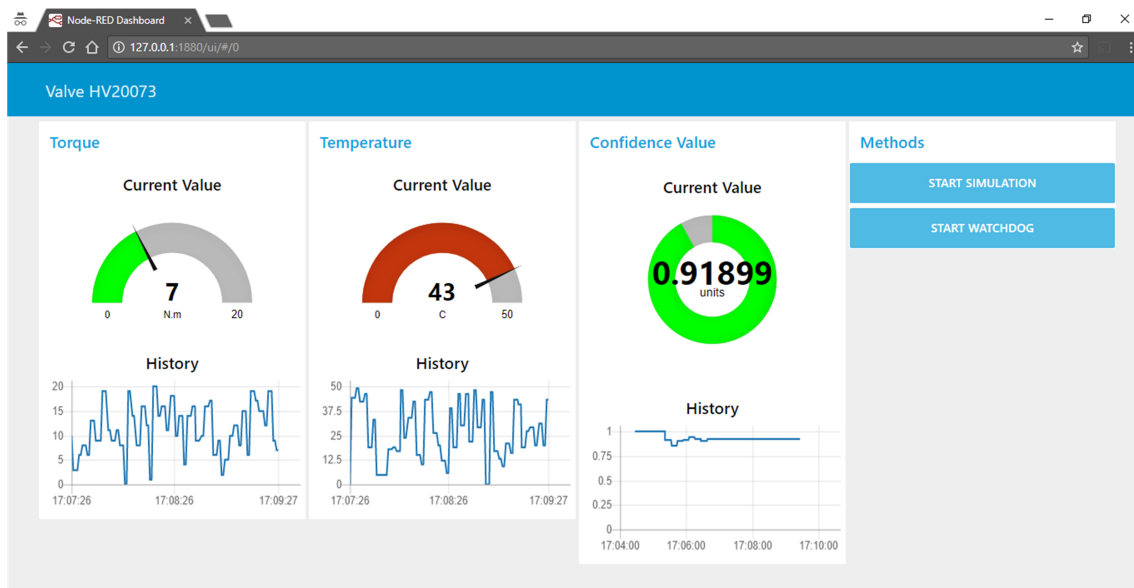


Figura 44: Dashboard da válvula.

A Figura 45 mostra o editor de fluxos do NodeRed com o fluxo criado, que basicamente solicita constantemente as informações do digital twin via solicitação HTTP, processa a resposta em formato JSON e atualiza os componentes visuais da *dashboard* em formato HTML. Uma vez criado, o fluxo pode ser exportado para um arquivo que armazena a configuração do fluxo em formato JSON.

O arquivo JSON com o fluxo da *dashboard* pode então ser vinculado a um com-

²NodeRed: <https://nodered.org/>

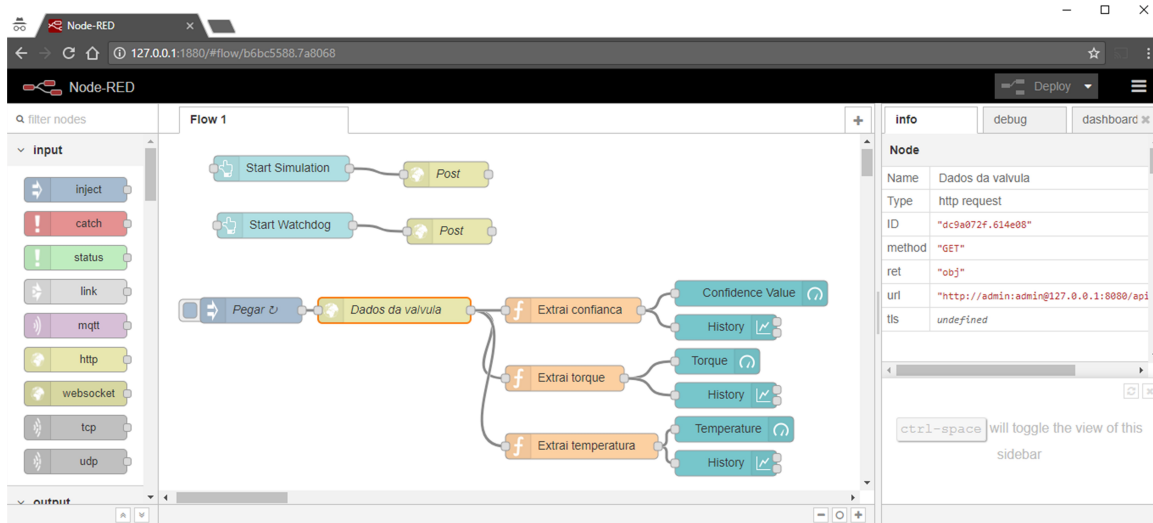


Figura 45: Fluxo NodeRed para a dashboard da válvula.

ponente RedNodeDashboard no modelo do digital twin, permitindo a integração do *dashboard* ao digital twin quando necessário.

7.1.6 Cenário 1: Digital Twin Conectado

Neste cenário existe apenas uma válvula que deve ser conectada ao seu respectivo digital twin através do protocolo Modbus (Figura 46). Este único digital twin deve fornecer todas as funcionalidades que suprem os objetivos deste estudo de caso conforme elencados na Seção 7.1.1.

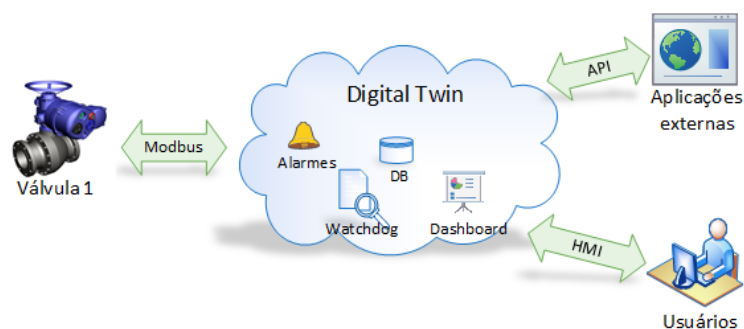


Figura 46: Visão geral do cenário 1 - Digital Twin conectado.

Uma vez que o modelo do dispositivo já está pronto, o segundo passo proposto na metodologia é a modelagem do digital twin. Para isso, foram utilizadas as SUCs apresen-

tadas na Seção 5.2.3 para criar uma nova instância do digital twin que deve estar dentro do modelo do dispositivo conforme padrão especificado na metodologia. O modelo resultante é mostrado de forma parcial (não expandida) na Figura 47. A numeração em vermelho ao lado de cada componente indica a funcionalidade que o componente está implementando conforme listadas na Seção 7.1.1.

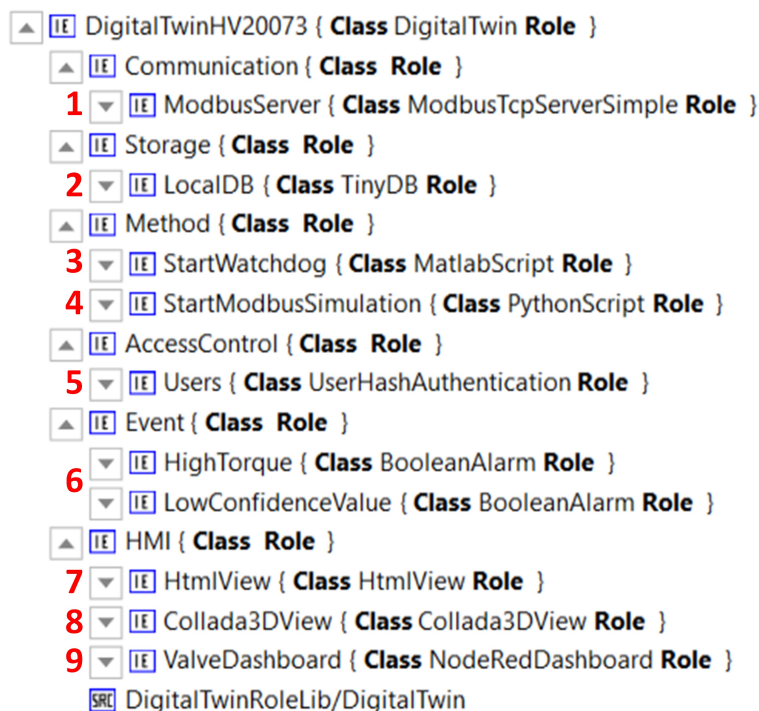


Figura 47: Modelo do digital twin para o cenário 1.

Visto que o editor do AutomationML permite que as SUCs sejam instanciadas na hierarquia de instâncias, foi possível perceber que a modelagem do digital twin é facilitada devido à reutilização dos componentes modelados nas bibliotecas de SUCs do digital twin.

Finalizada a modelagem e a configuração do digital twin, o software de implantação descrito na Seção 6.2 é executado, resultando na criação e operacionalização de todos os componentes. Para fins de teste, diversas mensagens HTTP foram criadas e enviadas ao digital twin através do software Postman³, que é uma ferramenta que permite a edição e

³<https://www.getpostman.com/>

envio de mensagens via protocolo HTTP. A Tabela 7 mostra as mensagens enviadas e os resultados obtidos.

