UNIVERSIDADE BEIRA INTERIOR

A Semantic Interoperability Model Based on the IEEE 1451 Family of Standards Applied to the Industry 4.0

Helbert da Rocha

Tese para obtenção do Grau de Doutor em

Engenharia Eletrotécnica e de Computadores

(3º ciclo de estudos)

Orientador: Prof. Doutor António Eduardo Vitória do Espírito Santo

Junho de 2023

Provas públicas realizadas em 06 de junho de 2023

Composição do júri

Presidente

Doutor António João Marques Cardoso
Professor Catedrático da Universidade da Beira Interior

Vogais

Doutor Vincenzo Paciello
Professor Associado da Università degli Studi di Salerno

Doutor Bruno Jorge Ferreira Ribeiro
Professor Auxiliar da Universidade da Beira Interior

Doutor António Eduardo Vitória do Espírito Santo
Professor Auxiliar da Universidade da Beira Interior

Doutor Gil Manuel Magalhães de Andrade Gonçalves
Professor Auxiliar Universidade do Porto

Declaração de Integridade

Eu, Helbert da Rocha, que abaixo assino, estudante com o número de inscrição D2532 de/o Engenharia Eletrotécnica e de Computadores da Faculdade Engenharia, declaro ter desenvolvido o presente trabalho e elaborado o presente texto em total consonância com o **Código de Integridades da Universidade da Beira Interior**.

Mais concretamente afirmo não ter incorrido em qualquer das variedades de Fraude Académica, e que aqui declaro conhecer, que em particular atendi à exigida referenciação de frases, extratos, imagens e outras formas de trabalho intelectual, e assumindo assim na íntegra as responsabilidades da autoria.

Universidade da Beira Interior, Covilhã 19/06/2023

Assinado por: **Helbert da Rocha**
Num. de Identificação: TR:PT-58K56K751
Data: 2023.06.19 18:05:46 +0100

CHAVE MÓVEL

This thesis was developed at the Measurement and Instrumentation Laboratory (IML) in the University of Beira Interior and supported by the portuguese project INDTECH 4.0 – Novas tecnologias para fabricação, que tem como objetivo geral a conceção e desenvolvimento de tecnologias inovadoras no contexto da Indústria 4.0/Factories of the Future (FoF), under the number POCI-01-0247-FEDER-026653.

Cofinanciado por:

COMPETE 2020 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Europeu de Desenvolvimento Regional

Acknowledgement

I wish to thank my family, that is always by my side, supporting me in the remarkable moments of my life, especially my mother, Neura. My brothers Marcelo, Marcio, and Helder. My nephews, Maycon, Maria, Amanda, and Benjamin, contributed significantly to the person I became.

In loving memory of my father, Neuri, who passed away in 2022.

A special thanks to my advisor Professor Ph.D. António Eduardo Vitória do Espírito Santo for the guidelines and long hours of discussions that resulted in insights for the development and conclusion of this thesis.

I am very grateful to the people who support me during this time of my thesis, for the long hours of talking and their friendship, especially João Pereira, Rita Pinto, Filippo Milano, and Ana Garcez.

I want to thank all the professors who made it possible to get here by sharing knowledge throughout the process for personal and professional growth. A special thanks to Professor Ph.D. Tania Monteiro, and the Professor Ph.D. Reza Abrishambaf.

Thank all my few good friends and all the other people who contributed in some form to the development of this thesis.

Finally, to thank the project INDTECH 4.0 for the grant that enables the development of this thesis.

Sincerely, Helbert da Rocha

"It always seems impossible until it is done." Nelson Mandela

Resumo

A Internet das Coisas tem vindo a crescer recentemente. É um conceito que permite conectar bilhões de dispositivos inteligentes através da Internet em diferentes cenários. Uma área que está sendo desenvolvida dentro da Internet das Coisas é a automação industrial, que abrange a comunicação máquina com máquina no processo industrial de forma automática. Essa interligação, representa o conceito da Internet das Coisas Industrial. Dentro da Internet das Coisas Industrial está a desenvolver o conceito de Indústria 4.0 (I4.0). Isso representa a quarta revolução industrial que aborda o uso de tecnologias utilizadas na Internet para melhorar a eficiência da produção de serviços em fábricas inteligentes. A Indústria 4.0 é composta por uma combinação de objetos do mundo físico e do mundo da digital que oferece funcionalidade dedicada e flexibilidade dentro e fora de uma rede da Indústria 4.0.

O I4.0 é composto principalmente por Sistemas Ciberfísicos. Os Sistemas Ciberfísicos permitem a integração do mundo físico com seu representante no mundo digital, por meio do Gémeo Digital. Sistemas Ciberfísicos são responsáveis por realizar a aplicação inteligente da ligação cruzada, que opera de forma auto-organizada e descentralizada, utilizada por fábricas inteligentes para criação de valor. Uma área em que o Sistema Ciberfísicos pode ser implementado na produção manufatureira, isso representa o desenvolvimento do conceito Sistemas de Produção Ciberfísicos. Esse sistema é a implementação da Indústria 4.0 e Sistema Ciberfísicos na fabricação e produção. A cruzar todos os níveis desde a produção entre os elementos e subsistemas autónomos e cooperativos. Ele é responsável por conectar o espaço virtual com o mundo físico, permitindo que as fábricas inteligentes sejam mais inteligentes, resultando em condições de produção melhores e inteligentes, aumentando a produtividade, a eficiência da produção e a qualidade do produto. A grande questão é como conectar dispositivos inteligentes com diferentes normas e protocolos. Cerca de 40% dos benefícios da Internet das Coisas não podem ser alcançados sem interoperabilidade. Esta tese está focada em promover a interoperabilidade de dispositivos inteligentes (sensores e atuadores) dentro da Internet das Coisas Industrial no contexto da Indústria 4.0.

O IEEE 1451 é uma família de normas desenvolvidos para gerenciar transdutores. Esta norma alcança o nível sintático de interoperabilidade dentro de uma indústria 4.0. No entanto, a Indústria 4.0 requer um nível semântico de comunicação para não haver a trocar dados de forma ambígua. Uma nova camada semântica é proposta nesta tese permitindo que a família de normas IEEE 1451 seja um *framework* completo para comunicação dentro da Indústria 4.0. Permitindo fornecer uma interface de rede interoperável com utilizadores e aplicações para recolher e compartilhar os dados dentro de um ambiente industrial.

Palavras-chave

Família de normas IEEE 1451, Internet das Coisas na Indústria, Indústria 4.0, Interoperabilidade Semântica.

Abstract

The Internet of Things (IoT) has been growing recently. It is a concept for connecting billions of smart devices through the Internet in different scenarios. One area being developed inside the IoT in industrial automation, which covers Machine-to-Machine (M2M) and industrial communications with an automatic process, emerging the Industrial Internet of Things (IIoT) concept. Inside the IIoT is developing the concept of Industry 4.0 (I4.0). That represents the fourth industrial revolution and addresses the use of Internet technologies to improve the production efficiency of intelligent services in smart factories. I4.0 is composed of a combination of objects from the physical world and the digital world that offers dedicated functionality and flexibility inside and outside of an I4.0 network.

The I4.0 is composed mainly of Cyber-Physical Systems (CPS). The CPS is the integration of the physical world and its digital world, i.e., the Digital Twin (DT). It is responsible for realising the intelligent cross-link application, which operates in a self-organised and decentralised manner, used by smart factories for value creation. An area where the CPS can be implemented in manufacturing production is developing the Cyber-Physical Production System (CPPS) concept. CPPS is the implementation of Industry 4.0 and CPS in manufacturing and production, crossing all levels of production between the autonomous and cooperative elements and sub-systems. It is responsible for connecting the virtual space with the physical world, allowing the smart factories to be more intelligent, resulting in better and smart production conditions, increasing productivity, production efficiency, and product quality. The big issue is connecting smart devices with different standards and protocols. About 40% of the benefits of the IoT cannot be achieved without interoperability. This thesis is focused on promoting the interoperability of smart devices (sensors and actuators) inside the IIoT under the I4.0 context.

The IEEE 1451 is a family of standards developed to manage transducers. This standard reaches the syntactic level of interoperability inside Industry 4.0. However, Industry 4.0 requires a semantic level of communication not to exchange data ambiguously. A new semantic layer is proposed in this thesis allowing the IEEE 1451 standard to be a complete framework for communication inside the Industry 4.0 to provide an interoperable network interface with users and applications to collect and share the data from the industry field.

Keywords

IEEE 1451 family of standards, Industrial Internet of Things, Industry 4.0, Semantic Interoperability.

Resumo Alargado

Introdução

Esta secção apresenta o resumo alargado, em português, do trabalho de investigação da tese de doutoramento intitulada "A Semantic Interoperability Model Based on the IEEE 1451 Family of Standards Applied to the Industry 4.0". Esta tese centra-se no estudo e proposta de uma nova camada semântica para interoperabilidade no contexto da Indústria 4.0 baseada na família de normas IEEE 1451. Nesse resumo é descrito a estrutura da tese, o problema abordado e a principal contribuição deste estudo.

Enquadramento da Tese

Essa tese foi elabora como parte do projeto "INDTECH 4.0 – Novas tecnologias para fabricação inteligente", financiado pelo Programa Portugal 2020 (PT 2020) com o Programa Operacional Competitividade e Internacionalização (POCI) pelo Fundo Europeu de Desenvolvimento Regional (FEDER). O principal objetivo da tese é desenvolver e normalizar a comunicação de transdutores inteligentes para operar em ambiente industrial.