Requisição:	GET *
Descrição:	Solicita informações gerais do dispositivo. Retorna todos os atributos e elementos internos.
Resultado:	{"attributes": {"ConfidenceValue": "1", "Torque": "0"}, "internal_elements": {"TuboProtetorHaste": null, "PlacaControladora": null, "Trifasico": null, "DigitalTwinHV20073": "DigitalTwinLib/DigitalTwin", "Motor": "ComponentesValvula/Motor", ...}}
Requisição:	GET *?Attribute=ConfidenceValue
Descrição:	Solicita apenas valor atual do atributo ConfidenceValue. Retorna o valor.
Resultado:	{"value": "1"}
Requisição:	GET *?History=2
Descrição:	Solicita últimos dois registros de alterações em geral. Retorna o número do registro seguido da data e valor.
Resultado:	{"5666": {"time": "Thu Feb,1 16:47:21 2018", "attrib": "Torque", "value": 3, "id": "0340e005-29f0-4fa9-861b-39c406ac6b77"}, "5667": {"time": "Thu Feb,1 16:47:58 2018", "attrib": "Torque", "value": 13, "id": "0340e005-29f0-4fa9-861b-39c406ac6b77"}}
Requisição:	GET *?Attribute=ConfidenceValue&History=2
Descrição:	Solicita últimos dois registros de alterações do atributo ConfidenceValue. Retorna o número do registro seguido da data e valor.
Resultado:	{"5658": {"time": "Mon Jan 22 13:53:47 2018", "attrib": "ConfidenceValue", "value": "0.49629", "id": "0340e005-29f0-4fa9-861b-39c406ac6b77"}, "5659": {"time": "Mon Jan 22 13:54:00 2018", "attrib": "ConfidenceValue", "value": "0.97108", "id": "0340e005-29f0-4fa9-861b-39c406ac6b77"}}
Requisição:	GET *?Search=.*[@RefBaseSystemUnitPath="ComponentesValvula/Sensor"]
Descrição:	Procura componentes que sejam da classe Sensor. Retorna a URL dos componentes encontrados.
Resultado:	{"9f150b1d-549f-4895-af1c-ce2adf281b0b": "http://127.0.0.1:8080/api/ValveHV20073/SensorTermicoMotor", "19232297-263c-4218-8e9b-94e5ff1ec114": "http://127.0.0.1:8080/api/ValveHV20073/SensorPosicao", "8f1cb442-c6ad-49fc-ae6-540f1d9a7df8": "http://127.0.0.1:8080/api/ValveHV20073/TransdutorTorque"}

Tabela 7: Testes de requisições ao digital twin do cenário 1. O Prefixo * representa <http://127.0.0.1:8080/api/ValveHV20073>

Além disto, todas as IHMs (página web, dashboard, visualização 3D) foram testadas

com resultados positivos. Por exemplo, a Figura 48 mostra a página Html gerada pelo componente `HtmlView`, que oferece a navegação pelos dados do digital twin mostrando os seus elementos internos, atributos e o histórico de mudanças, além de gerar um código QR que permite a identificação única de cada componente.



The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8080/hmi/aml_view/ValveHV20073`. The page content is as follows:

ValveHV20073

Class:
UFRGSlib/ComponentesGenericos/ValvulaAtuador

ID: 0340e005-29f0-4fa9-861b-39c406ac6b77

Internal elements:

Name	Class
TuboProtetorHaste	null
PlacaControladora	null
Trifasico	null
DigitalTwinHV20073	DigitalTwinLib/DigitalTwin
Motor	ComponentesValvula/Motor

Attributes:

Name	Value
ConfidenceValue	1
Torque	0

Log:

Date and time	Attribute	Value
Mon Jan 22 13:53:47 2018	ConfidenceValue	0.49629
Mon Jan 22 13:54:00 2018	ConfidenceValue	0.97108

Figura 48: Página Html gerada pelo componente `HtmlView` do digital twin.

7.1.7 Cenário 2: Digital Twin Multi-dispositivo

Neste cenário assume-se que existem quatro válvulas que devem ser conectadas a um único digital twin através do protocolo Modbus, caracterizando portanto a topologia multi-dispositivo (Figura 49). Um exemplo hipotético desta situação seria em uma refinaria que possui quatro válvulas em um duto de operação e é desejado implementar um único digital twin de forma a se compartilhar os recursos computacionais e funcionais.

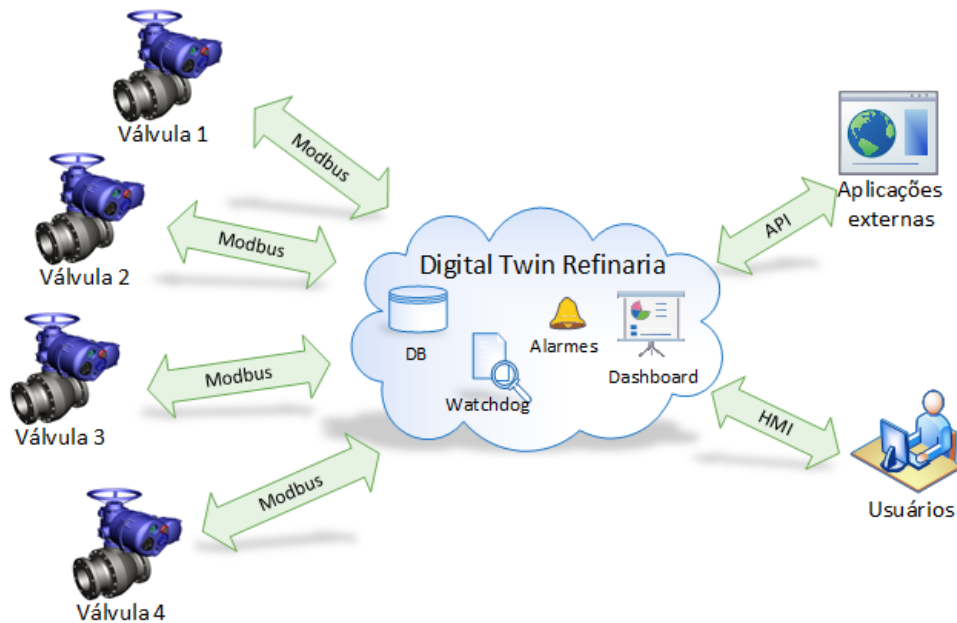


Figura 49: Visão geral do cenário 2 - Digital Twin Multi-dispositivo .

Seguindo a metodologia de modelagem proposta para esta topologia, deve-se criar um grupo lógico contendo os dispositivos e o seu respectivo digital twin, sendo que os demais passos de modelagem e configuração se assemelham ao cenário anterior, sendo portanto omitidos em grande parte. O modelo resultante é mostrado na Figura 50, onde o grupo lógico foi nomeado Refinaria para fins ilustrativos, seguindo o exemplo hipotético mencionado anteriormente.

Pode-se observar que os recursos do digital twin (banco de dados, servidor Modbus, etc) estão sendo compartilhados entre as válvulas pois não foi preciso criar um novo componente para cada dispositivo.

Os testes executados no cenário 1 foram repetidos, sendo que a única alteração foi na URL de acesso das válvulas, que passa a incluir o grupo lógico: `http://127.0.0.1:8080/api/Refinaria/ValveHV20073`. Pode-se observar, no entanto, que o sistema watchdog se adaptou de forma automática a esta alteração visto que ele procura as válvulas (usando a API) pela sua classe SUC. Isso o torna robusto a qualquer alteração na

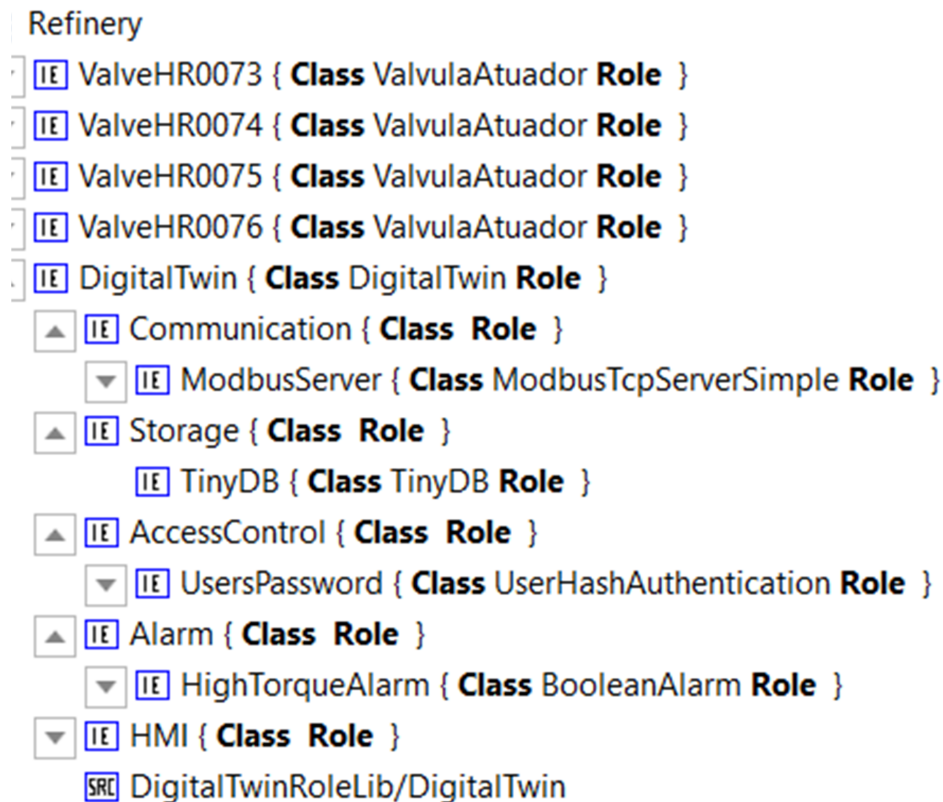


Figura 50: Modelo do digital twin para o cenário 2.

topologia do dispositivo, incluindo a adição de novas válvulas.

Foi possível observar uma elevação de aproximadamente 300% no processamento exigido pelo servidor web⁴. Isto era esperado, visto que a quantidade de dados e solicitações (via Modbus e API) cresce proporcionalmente ao número de válvulas. Isto pode se tornar problemático em sistemas grandes com muitos componentes.

7.1.8 Cenário 3: Digital Twin Agregado

Neste cenário assume-se que existem quatro válvulas que possuem o seu próprio digital twin e deve-se implementar um digital twin que agregue as funcionalidades deles, caracterizando portanto a topologia de digital twin agregado. É assumido que os digital twins das válvulas possuem um banco de dados local e os alarmes de monitoramento, en-

⁴Um laptop com processador Intel Core-i7 com 8GB de memória RAM foi usado nos experimentos, exibindo uma carga de processamento médio de aproximadamente 35% durante este experimento. A carga de processamento foi analisado usando o gerenciador de tarefas do Windows.

quanto o digital twin agregado deve conter o watchdog e a dashboard. A Figura 51 ilustra a topologia dos componentes deste cenário.