A interoperabilidade é essencial para a comunicação com dispositivos de diferentes fornecedores dentro da Internet das Coisas. Estimou-se que a interoperabilidade é responsável por 40% dos benefícios de conectar dispositivos dentro do mundo da Internet das Coisas [1]. Sendo a falta de interoperabilidade responsável pelo aumento do custo de desenvolvimento em diversas áreas, como a indústria. A Internet das Coisas voltada para o setor industrial recebe o conceito de Internet das Coisas Industrial. Dentro da Internet das Coisas Industrial centra-se no processo de fabricação chamado de Indústria 4.0 [2]. O I4.0 é composto pela parte física e sua representação digital. A parte física são os Sistemas Ciberfísicos e a contraparte digital é o Gémeo Digital.

Muitas normas podem ser utilizadas para criar a conexão entre a parte física e a contraparte digital. A interoperabilidade é um dos elementos essenciais para o sucesso da implementação de um Sistema Ciberfísico [3]. Dentro dos níveis de interoperabilidade, os níveis sintático e semântico são os mais empregados na indústria [4]–[6].

Esta tese tem como objetivo contribuir para a resolução do problema de interoperabilidade na Indústria 4.0 e na Internet das Coisas Industrial. Níveis de interoperabilidade podem ser alcançados durante a comunicação do nível do dispositivo para o nível semântico. A norma proposta para a indústria foi a família de normas IEEE 1451, com foco na aquisição e transmissão de dados de transdutores [7]. Esta norma atingiu o nível sintático de interoperabilidade em sua versão atual [8]. A interoperabilidade semântica foi proposta e implementada num vocabulário e ontologia baseados na norma família de normas IEEE 1451, transformando a norma num *framework* que alcança todos os níveis de comunicação necessários para a comunicação da Indústria 4.0 e da Internet das Coisas Industrial [9].

A família de normas IEEE 1451 permite o gerenciamento de dados com o transdutor para adquirir dados de sensores e enviar comandos para atuadores. Como a norma prevê redes de sensores como duas partes diferentes, de acordo com a norma, uma rede de sensores inclui um Processador de Aplicações Compatível com Rede (NCAP), que é um *gateway* entre o transdutor real (sensor e atuador) instalado no Módulo de Interface do Transdutor (TIM). Dentro do TIM é armazenada a Folha de Dados Eletrônicos do Transdutor (TEDS). Um TEDS contém informações sobre o TIM e o canal do transdutor utilizado para aceder os sensores ou atuadores conectados a TIM.

Esta tese propõe o desenvolvimento e documentação do vocabulário: um ambiente com todas as descrições de interfaces e serviços previstos pela norma, ser de código aberto para reduzir o esforço de aplicação da norma a novos projetos com redes de sensores e promover a aceitação e proliferação da norma.

Descrição do Problema

A interoperabilidade é um dos principais pontos para o sucesso da Indústria 4.0 e da Internet das Coisas Industrial. É dividido em níveis: físico, rede, sintático, semântico e plataforma [2]. O IEEE 1451 é uma família de normas desenvolvidos para gerenciar transdutores para atingir o nível sintático de interoperabilidade. Para atingir o nível semântico, é necessário utilizar um *framework*, como OPC UA e oneM2M [10]. Isso transforma o desenvolvimento de um componente industrial numa tarefa complexa. Uma vez que é necessário introduzir uma camada de *framework* que não foi desenvolvida para ser compatível com a família de normas IEEE 1451.

Para resolver o problema de interoperabilidade semântica na família de normas IEEE 1451, é necessário criar um modelo para permitir que o IEEE 1451 possa interoperar de maneira semântica. Este método precisa ser colocado dentro do NCAP e traduzir a comunicação da rede externa para a TIM e criar uma forma de levar a informação proveniente da TIM para a rede externa para transformá-la em comunicação semântica.

Hipótese de Investigação

O principal objetivo desta tese é alcançar um nível superior que o atual de interoperabilidade dentro da família de normas IEEE 1451. A revisão da literatura concluiu que o nível de interoperabilidade exigido dentro da Internet das Coisas Industrial e Indústria 4.0 é o nível semântico que permite a comunicação entre o emissor e o recetor de forma inequívoca.

Assim, o trabalho de pesquisa aqui descrito baseia-se na seguinte declaração de tese:

Como atingir o nível semântico de interoperabilidade dentro da família de normas IEEE 1451 e permitir que o IEEE 1451 seja um framework completo para comunicação dentro do contexto da Indústria 4.0.

Seguindo a declaração de tese acima, o plano de pesquisa que sustenta esta tese foi organizado.

Plano de Investigação

- *Introdução e organização do trabalho:* Este trabalho aborda a utilização da interoperabilidade de sensores inteligentes no contexto da Internet das Coisas Industrial e na Indústria 4.0. O objetivo principal é conectar a diversidade de normas em desenvolvimento. A família de normas IEEE 1451 fornece as principais diretrizes para conectar dispositivos utilizando o paradigma publicar/assinar ou cliente-servidor. Esta tarefa visa prospetar o impacto da interoperabilidade, permitindo que outros se comuniquem com as normas IEEE 1451. Esta tarefa permitirá entender o problema de interoperabilidade e prospetar uma direção.
- *Compreensão dos conceitos:* Esta tarefa será responsável por um estudo detalhado sobre a Internet da Coisas relacionado a Internet das Coisas Industrial, Sistemas Ciberfísicos, Sistemas de Produção Ciberfísicos, Indústria 4.0 e Gémeo Digital. encontrados na literatura. A pesquisa sistemática diz respeito aos seguintes aspetos: a visão geral do conceito, métodos de implementação, desafios, problemas e valor agregado, ferramentas de validação e o papel da normalização para conectar

dispositivos na Internet das Coisas Industrial e Indústria 4.0. Novas ideias são esperadas a partir da revisão e *brainstorming* resultantes desta tarefa. Isso ajudará a desenvolver um conceito sobre essas tecnologias. Isso fornecerá uma visão geral do trabalho realizado em campo, aplicativos e esforços de normalização relacionados à Internet das Coisas Industrial e Indústria 4.0. Além dos níveis de interoperabilidade que são requeridos para a indústria.

- *Uma proposta de modelo sintático para a Internet das Coisas Industrial e Indústria 4.0:* A família de normas IEEE 1451 define uma estrutura universal para troca de dados com transdutores inteligentes em um ambiente de aplicação da Indústria 4.0. Esse processo resulta da conversão de dados brutos, disponibilizando-os ao utilizador como informação. As normas IEEE 1451 preveem uma série de serviços feitos pelo NCAP, TIM e aplicações. Esta tarefa discute os modelos de normas Internet das Coisas Industrial e Indústria 4.0 sob a perspetiva da família de normas IEEE 1451. Além disso, fornece uma análise crítica da adoção do IEEE 1451 neste contexto. Com foco a atingir a interoperabilidade sintática em um contexto industrial.
- *Metodologia de Interoperabilidade para Internet das Coisas Industrial e Indústria 4.0:* A conhecer os conceitos de uma rede de sensores para Internet das Coisas Industrial e Indústria 4.0, e a ter uma visão geral, e a desenvolver um modelo de interoperabilidade baseado na família de normas IEEE 1451, é hora de estudar a aplicação destes conceitos e o modelo. Esta tarefa apresentará um estudo abrangente dos mecanismos de interoperabilidade (por exemplo, protocolos, normas, especificação de redes, modelos de referência da arquitetura) e níveis de interoperabilidade específicos para Internet das Coisas Industrial e Indústria 4.0 com base numa metodologia para melhorar o modelo. Esta tarefa compreenderá a implementação do modelo que possibilitará a interoperabilidade semântica em fábricas inteligentes. A introdução da interoperabilidade semântica aprimorará o modelo de Internet das Coisas Industrial e Indústria 4.0. Os mecanismos de interoperabilidade desenvolvidos serão implementados em fábricas inteligentes.
- *Implementação da Bancada de Testes de Interoperabilidade:* Esta tarefa visa implementar um exemplo de conceitos de interoperabilidade, modelar e validar a interoperabilidade com base em uma metodologia dentro de uma Internet das Coisas Industrial e Indústria 4.0. Essa implementação consistirá em um mecanismo interoperável entre o NCAP e a aplicação do usuário final. A aplicação do utilizador pode se comunicar a utilizar um protocolo de aplicação (por exemplo, HTTP, MQTT, SNMP, WebServices). Em seguida, a solicitação chega ao NCAP. A NCAP recebe a solicitação para obter os dados do TIM. O TIM responde a NCAP com os dados. A

NCAP traduz as informações recebidas do TIM usando a camada semântica para codificar os dados com base no contexto definido para a norma IEEE 1451. A NCAP responde à solicitação com informações semânticas. Em seguida, o utilizador recebe os dados solicitados e os decodifica a utilizar o mesmo contexto usado pela NCAP para codificar na camada semântica.

- *Apresentação, Avaliação e Discussão de Resultados:* Uma diversidade de normas encontrados no contexto Internet das Coisas Industrial e Indústria 4.0. No entanto, cada norma tem sua maneira de conectar e transferir os dados. A interoperabilidade é uma via essencial neste contexto. Portanto, mover dados de um ambiente para outro sem se preocupar com os dispositivos ou rede que receberão ou solicitarão essas informações melhora drasticamente a comunicação no contexto Internet das Coisas Industrial e Indústria 4.0.

O objetivo é desenvolver um conceito, construir um modelo e validar o modelo baseado na interoperabilidade entre as normas conectados à Internet das Coisas Industrial e Indústria 4.0. O *hardware* e *software* implementados com base na família de normas IEEE 1451 devem melhorar a interoperabilidade do sistema, fornecendo dados do ambiente em tempo real. O conhecimento adquirido ajudará a melhorar aspetos relacionados à interoperabilidade em Internet das Coisas Industrial e Indústria 4.0. Essas melhorias da interoperabilidade, são validadas continuamente durante desenvolvimentos desta tese.