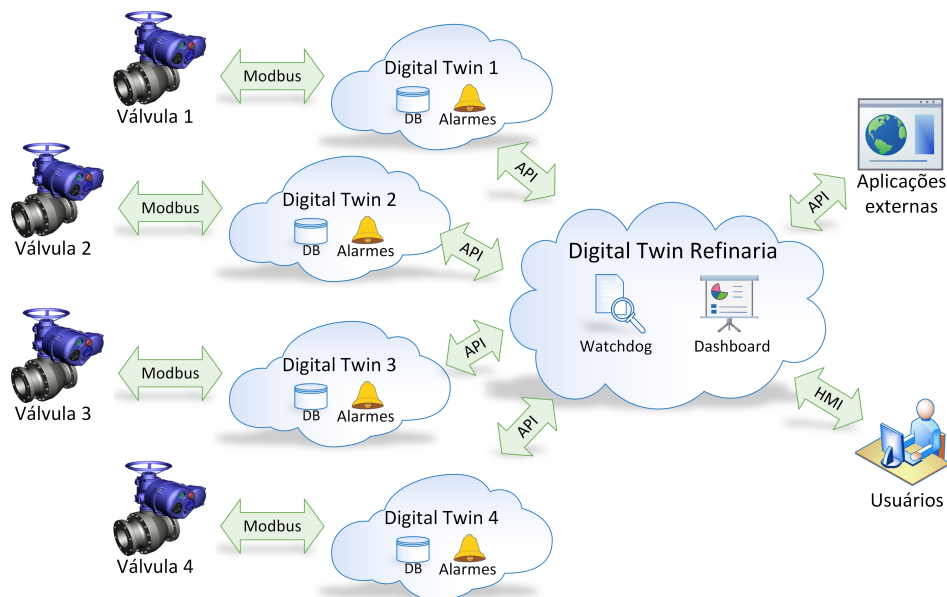


Figura 51: Visão geral da topologia do cenário 3.

Um exemplo hipotético para este cenário seria se as válvulas já tivessem uma implementação do digital twin e deseja-se agora criar um novo digital twin para a refinaria que não apenas forneça as funcionalidades já existentes mas também possibilite a adição de novas funcionalidades.

Seguindo a metodologia proposta para esta topologia, deve-se adicionar ao modelo de cada válvula um digital twin individual e outro digital twin para o grupo lógico, onde fica implícito portanto que os dispositivos serão agregados. Diferentemente do cenário anterior, o digital twin do grupo (agregado) não terá comunicação direta com as válvulas pois isto fica sob responsabilidade dos digital twins individuais das válvulas. A comunicação entre o agregado e individual é feita exclusivamente através da API e é gerenciada automaticamente pela lógica de execução do digital twin agregado através do encaminhamento de mensagens.

O modelo em AutomationML resultante é mostrado na Figura 52 e suas características

são destacadas em seguida:

- Cada válvula tem seu próprio digital twin, executando em processos diferentes e com configurações TCP/IP diferentes. Cada digital twin está conectado a uma válvula via protocolo Modbus.
- O digital twin de cada válvula tem sua própria base de dados local.
- Todas as solicitações relativas as válvulas serão encaminhadas para os seus respectivos digital twins.
- O sistema watchdog no Matlab e o painel de controle (*dashboard*) Node-Red são aplicações executadas no digital twin agregado. Uma vez que estas aplicações acessam o digital twin da refinaria por meio de sua API não há diferença para o cenário 2, não sendo necessário nenhuma alteração.



Figura 52: Modelo do digital twin para o cenário 3.

Os mesmos experimentos do cenário 2 foram realizados, sendo que os resultados foram idênticos, o que demonstra que foi possível alterar a topologia interna do sistema sem alterar os resultados de um ponto de vista externo.

O diagrama de sequência da Figura 53 mostra o processo de execução do watchdog, onde pode-se perceber o encaminhamento de mensagens para os digital twins das válvulas.

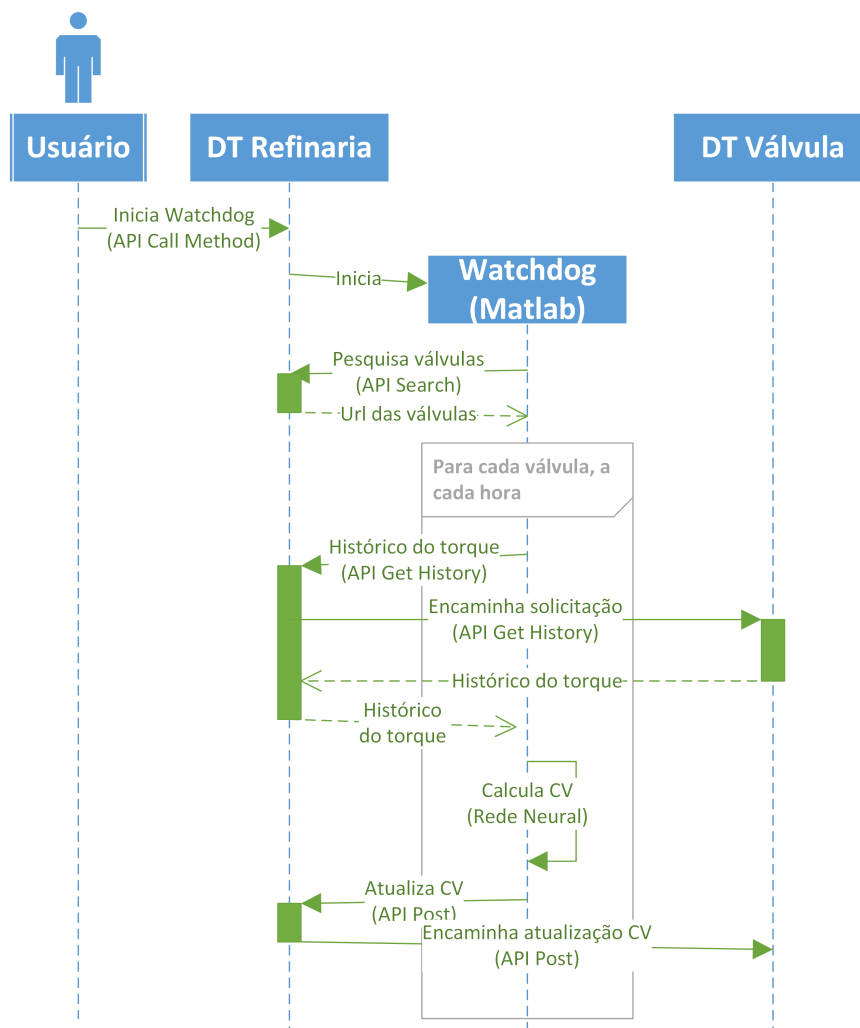


Figura 53: Diagrama de sequência da execução do watchdog para o digital twin agregado.

7.1.9 Cenário 4: Digital Twin com Fiware

Neste cenário, usou-se o modelo da válvula em AutomomationML ilustrado na Figura 54. A proposta deste cenário foi testar a troca de dados usando o Fiware, um Middleware de Internet das Coisas, para possibilitar a troca de dados e os serviços usados com o Web Services. Os atributos a serem monitorados foram extraídos do modelo da figura, sendo eles: (1) posição, (2) tensão e (3) temperatura. Estes atributos são enviados para o Fiware,

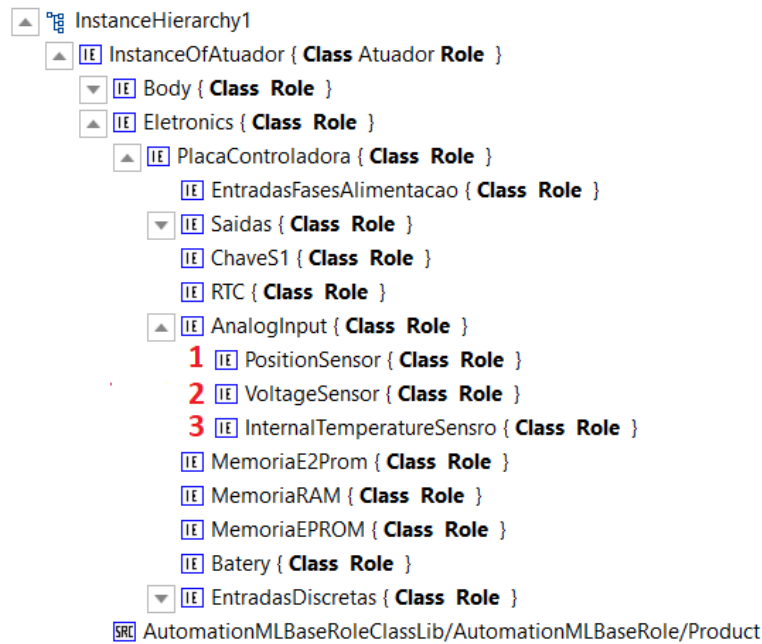


Figura 54: Modelo de atuador da válvula em AutomationML.

permitindo que outros sistemas possam ter acesso a eles. A parte do código em XML onde estes atributos estão representados, é a seguinte:

```
<InternalElement Name="AnalogInput" ID="33e5515a-339f-4432-9831-
  ↪ e0712aa48afd">
  <InternalElement Name="PositionSensor" ID="a1f5fc28-9852-4581-
    ↪ b16e-d213167f233a">
    <Attribute Name="position" AttributeDataType="xs:float" />
  </InternalElement>
  <InternalElement Name="VoltageSensor" ID="d9405f22-019b-4dcf
    ↪ -8623-f39f009e5d44">
    <Attribute Name="voltage" AttributeDataType="xs:float" />
  </InternalElement>
  <InternalElement Name="InternalTemperatureSensor" ID="339997e0
    ↪ -4099-454c-9486-1a485023158e">
    <Attribute Name="temperature" AttributeDataType="xs:float" /
      ↪ >
  </InternalElement>
</InternalElement>
```

A Figura 55 ilustra o esquema usado neste cenário. O item (1) é o componente físico, neste caso, a válvula. O item (2) ilustra a criação do modelo deste atuador no AutomationML, destacando principalmente os atributos que queremos monitorar. Em (3) mostra que os atributos selecionados são enviados para o middleware, permitindo que sejam vi-

sualizados em outros sistemas. O item (4) é o middleware escolhido, o Fiware e, em (5) tem-se os sistemas e plataformas externas que podem acessar os dados.