Principais Contribuições

A primeira contribuição desta tese é promover um levantamento dos conceitos de Internet das Coisas, Internet das Coisas Industrial, Indústria 4.0, Sistemas Ciberfísicos, Sistemas Ciberfísicos em Produção e Gémeo Digital. Isso nos permite-nos entender esses conceitos e focar a tese na Internet das Coisas Industrial, Indústria 4.0 e nos Sistemas Ciberfísicos.

A segunda contribuição foi a revisão da interoperabilidade entre os conceitos apresentados anteriormente e o requisito de interoperabilidade dentro do contexto da Indústria 4.0.

A terceira contribuição é o estudo, modelagem e desenvolvimento de uma camada de comunicação sintática para suprir o requisito de interoperabilidade da Indústria 4.0 baseada na família de normas IEEE 1451 com outras normas.

A quarta contribuição é a proposta de uma nova camada semântica dentro da NCAP para codificar as mensagens usando o vocabulário semântico baseado na ontologia pesada desenvolvida com base na norma IEEE 1451.

A quinta contribuição é abrir o vocabulário para permitir que outros desenvolvedores criem sua própria ontologia leve com base no vocabulário disponível.

Trabalhos Futuros

Para trabalhos futuros há algumas direções que podem ser abordadas:

- Continuar o melhoramento do vocabulário desenvolvido durante a tese.
- Fazer a melhoria para a nova versão da norma IEEE 1451.0-2007 para a norma IEEE P1451.0 quando está revisão for publicada. Junto com o estudo e adequação para as novas normas que estão a ser desenvolvidas para a família de normas IEEE 1451.
- Outra característica importante para Indústria 4.0 é a segurança. Estudar e implementar mais protocolos de segurança fornecidos pela norma, como por exemplo, mecanismo de blockchain.
- Utilizar inteligência artificial para melhorar a ontologia proposta e também em casos de usuário específicos.
- Desenvolver uma plataforma gráfica *low-code* ou *no-code* para ajudar no desenvolvimento de novas ontologias baseadas na família de normas IEEE 1451.

Referências

[1] M. Noura, M. Atiquzzaman, and M. Gaedke, "Interoperability in Internet of Things infrastructure: classification, challenges, and future work," 2017, pp. 11–18.

[2] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, Opportunities, and Directions," *IEEE Trans. Ind. Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[3] L. Monostori *et al.*, "Cyber-physical systems in manufacturing," *Cirp Ann.*, vol. 65, no. 2, pp. 621–641, 2016.

[4] J. Nilsson and F. Sandin, "Semantic interoperability in Industry 4.0: survey of

recent developments and outlook," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, 2018, pp. 127–132.

[5] C. Boje, A. Guerriero, S. Kubicki, and Y. Rezgui, "Towards a semantic Construction Digital Twin: Directions for future research," *Autom. Constr.*, vol. 114, p. 103179, 2020, doi: https://doi.org/10.1016/j.autcon.2020.103179.

[6] T. Burns, J. Cosgrove, and F. Doyle, "A Review of Interoperability Standards for Industry 4.0.," *Procedia Manuf.*, vol. 38, pp. 646–653, 2019.

[7] I. S. IEEE 1451.0, "IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats," *IEEE Std 1451.0-2007*, pp. 1–335, 2007, doi: 10.1109/IEEESTD.2007.4338161.

[8] J. Higuera, W. Hertog, M. Perálvarez, J. Polo, and J. Carreras, "Smart Lighting System ISO/IEC/IEEE 21451 Compatible," *IEEE Sens. J.*, vol. 15, no. 5, pp. 2595–2602, 2015.

[9] R. Joshi, P. Didier, J. Jimenez, and T. Carey, "The Industrial Internet of Things Volume G5: Connectivity Framework," *Ind. Internet Consort.* , 2017, [Online]. Available: https://www.iiconsortium.org/pdf/IIC_PUB_G5_V1.0_PB_20170228.pdf.

[10] U. Löwen, R. Neubert, G. Bachmann, S. M. IIC, and H. Munz, "Architecture Alignment and Interoperability," 2017, [Online]. Available: https://www.iiconsortium.org/pdf/JTG2_Whitepaper_final_20171205.pdf.

Contents

List of Figures

List of Tables

Acronyms

AAS	Asset Administration Shells
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
CIFB	Communication Interface Function Blocks
CoAP	Constrained Application Protocol
CoE	Center of Expertise
CPPS	Cyber-Physical Production Systems
CPS	Cyber-Physical Systems
DDS	Data Distribution Service
DT	Digital Twin
ECC	Execution Control Chart
ERP	Enterprise Resource Planning
FB	Function Block
HTTP	Hypertext Transfer Protocol
I4.0	Industry 4.0
IEC	International Electrotechnical Commission
IECON	Annual Conference of the Industrial Electronics Society
IEEE	Institute of Electrical and Electronics Engineers
IES	Industrial Electronics Society
IICF	Industrial Connectivity Framework
IIoT	Industrial Internet of Things
IIRA	Industrial Internet Reference Architecture
IoT	Internet of Things
iRTT	Initial Retransmission Time
ISO	International Organization for Standardization
ISIE	Symposium on Industrial Electronics
IT	Information Technology
JSON	JavaScript Object Notation
LD	Linked Data
LR-WPAN	Low-Rate Wireless personal Area Network
M2M	Machine-to-Machine
MES	Manufacturing Execution System
MIB	Manger Information Base

MMI	Mixed-Mode Interface
MQTT	Message Queueing Telemetry Transport
ms	Milliseconds
NCAP	Network Capable Application Processor
O&M	Observation and Measurement
OGC	Open Geospatial Consortium
OPC UA	Open Platform Communications Unified Architecture
OSI	Open Systems Interconnection
OT	Operational Technology
OWL	Web Ontology Language
PHY	Physical Communication Layer
PLC	Programmable Logic Controller
PT	Portugal
QoS	Quality of Service
RAMI 4.0	Reference Architecture Model for Industry 4.0
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RFID	Radio Frequency Identification
SCADA	Supervisory Control and Data Acquisition
SIFB	Service Interface Function Blocks
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SSN	Semantic Sensor Network
SOS	Sensor Observation Service
SOSA	Sensor, Observation, Sample, and Actuator
STIM	Smart Transducer Interface Model
SWE	Sensor Web Enablement
TDL	Template Description Language
TEDS	Transducer Electronic Data Sheet
TCP	Transmission Control Protocol
TII	Transducer Independent Interface
TIM	Transducer Interface Module
TTL	Terse RDF Triple
UART	Universal Asynchronous Receiver-Transmitter
UBI	University of Beira Interior
UML	Unified Modeling Language
URI	Uniform Resource Identifier

URL	Uniform Resource Locator
USA	United States of America
XMPP	Extensible Messaging and Presence Protocol
W3C	World Wide Web Consortium
WTIM	Wireless Transducer Interface Module

Chapter 1

Introduction

This Chapter summarises the research work by the Ph.D. thesis titled "A Semantic Interoperability Model Based on the IEEE 1451 Family of Standards Applied to the Industry 4.0". This thesis focuses on the study and proposal of a new semantic layer to promote interoperability in the Industry 4.0 context based on the IEEE 1451 family of standards. The first Chapter describes the thesis structure, the problem addressed, and the main contribution of this study.

1.1 Motivation

The thesis is a part of the project "INDTECH 4.0 – Novas tecnologias para fabricação inteligente", funded by Program Portugal 2020 (PT 2020) within the Programa Operacional Competitividade e Internacionalização (POCI) by the Fundo Europeu de Desenvolvimento Regional (FEDER). The main objective of this thesis is to develop and standardise intelligent transducer communication to operate in an industrial environment.

Interoperability is essential for communicating with devices from different vendors inside the Internet of Things (IoT). It was estimated that interoperability is responsible for 40% of the benefits of connecting devices inside the IoT world [1]. The lack of interoperability is responsible for the increase in development costs in many areas, such as the industry. IoT focused on the industry sector receives the concept of Industrial IoT (IIoT). Inside the IIoT concentrate on the manufacturing process is the industry 4.0 (I4.0) [2]. The I4.0 is composed of the physical part and its digital representation. The physical part is the Cyber-Physical Systems (CPS) its digital contra part is the Digital Twin (DT).

Many standards can be employed to create the connection between the physical part and the digital domain. It has been addressed in both the academy and the industry. Being interoperability is one of the essential elements of a successful CPS [3]. Inside the levels of interoperability, the syntactic and semantic levels are the most employed in the industry [4]–[6].

This thesis aims to solve the interoperability problem in Industry 4.0 and the Industrial Internet of Things. Levels of interoperability can be achieved during the communication from the device level to the semantic level. The standard proposed for the industry was the IEEE 1451 family of standards, focusing on transducers' data acquisition and transmission [7]. This standard reached the syntactic level of interoperability in its current version [8]. Semantic interoperability was proposed and implemented in a vocabulary and ontology based on the IEEE 1451, turning the standard into a framework reaching all the communication levels needed for Industry 4.0 and the Industrial Internet of Things communication [9].

The IEEE 1451 family of standards enables data management with the transducer to acquire sensor data and sends commands to actuators. As the standard predicts sensor networks as two different parts, according to the standard, a sensor network includes a Network Capable Application Processor (NCAP), which is a gateway between the actual transducer (sensor and actuator) installed in the Transducer Interface Module (TIM). Inside the TIM is stored the Transducer Electronic Data Sheet (TEDS). A TEDS contain information about the TIM and the transducer channel utilised to access the sensors or actuators connected to the TIM.

This thesis proposed to develop the vocabulary and documentation: an environment with all descriptions of interfaces and services predicted by the standard, to be open source to reduce the effort of applying the standard to new projects with sensor networks and promote the acceptance and proliferation of the standards.