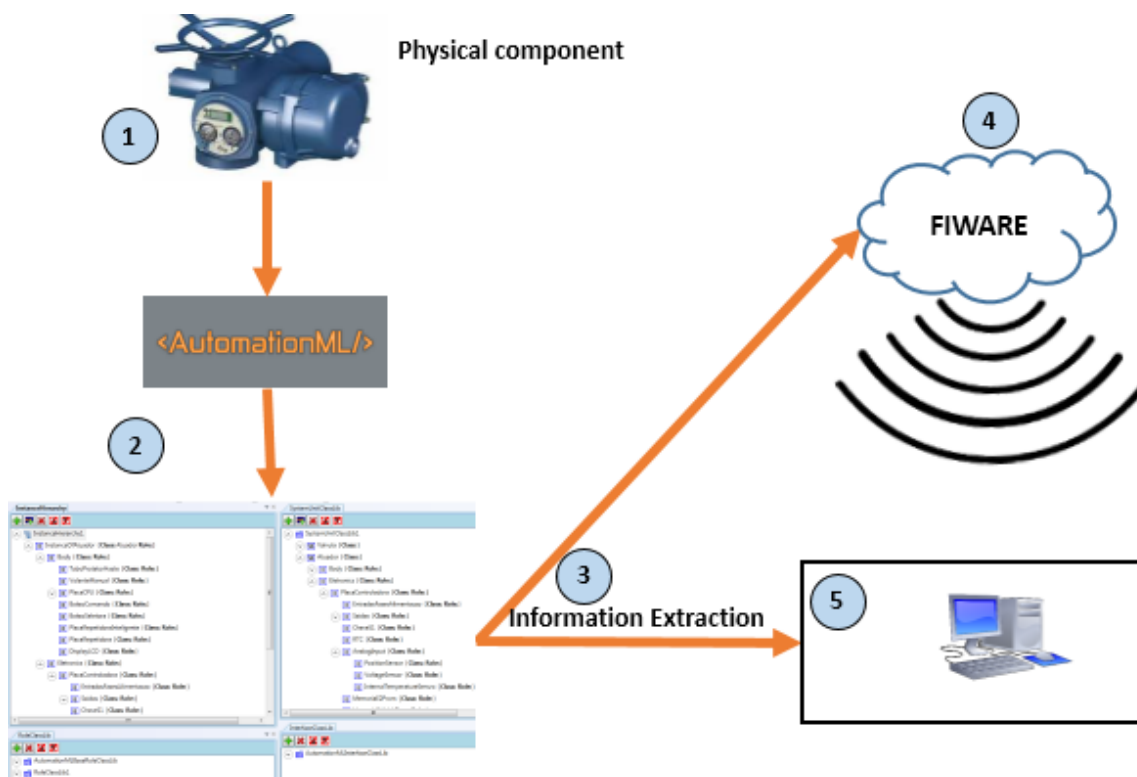


Figura 55: Esquema do funcionamento do estudo de caso da válvula.

A Figura 56 mostra os resultados obtidos. Os atributos foram extraídos do modelo em AutomationML e cadastrados no Fiware. De um lado, foi simulado a partir de um terminal, o envio de dados correspondentes aos atributos, como se este terminal fosse a válvula enviando dados de seus sensores. Do outro lado, foi requisitada uma consulta dos dados no middleware, que foi executado com sucesso.

7.1.10 Discussão

A implantação do digital twin em quatro cenários diferentes para a mesma aplicação permitiu algumas observações:

- A modelagem e implantação do digital twin foi facilitada devido à reutilização dos componentes modelados nas SUCs, o que vai de encontro à proposta da metodolo-

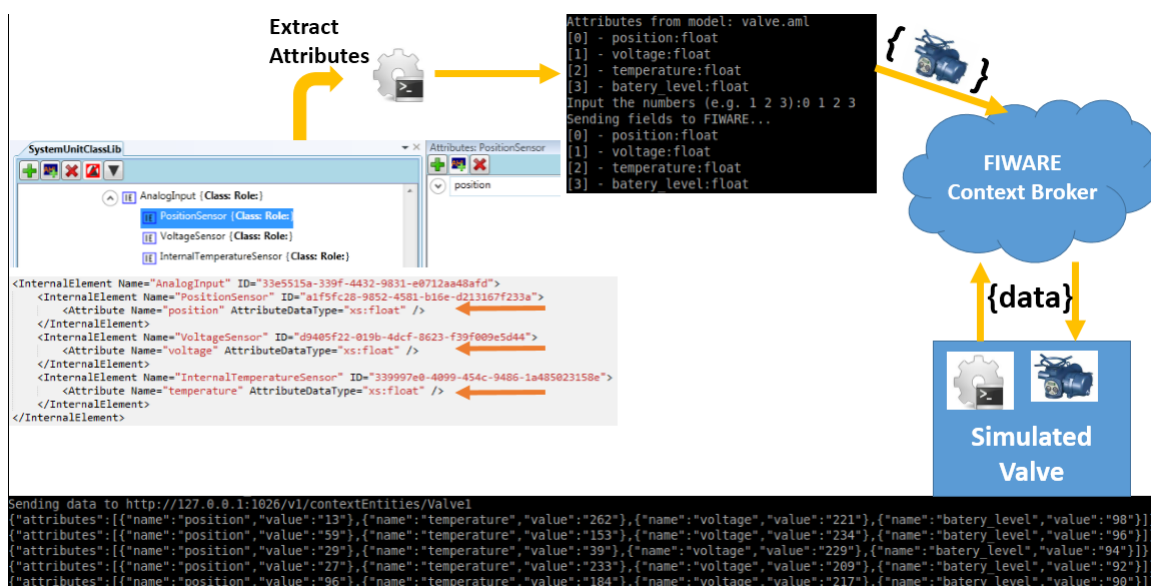


Figura 56: Exemplo do processo como um todo, tem-se o modelo em AML, os atributos selecionados para enviar dados para o Fiware e, por último, os dados sendo exibidos em outra plataforma.

gia.

- Não houve grandes alterações no modelo do digital twin e no procedimento de implantação entre os cenários 1, 2 e 3, o que indica que a metodologia se adapta a alterações da topologia.
- De um ponto de vista da aplicação externa (acesso via API), não houve diferença dos resultados entre os cenários 1, 2 e 3. Ou seja, a topologia interna do digital twin é indiferente do ponto de vista externo, permitindo a sua abstração.
- A troca inicial do uso de Web Services pelo middleware Fiware foi sucedida com sucesso, conforme visto no cenário 4, permitindo a flexibilidade do uso de outras plataformas para a implementação do modelo.

Estes resultados demonstram que a metodologia é flexível e facilmente aplicável a diferentes cenários, tanto de um ponto de vista da modelagem do digital twin quanto a subsequente implantação de forma automatizada.

7.2 Estudo de caso 2: Robô Edubot

Este estudo de caso faz uso de uma plataforma robótica para educação denominada “EduBOT” para auxiliar no ensino de mecatrônica, englobando componentes de hardware (mecânica e eletrônica) e software (programação). A proposta desta plataforma tem como meta o baixo custo e a utilização da metareciclagem para incentivar a criatividade sustentável na era digital.

O robô proposto é móvel, autônomo e opera em ambientes desconhecidos (Figura 57). Juntamente com a descrição dos elementos mecânicos, eletromecânicos e computacionais do projeto encontram-se os esquemáticos para o funcionamento do protótipo. Um algoritmo faz a identificação de obstáculos para verificar se executa ou não manobras evasivas, evitando a colisão.

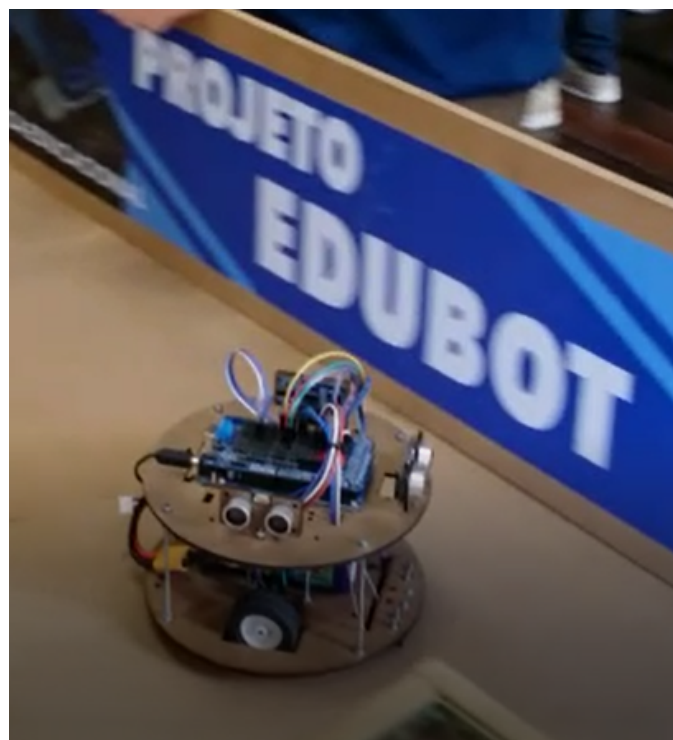


Figura 57: Robô Edubot.

7.2.1 Objetivo

Neste estudo de caso pretende-se criar um digital twin para o robô com as seguintes funcionalidades:

1. Comunicação via protocolo Modbus com o robô, recebendo informações da sua posição e nível da bateria.
2. Manutenção de um log (histórico) de operação do robô, permitindo acesso a estes dados quando for requisitado.
3. Disparo de alarme de bateria baixa quando o nível cai para menos do que 15%.
4. Navegação pelos dados do modelo do robô através de um navegador web.
5. Visualização da posição do robô, juntamente com seu histórico, através de um navegador web em qualquer lugar do mundo.
6. Acionamento de um simulador de comunicação Modbus que envia dados aleatórios da posição do robô.
7. Acionamento de um simulador de navegação do robô, que simula seu funcionamento em um mundo virtual.

Para isso, é utilizada a metodologia proposta nos capítulos anteriores, sendo o resultado apresentado nas subseções a seguir.

7.2.2 Modelagem do robô

A Figura 58 mostra o modelo em AutomationML do robô. Destaca-se que os componentes contém informações semânticas modeladas através do uso de classes e *roles*. Cada componente possui também propriedades (não mostradas nas figuras).

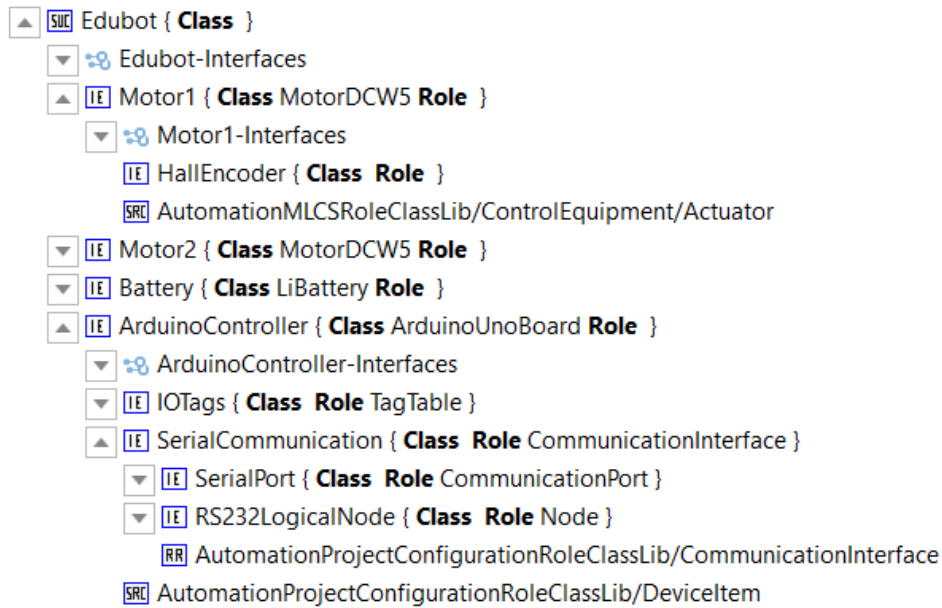


Figura 58: Modelo em AutomationML do robô Edubot.

Foi utilizado também um simulador do robô, que simula a sua navegação em um labirinto virtual conforme mostrado na Figura 59. O simulador contendo o ambiente virtual e a lógica do robô foi implementado utilizando-se a ferramenta Blender, a qual é usada para desenho de objetos gráficos, e é capaz de enviar a coordenadas de posição do robô no labirinto via protocolo TCP, UDP ou HTTP.

7.2.3 Modelagem do digital twin em AML

A Figura 60 mostra o modelo em AutomationML do digital twin do robô. A numeração em vermelho a esquerda de cada um dos componentes indica a respectiva funcionalidade do digital twin que o componente implementa conforme a Seção anterior.

7.2.4 Configuração e Implantação

As principais configuração realizadas foram as seguintes:

- Ligação, através da adição de *links* no editor do AutomationML, entre as interfaces dos *holding registers* do servidor modbus e os atributos do modelo do robô. Isto

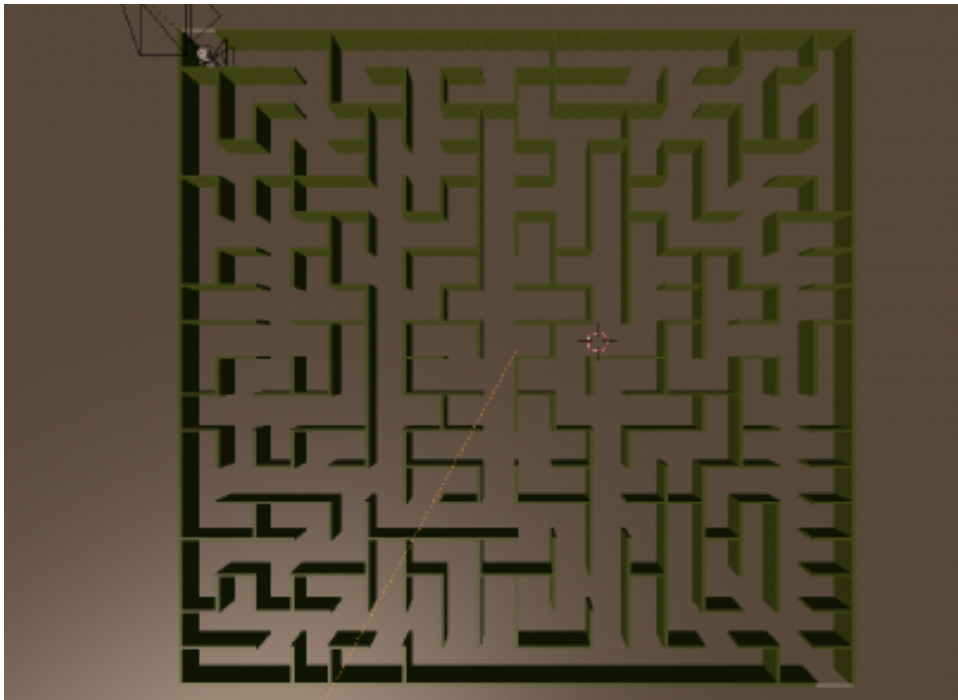


Figura 59: Labirinto Virtual para a simulação de envio de dados do robô.

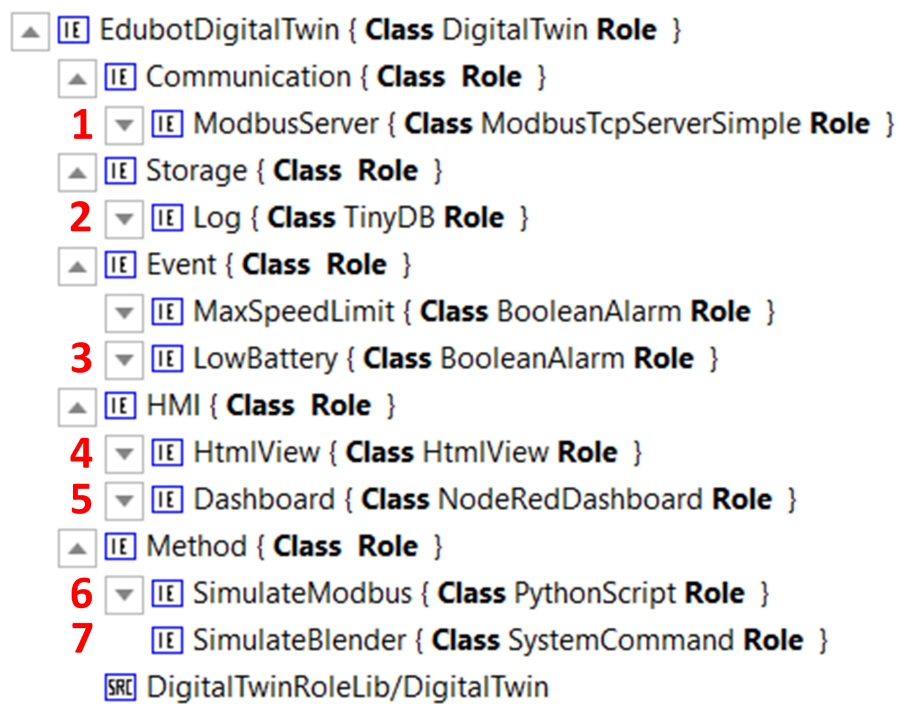


Figura 60: Modelo em AutomationML do digital twin do robô Edubot.

indica para o digital twin qual atributo será recebido em cada um dos endereços modbus.

- Ligação entre o alarme de bateria baixa com o atributo *capacity* da bateria, assim como a edição da expressão $capacity < 15$ no campo da expressão do alarme.
- Criação da *dashboard* na ferramenta Node-red e a importação via arquivo em formato JSON no componente da dashboard.
- Especificação do comando que executa o simulador do labirinto.

Uma vez terminada a configuração do modelo do digital twin, o arquivo AML é passado para o software de implantação automatizada que se encarrega da criação e execução de forma automática de todos os componentes.

7.2.5 Resultados e discussão

Durante a execução do digital twin pode-se observar que todos os componentes comportaram-se como o esperado, com exceção do simulador do labirinto, que apresentou uma taxa de atualização muito baixa (cerca de 1 fps), sendo que tal comportamento não foi notado quando o simulador é executado de forma isolada, sem enviar os dados para o digital twin. A causa averiguada para tal lentidão é um gargalo na comunicação entre digital twin e simulador. A comunicação feita via HTTP, onde o simulador manda as informações de posição do robô via mensagens POST para a API do digital twin, mostrou-se incapaz de tratar dados em tempo de execução. Este problema pode ser solucionado implementando-se o protocolo Modbus no simulador ou uma comunicação direta via UDP no digital twin, uma vez que o simulador pode mandar os dados via socket TCP ou UDP.

Pode-se ressaltar que os componentes utilizados neste estudo de caso foram em grande maioria os mesmos do estudo de caso anterior, comprovando a flexibilidade da metodo-

logia e possibilidade de reutilização de componentes.

8 CONCLUSÃO

Foi proposta uma arquitetura de referência para o digital twin, seguida de um modelo genérico e flexível que atende as necessidades da arquitetura. Destacou-se a importância de se ter um modelo, tanto do dispositivo quanto do próprio digital twin, que passam a ter papel central na arquitetura. Foram levantadas as diferentes topologias que os componentes de um sistema, como digital twin, pode apresentar, o que originou uma proposta para categorização do digital twin baseado nesta topologia. Cinco categorias foram descritas: desconectado, embarcado, conectado, multi-dispositivo e agregado.