1.2 Problem Definition

Interoperability is one of the main points for success in Industry 4.0 and the Industrial Internet of Things. It is divided into levels: physical, network, syntactic, semantic, and platform [2]. The IEEE 1451 is a family of standards developed to manage transducers to achieve the syntactic level of interoperability. To accomplish the semantic level, it needs to use a framework, such as OPC UA and oneM2M [10]. It turns the development of an industrial component into a complex task. It needs to introduce a framework layer that was not developed to work with the IEEE 1451 family of standards.

To address the semantic interoperability problem in the IEEE 1451 family of standards need to create a model to allow the IEEE 1451 to interoperate semantically. This model needs to be placed inside the NCAP, translating the communication from the external network to the TIM. Then, it reads TIM's data and turns it into a semantic communication.

1.3 Research Hypothesis

The main objective of this thesis is to reach a higher level of interoperability inside the IEEE 1451 family of standards. The literature review concluded that the level of interoperability required inside the Industrial Internet of Things and Industry 4.0 is the semantic level that allows communication between the sender and the receiver in an unambiguous way.

Thus, the research work described here builds upon the following thesis statement:

How to reach the semantic level of interoperability inside the IEEE 1451 family of standards allowing the IEEE 1451 to be a complete framework for communication inside the Industry 4.0 context.

Following the thesis statement above, the research plan underpinning this thesis was organised into the following tasks:

- *Work introduction and organisation*: This work extends the utilisation of the Interoperability of intelligent sensors in the Industrial Internet of Things (IIoT) and Industry 4.0 (I4.0). The main objective is to connect the diversity of standards in development. The work fits into the context of IIoT and I4.0. The IEEE 1451 family of standards provides the main guidelines to connect devices using the paradigm publish/subscribe or client-server. This task intends to prospect the interoperability impact, allowing others to communicate to IEEE 1451 standards. This task will enable an understanding of the interoperability problem and prospect a direction.
- *Understanding the concepts*: This task will be responsible for a detailed study of the IoT related to the IIoT, CPS, CPPS, I4.0, and DT found in the literature. The systematic research concerns the following aspects: the concept overview, implementation methods, challenges, issues and added value, validation tools, and the role of the standardised to connect devices in IIoT and I4.0. New ideas are expected from the review and brainstorming resulting from this task. That will help to develop a concept about these technologies. This will give an overview of the work done in the field, applications, and standardisation efforts related to

IIoT and Industry 4.0. Moreover, interoperability levels are required for the industry.

- *A syntactic model proposal for the Industrial Internet of Things and Industry 4.0*: The IEEE 14151 family of standards defines a universal framework for exchanging data with smart transducers in an Industry 4.0 application environment. This process results from raw data conversion, making it available to the user as information. The IEEE 1451 standards predict a series of services done by the Network Capable Application Processor (NCAP), Transducer Interface Management (TIM), and applications. This task discusses the IIoT and I4.0 standards models from the perspective of the IEEE 1451 family of standards. Also, it provides a critical analysis of the adoption of IEEE 1451 in this context. With a focus on achieving syntactic interoperability in an industrial context.
- *Interoperability Methodology for Industrial Internet of Things and Industry 4.0:* Knowing the concepts of an IIoT and I4.0, having an overview, and developing a model of interoperability based on the IEEE 1451 family of standards, it is time to study the application of these concepts and the model. This task will present a comprehensive study of the interoperability mechanisms (e.g., protocols, standards, networks specification, reference architecture models) and interoperability levels specific to IIoT and I4.0 based on a methodology to improve the model. This task will help us understand the syntactic interoperability mechanism implementation in smart factories. Introducing semantic interoperability will enhance the model of IIoT and I4.0. The interoperability mechanisms developed will be implemented in smart factories.
- *Implementation of Interoperability's Testing Workbench:* This task aims to implement an example of interoperability concepts, model and validate the interoperability based on a methodology inside an IIoT and I4.0. This implementation will consist of an interoperable mechanism between the Network Capable Application Processor (NCAP) and the final user's application. The user's application can communicate using an application protocol (e.g., HTTP, MQTT, SNMP, Webservices). Then the request reaches the NCAP. The NCAP receives the request to get the data from the TIM. The TIM answers the NCAP with the data. Then, the NCAP translates the received information from the TIM using the semantic layer to encode the data based on the context defined for the IEEE 1451 standards. The NCAP responds to the request with semantic information. Then the user received the requested data and decoded it using the same context used by the NCAP to encode the semantic layer.

- *Results Presentation, Evaluation, and Discussion:* A diversity of standards found application in the IIoT and I4.0 context. However, each standard has its way of connecting and transferring the data. Interoperability is an essential track in this context. So, moving data from one environment to another without worries about the devices or network that will receive or request that information dramatically improves the IIoT and I4.0.

The objective is to develop a concept, build a model, and validate the model based on interoperability between standards connected to the IIoT and I4.0. The hardware and software implemented based on IEEE 1451 family standards should improve the system's interoperability, providing real-time data from the environment. The knowledge will help improve interoperability aspects in IIoT and I4.0. Those improvements and what needs to enhance interoperability continuously validate all developments within this thesis.