Depois, foi apresentada uma metodologia para modelagem do digital twin usando AutomationML que segue a arquitetura e modelo de referência proposto anteriormente. As funcionalidades da linguagem AutomationML possibilitaram a modelagem em diferentes níveis de abstração, o que permite a agregação de valor semântico ao modelo e a generalização de conceitos. Foram criadas bibliotecas para *role classes*, *system unit classes* e *interfaces*, permitindo a reutilização dos componentes para a modelagem de novos digital twins de forma eficiente. Foi demonstrado que a metodologia proposta pode ser usada para modelagem de qualquer uma das topologias identificadas anteriormente, o que afirma que a metodologia é flexível e genérica.

Depois, um software que automatiza a implantação do digital twin baseado em modelo

foi proposto. Foi dada uma visão geral da sua lógica de funcionamento e das ferramentas utilizadas para a implementação de seus componentes, onde destaca-se a implementação das SUCs, que agregam as funcionalidades ao digital twin. A implementação da API do digital twin foi descrita, permitindo a padronização das operações de leitura e alteração dos dados via solicitações HTTP. A agilidade na implantação de um sistema é crucial no contexto da Indústria 4.0, sendo que a automatização deste processo pode permitir grandes ganhos de eficiência, principalmente na implantação de sistemas complexos e que necessitem constantemente ser reconfigurados para se adaptar a novas exigências do mercado.

Quanto a limitação no uso da metodologia destaca-se a impossibilidade de comunicação em tempo de execução no envio de dados entre o digital twin e o simulador para o robô. Quando testou-se a comunicação via HTTP onde o simulador enviava as informações de posição do robô via mensagens POST para a API do digital twin, a execução mostrou-se como não esperado.

Outra limitação encontrada foi no uso de métodos: a flexibilidade do AutomationML permite que qualquer método seja executado. Como um método pode ser um programa executável, isto pode ser considerado um risco pelo fato de programas maliciosos correrem o risco de ser executados.

Apesar das limitações apontadas em relação à metodologia, ficou constatado que nenhuma destas limitações inviabilizou a validação do modelo e da arquitetura e a verificação dos conceitos apresentados.

A metodologia proposta permite criar um modelo digital desde um simples produto manufaturado, passando pelos equipamentos industriais e chegando no nível de digitalizar a fábrica inteira. O sistema permite que o digital twin esteja sempre sendo atualizado

com dados dos sensores no equipamento físico, permite que informações sejam extraídas destes dados, que algoritmos de predição de falhas sejam incorporados no modelo.

Mesmo os trabalhos relacionados abordando o uso de arquiteturas e modelos para o digital twin, não são encontrados estudos detalhados para a implementação de um sistema como o proposto neste trabalho. Estes aspectos determinaram o desafio científico desta proposta.

Por fim, embora o uso do digital twin se estenda por todo o ciclo de vida, com a proposta desta Tese, dispositivos físicos já em fase de uso podem ter seu digital twin construído e começar a monitorar o restante de sua vida.

8.1 Principais contribuições

As principais contribuições obtidas com o desenvolvimento desta Tese atendem ao principal objetivo da mesma: fornecer uma metodologia de modelagem e uma arquitetura de referência para o digital twin no contexto de sistemas ciber físicos. Sendo estas contribuições listadas a seguir:

- Modelo genérico e flexível do digital twin que incorpora as características da arquitetura de um sistema para o compartilhamento de dados através da nuvem, descrito em AutomationML;
- Modelagem do dispositivo físico abrangente podendo conter diversas informações relevantes do dispositivo físico, tais como especificações do seu projeto construtivo, propriedades técnicas, identificação única, descrição dos seus componentes, estado atual de suas propriedades dinâmicas, especificação de operação, etc;
- Definição de serviços para digital twin que permitam o compartilhamento dos dados

e acesso às funcionalidades providas pelo digital twin;

- Uso de armazenamento de dados para manter o histórico de operação do dispositivo físico e do digital twin.
- Gerenciamento e disparo de eventos e possível notificação da ocorrência de tais eventos para outros componentes do sistema, aplicações ou usuários;
- Implementação de controle de acessos para definir regras específicas de controle de acesso aos recursos do digital twin, incluindo informações e funcionalidades
- Implementação de interfaces que realizam a comunicação com o dispositivo físico, sendo assim responsável pela atualização das propriedades do digital twin quando o dispositivo físico sofrer alguma alteração no seu estado e vice-versa;
- Uma nomenclatura, tanto para as topologias observadas atualmente na literatura, quanto para as topologias suportadas pela arquitetura e modelo de referência, visto que um sistema com digital twin pode apresentar diferentes formas de organização.

8.2 Trabalhos Futuros

O trabalho realizado nesta Tese serve como um trabalho inicial para o desenvolvimento do digital twin. Com isso, e mediante o potencial do digital twin observado na literatura, os seguintes trabalhos futuros podem ser implementados:

- Novas interfaces podem ser implementadas, como por exemplo, sistemas de realidade aumentada;
- Algumas referências alegam que o digital twin pode ter inteligência, o que é viável a partir deste trabalho;

- Uma das mais citadas aplicações do digital twin encontrada na literatura, é referente ao uso do digital twin para simulações. O modelo em AutomationML permite que modelos de simulações possam ser incorporados;
- Um estudo sobre casos de topologias mistas pode ser realizado futuramente, que exibem as características de duas ou mais topologias (por exemplo, um digital twin é agregado e conectado ao mesmo tempo);

REFERÊNCIAS

- ABRAMOVICI, M. Future Trends in Product Lifecycle Management (PLM). In: CIRP DESIGN CONFERENCE OF THE FUTURE OF PRODUCT DEVELOPMENT, 17., 2007, Heidelberg, Germany. **Proceedings...** Berlin:Springer, 2007. p.665–674.
- ALAM, K. M.; SADDIK, A. E. C2PS: a digital twin architecture reference model for the cloud-based cyber-physical systems. **IEEE Access**, New York, NY, v.5, p.2050–2062, 2017.
- ANGRISH, A. et al. A flexible data schema and system architecture for the virtualization of manufacturing machines (VMM). **Journal of Manufacturing Systems**, [S.l.], v.45, p.236 – 247, 2017.
- ATZORI, L.; IERA, A.; MORABITO, G. The Internet of Things: a survey. **Computer Networks**, [S.l.], v.54, n.15, p.2787 – 2805, 2010.
- AUTOMATIONML. **AutomationML Site**. Disponível em: <<https://www.automationml.org>>. Acesso em: janeiro 2018.
- BAJAJ, M.; COLE, B.; ZWEMER, D. Architecture To Geometry-Integrating System Models With Mechanical Design. In: AMERICAN INSTITUTE OF AERONAUTICS AND ASTRONAUTICS SPACE FORUM, 2016, 2016, Long Beach, California. **Proceedings...** Washington:AIAA, 2016. p.5470.
- BARNES, M.; FINCH, E. **COLLADA-Digital Asset Schema Release 1.5. 0 specification**. Disponível em: <<https://www.khronos.org/collada/>>. Acesso em: fevereiro 2018.
- BAZILEVS, Y. et al. Isogeometric fatigue damage prediction in large-scale composite structures driven by dynamic sensor data. **Journal of Applied Mechanics**, [S.l.], v.82, n.9, 2015.
- BIELEFELDT, B.; HOCHHALTER, J.; HARTL, D. Computationally efficient analysis of SMA sensory particles embedded in complex aerostructures using a substructure approach. In: ASME 2015 CONFERENCE ON SMART MATERIALS, ADAPTIVE STRUCTURES AND INTELLIGENT SYSTEMS, 2015, Colorado Springs, Colorado, USA. **Proceedings...** New York:ASME, 2015.
- BORODULIN, K. et al. Towards Digital Twins Cloud Platform: microservices and computational workflows to rule a smart factory. In: INTERNATIONAL CONFERENCE ON UTILITY AND CLOUD COMPUTING, 10., 2017, Austin, Texas, USA. **Proceedings...** New York:ACM, 2017. p.209–210.

BOTTANI, E. et al. From the Cyber-Physical System to the Digital Twin: the process development for behaviour modelling of a cyber guided vehicle in m2m logic. In: SUMMER SCHOOL FRANCESCO TURCO–INDUSTRIAL SYSTEMS ENGINEERING, 22., 2017, Palermo, Italy. **Proceedings...** Rome:AIDI, 2017. p.1–7.

BRETTEL, M. et al. How Virtualization, Decentralization and Network Building Change the Manufacturing Landscape: an industry 4.0 perspective. **International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering**, Germany, v.8, n.1, p.37 – 44, 2014.

BROCK, D. L. The electronic product code (epc)– A Naming Scheme for Physical Objects. **MIT Auto-ID Center White Paper**, Cambridge, MA, EUA, 2001. Disponível em: <<http://cocoa.ethz.ch/media/documents/2014/06/archive/MIT-AUTOID-WH-002.pdf>>. Acesso em: maio 2017.

CAI, Y. et al. Sensor Data and Information Fusion to Construct Digital-twins Virtual Machine Tools for Cyber-physical Manufacturing. **Procedia Manufacturing**, Los Angeles, EUA, v.10, p.1031–1042, 2017.

CANEDO, A. Industrial IoT Lifecycle via Digital Twins. In: IEEE/ACM/IFIP INTERNATIONAL CONFERENCE ON HARDWARE/SOFTWARE CODESIGN AND SYSTEM SYNTHESIS, 11., 2016, New York, NY, USA. **Proceedings...** ACM, 2016. p.29:1–29:1. (CODES '16).

CHEN, M.; MAO, S.; LIU, Y. Big Data: a survey. **Mobile Networks and Applications**, Springer, v.19, n.2, p.171–209, Apr 2014.

CIAVOTTA, M. et al. A Microservice-based Middleware for the Digital Factory. **Procedia Manufacturing**, Modena, Italy, v.11, p.931–938, 2017.

COESTER. **Coester Soluções Inovadoras em Automação**. Disponível em: <<http://www.coester.com.br>>. Acesso em: fevereiro 2018.

DJURDJANOVIC, D.; LEE, J.; NI, J. Watchdog Agent—an infotonics-based prognostics approach for product performance degradation assessment and prediction. **Advanced Engineering Informatics**, [S.l.], v.17, n.3, p.109 – 125, 2003.

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

ESTEFAN, J. A. et al. Survey of model-based systems engineering (MBSE) methodologies. **Incose MBSE Focus Group**, [S.l.], v.25, n.8, p.1–12, 2007.

FERGUSON, S.; BENNETT, E.; IVASHCHENKO, A. Digital twin tackles design challenges. **World Pumps**, New York:Elsevier, v.2017, n.4, p.26 – 28, 2017.

FRIEDENTHAL, S.; MOORE, A.; STEINER, R. **A practical guide to SysML: the systems modeling language**. [S.l.]: Morgan Kaufmann, 2014.

GABOR, T. et al. A Simulation-Based Architecture for Smart Cyber-Physical Systems. In: AUTONOMIC COMPUTING (ICAC), 2016 IEEE INTERNATIONAL CONFERENCE ON., 2016, Wurzburg, Germany. **Proceedings...** New York:IEEE, 2016. p.374–379.

GANDOMI, A.; HAIDER, M. Beyond the hype: big data concepts, methods, and analytics. **International Journal of Information Management**, [S.l.], v.35, n.2, p.137 – 144, 2015.

GANTZ, J.; REINSEL, D. Extracting value from chaos. **IDC iView**, [S.l.], v.1142, n.2011, p.1–12, 2011.

GIANNI, D.; D'AMBROGIO, A.; TOLK, A. **Modeling and Simulation-Based Systems Engineering Handbook**. Boca Raton, Florida: CRC Press, 2014.

GLAESSGEN, E. H.; STARGEL, D. The Digital Twin paradigm for future NASA and US Air Force vehicles. In: STRUCT. DYN. MATER. CONF. SPECIAL SESSION: DIGITAL TWIN, 53., 2012, Honolulu, HI, US. **Proceedings...** Washington:AIAA, 2012. p.1–14.

GOCKEL, B. T. et al. Challenges with Structural Life Forecasting using Realistic Mission Profiles. In: AIAA/ASME/ASCE/AHS/ASC STRUCTURES, STRUCTURAL DYNAMICS AND MATERIALS CONFERENCE, 53., 2012, Honolulu, HI, US. **Proceedings...** Washington:AIAA, 2012. p.1813.

GORECKY, D. et al. Human-machine-interaction in the industry 4.0 era. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL INFORMATICS (INDIN), 12., 2014, Porto Alegre, Brasil. **Proceedings...** New York:IEEE, 2014. p.289–294.

GRIEVES, M. **Product Lifecycle Management: driving the next generation of lean thinking: driving the next generation of lean thinking: driving the next generation of lean thinking**. Chicago, IL, USA: McGraw Hill Professional, 2005.

GRIEVES, M. Digital twin: manufacturing excellence through virtual factory replication. **White Paper**, online, 2014. Disponível em: <http://innovate.fit.edu/plm/documents/doc_mgr/912/1411.0_Digital_Twin_White_Paper_Dr_Griev.pdf>. Acesso em: maio 2017.

GUBBI, J. et al. Internet of Things (IoT): a vision, architectural elements, and future directions. **Future generation computer systems**, [S.l.], v.29, n.7, p.1645–1660, 2013.

HAGGE, N.; WAGNER, B. Implementation alternatives for the OMAC state machines using IEC 61499. In: IEEE INTERNATIONAL CONFERENCE ON EMERGING TECHNOLOGIES AND FACTORY AUTOMATION, 2008., 2008, Hamburg, Germany. **Proceedings...** New York:IEEE, 2008. p.215–220.

HENßEN, R.; SCHLEIPEN, M. Interoperability between OPC UA and AutomationML. **Procedia CIRP**, Stuttgart, Germany, v.25, p.297 – 304, 2014. Part of special issue: 8th International Conference on Digital Enterprise Technology - DET 2014 Disruptive Innovation in Manufacturing Engineering towards the 4th Industrial Revolution.

HERMANN, M.; PENTEK, T.; OTTO, B. Design principles for industrie 4.0 scenarios. In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES (HICSS), 49., 2016, Koloa, HI, USA. **Proceedings...** New York:IEEE, 2016. p.3928–3937.

HIRMER, P. et al. Automating the Provisioning and Configuration of Devices in the Internet of Things. **Complex Systems Informatics and Modeling Quarterly (CSIMQ)**, [S.l.], v.9, p.28–43, 2016.

HRIBERNIK, K.; WUEST, T.; THOBEN, K.-D. Towards Product Avatars Representing Middle-of-Life Information for Improving Design, Development and Manufacturing Processes. In: DIGITAL PRODUCT AND PROCESS DEVELOPMENT SYSTEMS, 2013., 2013, Berlin, Heidelberg. **Proceedings...** Springer Berlin Heidelberg, 2013. p.85–96.

IGLESIAS, D. et al. Digital twin applications for the JET divertor. **Fusion Engineering and Design**, [S.l.], v.125, p.71 – 76, 2017.

JAZDI, N. Cyber physical systems in the context of Industry 4.0. In: AUTOMATION, QUALITY AND TESTING, ROBOTICS, 2014 IEEE INTERNATIONAL CONFERENCE ON, 2014, Cluj-Napoca, Romania. **Proceedings...** New York:IEEE, 2014. p.1–4.

KAGERMANN, H. et al. **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.** Germany: acatech – National Academy of Science and Engineering, 2013. Disponível em: <<https://www.acatech.de/publikationen/>>. Acesso em: novembro 2016.

KIRITSIS, D. Closed-loop PLM for intelligent products in the era of the Internet of things. **Computer-Aided Design**, [S.l.], v.43, n.5, p.479 – 501, 2011. Emerging Industry Needs for Frameworks and Technologies for Exchanging and Sharing Product Lifecycle Knowledge.

KNAPP, G. et al. Building blocks for a digital twin of additive manufacturing. **Acta Materialia**, [S.l.], v.135, p.390 – 399, 2017.

KONSTANTINOV, S. et al. The Cyber-physical E-machine Manufacturing System: virtual engineering for complete lifecycle support. **Procedia CIRP**, Taichung, Taiwan, v.63, p.119 – 124, 2017. Manufacturing Systems 4.0 – Proceedings of the 50th CIRP Conference on Manufacturing Systems.

KRAFT, E. M. The US Air Force Digital Thread/Digital Twin–Life Cycle Integration and Use of Computational and Experimental Knowledge. In: AIAA AEROSPACE SCIENCES MEETING, AIAA SCITECH, 54., 2016, San Diego, California, USA. **Proceedings...** Washington:AIAA, 2016. p.0897.

LEE, E. et al. Cyber physical systems: design challenges. In: IEEE INTERNATIONAL SYMPOSIUM ON OBJECT ORIENTED REAL-TIME DISTRIBUTED COMPUTING (ISORC), 11., 2008, Orlando, FL, USA. **Proceedings...** New York:IEEE, 2008. p.363–369.

LEE, J. Smart factory systems. **Informatik-Spektrum**, Berlin Heidelberg, GE, v.38, n.3, p.230–235, 2015.

LEE, J.; BAGHERI, B.; KAO, H.-A. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. **Manufacturing Letters**, [S.l.], v.3, p.18–23, 2015.

LEE, J. et al. Recent advances and trends in predictive manufacturing systems in big data environment. **Manufacturing Letters**, [S.l.], v.1, n.1, p.38–41, 2013.

LEE, J.; JIN, C.; LIU, Z. Predictive Big Data Analytics and Cyber Physical Systems for TES Systems. In: REDDING, L.; ROY, R.; SHAW, A. (Ed.). **Advances in Through-life Engineering Services**. Cham: Springer International Publishing, 2017. p.97–112.

LI, C. et al. A dynamic Bayesian network approach for digital twin. In: AIAA NON-DETERMINISTIC APPROACHES CONFERENCE, 19., 2017, Grapevine, Texas, USA. **Proceedings...** Washington:AIAA, 2017. p.1566.

MAGARGLE, R. et al. A Simulation-Based Digital Twin for Model-Driven Health Monitoring and Predictive Maintenance of an Automotive Braking System. In: INTERNATIONAL MODELICA CONFERENCE, 12., 2017, Prague, Czech Republic. **Proceedings...** Modelica Association and Linköping University Electronic Press, 2017. n.132, p.35–46.

MAJUMDAR, P. K.; HAIDER, M.; REIFSNIDER, K. Multi-Physics Response of Structural Composites and Framework for Modeling using Material Geometry. In: AIAA/ASME/ASCE/AHS/ASC STRUCTURES, STRUCTURAL DYNAMICS, AND MATERIALS CONFERENCE, 54., 2013, San Diego, California, USA. **Proceedings...** Washington:AIAA, 2013. p.1577.

MANYIKA, J. et al. **Big data**: the next frontier for innovation, competition, and productivity. [S.l.]: McKinsey Global Institute, 2011.

MAYER-SCHÖNBERGER, V.; CUKIER, K. **Big Data**: a revolution that will transform how we live, work, and think. Boston, MA: Houghton Mifflin Harcourt, 2013. (An Eamon Dolan book).

MORA, D. et al. Service-oriented architecture approach for industrial “System of Systems”: state-of-the-art for energy management. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL INFORMATICS (INDIN), 10., 2012, Beijing, China. **Proceedings...** New York:IEEE, 2012. p.1246–1251.

MORENO, A. et al. Virtualisation process of a sheet metal punching machine within the Industry 4.0 vision. **International Journal on Interactive Design and Manufacturing (IJIDeM)**, [S.l.], v.11, n.2, p.365–373, 2017.

NEGRI, E.; FUMAGALLI, L.; MACCHI, M. A Review of the Roles of Digital Twin in CPS-based Production Systems. **Procedia Manufacturing**, Modena, Italy, v.11, p.939 – 948, 2017. 27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017.

PAUTASSO, C.; ZIMMERMANN, O.; LEYMANN, F. Restful web services vs. big’ web services: making the right architectural decision. In: WORLD WIDE WEB, 17., 2008, Beijing, China. **Proceedings...** New York:ACM, 2008. p.805–814.

PEREIRA, C. E. Sistemas de automação industrial em tempo real. In: IV ESCOLA REGIONAL DE INFORMATICA, 1996, Londrina, Itajaí, Canoas. **Anais...** SBC, 1996. p.195–203.

PLCOPEN, X. **PLCopen format**. Disponível em: <<http://www.plcopen.org/>>. Acesso em: fevereiro 2018.

QI, Q.; TAO, F. Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 degree comparison. **IEEE Access**, New York, USA, v.6, p.3585–3593, 2018.

REID, J. B.; RHODES, D. H. Digital System Models: an investigation of the non-technical challenges and research needs. In: CONFERENCE ON SYSTEMS ENGINEERING RESEARCH, 2016, 2016, Alabama. **Proceedings...** Boston:MIT Press, 2016.

REIFSNIDER, K.; MAJUMDAR, P. Multiphysics stimulated simulation digital twin methods for fleet management. In: AIAA/ASME/ASCE/AHS/ASC STRUCTURES, STRUCTURAL DYNAMICS, AND MATERIALS CONFERENCE, 54., 2013, San Diego, California, USA. **Proceedings...** Washington:AIAA, 2013. p.1578.

RIOS, J. et al. Product Avatar as Digital Counterpart of a Physical Individual Product: literature review and implications in an aircraft. In: ISPE INC. INTERNATIONAL CONFERENCE ON CONCURRENT ENGINEERING, 22., 2015, Netherlands. **Proceedings...** Amsterdam:IOS Press, 2015. p.657–666.

RIOS, J. et al. Framework to support the aircraft digital counterpart concept with an industrial design view. **International Journal of Agile Systems and Management**, Inderscience Publishers (IEL), v.9, n.3, p.212–231, 2016.

ROARKE, E. **Device Smart**. Disponível em: <<https://devicesmart.wordpress.com/tag/cyber-physical-systems/>>. Acesso em: fevereiro 2018.

ROSEN, R. et al. About The Importance of Autonomy and Digital Twins for the Future of Manufacturing. **IFAC-PapersOnLine**, Ottawa, Canada, v.48, n.3, p.567 – 572, 2015. Part of special issue: 15th IFAC Symposium on Information Control Problems in Manufacturing.

RÜSSMANN, M. et al. **Industry 4.0**: the future of productivity and growth in manufacturing industries. Disponível em: <<https://i40.de/en/2590/>>. Acesso em: fevereiro 2018.

SANTAELLA, L. et al. Desvelando a internet das coisas. **Revista GEMInIS**, Florianopolis, SC, v.4, n.2, p.19–32, 2013.

SCHACH, S. R. **Object-oriented and classical software engineering**. New York: McGraw-Hill New York, 2007. v.6.

SCHLEICH, B. et al. Shaping the digital twin for design and production engineering. **CIRP Annals**, New York, v.66, n.1, p.141 – 144, 2017.

SCHLEIPEN, M.; DRATH, R.; SAUER, O. The system-independent data exchange format CAEX for supporting an automatic configuration of a production monitoring and control system. In: **IEEE INTERNATIONAL SYMPOSIUM ON INDUSTRIAL ELECTRONICS**, 2008., 2008, Cambridge, UK. **Proceedings...** New York:IEEE, 2008. p.1786–1791.

SCHLUSE, M.; ATORF, L.; ROSSMANN, J. Experimentable digital twins for model-based systems engineering and simulation-based development. In: **SYSTEMS CONFERENCE (SYSCON), 2017 ANNUAL IEEE INTERNATIONAL**, 2017, Montreal, Quebec, Canada. **Proceedings...** New York:IEEE, 2017. p.1–8.

SCHLUSE, M. et al. Experimentable Digital Twins - Streamlining Simulation-based Systems Engineering for Industry 4.0. **IEEE Transactions on Industrial Informatics**, New York, v.PP, n.99, p.1–1, 2018.

SCHLUSE, M.; ROSSMANN, J. From simulation to experimentable digital twins: simulation-based development and operation of complex technical systems. In: **SYSTEMS ENGINEERING (ISSE), 2016 IEEE INTERNATIONAL SYMPOSIUM ON**, 2016, Edinburgh, Scotland. **Proceedings...** New York:IEEE, 2016. p.1–6.

SCHROEDER, G. N. et al. Digital twin data modeling with automationML and a communication methodology for data exchange. **IFAC-PapersOnLine**, [S.l.], v.49, n.30, p.12–17, 2016.

SHA, L. et al. Cyber-Physical Systems: a new frontier. In: **IEEE INTERNATIONAL CONFERENCE ON SENSOR NETWORKS, UBIQUITOUS, AND TRUSTWORTHY COMPUTING (SUTC 2008)**, 2008., 2008, Taichung, Taiwan. **Anais...** New York:IEEE, 2008. p.1–9.

SHAFTO, M. et al. Draft modeling, simulation, information technology & processing roadmap. **National Aeronautics and Space Administration**, Washington, DC, The United States of America, 2010.

SHAFTO, M. et al. Modeling, simulation, information technology & processing roadmap: technology area 11. **National Aeronautics and Space Administration**, Washington, DC, The United States of America, 2012.

SHI, X. Sharing Service Semantics using SOAP-Based and REST Web Services. **IT Professional**, New York:IEEE Computer Society, v.8, p.18–24, 03 2006.

SIERLA, S. et al. Automatic assembly planning based on digital product descriptions. **Computers in Industry**, [S.l.], v.97, p.34 – 46, 2018.

SNELL, J.; TIDWELL, D.; KULCHENKO, P. **Programming Web services with SOAP**. California, EUA: O'Reilly Media, Inc., 2001.

SODERBERG, R. et al. Toward a Digital Twin for real-time geometry assurance in individualized production. **CIRP Annals in Manufacturing Technology**, New York, v.66, p.137–140, 2017.

SRIVASTAVA, B.; KOEHLER, J. Web service composition-current solutions and open problems. In: **ICAPS 2003 WORKSHOP ON PLANNING FOR WEB SERVICES**, 2003, Trento, Italy. **Proceedings...** [S.l.: s.n.], 2003. v.35, p.28–35.

TAMMARO, A. et al. Extending industrial digital twins with optical object tracking. In: CONGRESO ESPAÑOL DE INFORMÁTICA GRÁFICA, 27., 2017, Sevilla. **Proceedings...** [S.l.: s.n.], 2017. p.23–26.

TAO, F. et al. Digital twin-driven product design, manufacturing and service with big data. **The International Journal of Advanced Manufacturing Technology**, [S.l.], p.1–14, 2017.

TAVARES, P. et al. Flexible Work Cell Simulator Using Digital Twin Methodology for Highly Complex Systems in Industry 4.0. In: ROBOT 2017: IBERIAN ROBOTICS CONFERENCE, 3, 2017, Cham. **Proceedings...** Berlin:Springer, 2017. p.541–552.

TUEGEL, E. The airframe digital twin: some challenges to realization. In: AIAA/ASME/ASCE/AHS/ASC STRUCTURES, STRUCTURAL DYNAMICS AND MATERIALS CONFERENCE, 53., 2012, Honolulu, HI, US. **Proceedings...** Washington:AIAA, 2012. p.1812.

UDDIN, M. F.; GUPTA, N. et al. Seven V's of Big Data understanding Big Data to extract value. In: ZONE 1 CONFERENCE OF THE AMERICAN SOCIETY FOR ENGINEERING EDUCATION, 2014., 2014, Bridgeport, Connecticut, USA. **Proceedings...** New York:IEEE, 2014. p.1–5.

UHLEMANN, T. H.-J.; LEHMANN, C.; STEINHILPER, R. The Digital Twin: realizing the cyber-physical production system for industry 4.0. **Procedia CIRP**, [S.l.], v.61, p.335–340, 2017.

UM, J.; WEYER, S.; QUINT, F. Plug-and-Simulate within Modular Assembly Line enabled by Digital Twins and the use of AutomationML. **IFAC-PapersOnLine**, [S.l.], v.50, n.1, p.15904–15909, 2017.

WANG, S. et al. Implementing smart factory of industrie 4.0: an outlook. **International Journal of Distributed Sensor Networks**, [S.l.], v.12, n.1, p.3159805, 2016.

WEYER, S. et al. Future Modeling and Simulation of CPS-based Factories: an example from the automotive industry. **IFAC-PapersOnLine**, [S.l.], v.49, n.31, p.97–102, 2016.

WÜBBEKE, J. et al. Made in China 2025. **Merics: Mercator Institute for China Studies**, Berlin, Germany, v.2, p.14–41, 2016.

YUN, S.; PARK, J.-H.; KIM, W.-T. Data-centric middleware based digital twin platform for dependable cyber-physical systems. In: INTERNATIONAL CONFERENCE ON UBIQUITOUS AND FUTURE NETWORKS (ICUFN), 9, 2017, Milan, Italy. **Proceedings...** New York:IEEE, 2017. p.922–926.

ZHANG, H. et al. A Digital Twin-Based Approach for Designing and Multi-Objective Optimization of Hollow Glass Production Line. **IEEE Access**, New York, NY, USA, v.5, p.26901–26911, 2017.

ZHUANG, C.; LIU, J.; XIONG, H. Digital twin-based smart production management and control framework for the complex product assembly shop-floor. **The International Journal of Advanced Manufacturing Technology**, [S.l.], p.1–15, 2018.

ZUEHLKE, D. SmartFactory—Towards a factory-of-things. **Annual Reviews in Control**, [S.l.], v.34, n.1, p.129–138, 2010.