1.4 Main Contributions

The first contribution of this thesis is to promote a survey of the concepts of the Internet of Things, Industrial Internet of Things, Industry 4.0, Cyber-Physical Systems, Cyber-Physical Production Systems, and Digital Twin. This allows us to understand these concepts and focus the thesis on Industry 4.0 and Cyber-physical Systems.

The second contribution was the review of the interoperability between the concepts presented before and the requirement for interoperability inside the Industry 4.0 context.

The third contribution is studying, modelling, and developing a syntactic communication layer to supply the interoperability requirement for Industry 4.0 based on the IEEE 1451 family of standards with other standards.

The fourth contribution is the proposed new semantic layer inside the NCAP to encode the messages using the semantic vocabulary based on the heavyweight ontology developed by the IEEE 1451 standards.

The fifth contribution is to open the vocabulary to allow others to create their lightweight ontology based on the available vocabulary.

1.5 The Organisation of the Thesis

This thesis consists of five Chapters, which are organised as follows. The first Chapter presents the scope of the thesis, focusing on motivation, the definition of the problem, the research hypothesis, the main contribution, and the thesis organisation.

Chapter 2 reviews the concepts employed in this thesis entitled "Background and Concepts Review". This Chapter has presented the concepts of the Internet of Things, Industrial Internet of Things, Industry 4.0, Cyber-Physical Systems, Cyber-Physical Production Systems, and Digital Twin, along with the interoperability levels and reference architecture models. Moreover, this Chapter has exposed the platforms, middleware, operational systems, and hardware for the Internet of Things.

Chapter 3, entitled "Syntactic Interoperable Model for the Industrial Internet of Things and Industry 4.0", used the concept defined in the background to develop a syntactic level of interoperability between the IEEE 1451 and the IEC 61499 standards. The IEEE 1451 family of standards manage the transducers, whereas the IEC 61499 standard was employed on the control system. Semantically interoperable scenarios and interoperability validation are expressed in this Chapter.

Chapter 4 presents the semantic layer proposal to reach semantic interoperability based on the IEEE 1451 family of standards, entitled "Proposed Semantic Interoperability Methodology for Industrial Internet of Things and Industry 4.0". This Chapter presents the model and methodology developed to promote the semantic level of interoperability based on IEEE 1451. The vocabulary and ontology developed in this thesis are shown, and the validation based on the Digital Twin implementations is presented.

Chapter 5 presented the conclusion and future work entitled "Conclusions and Future Work". This Chapter has exposed the relevance of the work and future directions based on the work developed in this thesis.

References

[1] M. Noura, M. Atiquzzaman, and M. Gaedke, "Interoperability in Internet of Things infrastructure: classification, challenges, and future work," 2017, pp. 11–18.

[2] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, Opportunities, and Directions," *IEEE Trans. Ind.*

Informatics, vol. 14, no. 11, pp. 4724–4734, 2018.

[3] L. Monostori *et al.*, “Cyber-physical systems in manufacturing,” *Cirp Ann.*, vol. 65, no. 2, pp. 621–641, 2016.

[4] J. Nilsson and F. Sandin, “Semantic interoperability in Industry 4.0: survey of recent developments and outlook,” in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, 2018, pp. 127–132.

[5] C. Boje, A. Guerriero, S. Kubicki, and Y. Rezgui, “Towards a semantic Construction Digital Twin: Directions for future research,” *Autom. Constr.*, vol. 114, p. 103179, 2020, doi: https://doi.org/10.1016/j.autcon.2020.103179.

[6] T. Burns, J. Cosgrove, and F. Doyle, “A Review of Interoperability Standards for Industry 4.0.,” *Procedia Manuf.*, vol. 38, pp. 646–653, 2019.

[7] I. S. IEEE 1451.0, “IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats,” *IEEE Std 1451.0-2007*, pp. 1–335, 2007, doi: 10.1109/IEEESTD.2007.4338161.

[8] J. Higuera, W. Hertog, M. Perálvarez, J. Polo, and J. Carreras, “Smart Lighting System ISO/IEC/IEEE 21451 Compatible,” *IEEE Sens. J.*, vol. 15, no. 5, pp. 2595–2602, 2015.

[9] R. Joshi, P. Didier, J. Jimenez, and T. Carey, “The Industrial Internet of Things Volume G5: Connectivity Framework,” *Ind. Internet Consort.* , 2017, [Online]. Available: https://www.iiconsortium.org/pdf/IIC_PUB_G5_V1.0_PB_20170228.pdf.

[10] U. Löwen, R. Neubert, G. Bachmann, S. M. IIC, and H. Munz, “Architecture Alignment and Interoperability,” 2017, [Online]. Available: https://www.iiconsortium.org/pdf/JTG2_Whitepaper_final_20171205.pdf.

Chapter 2

Concept Review and Background

2.1 Introduction

This Chapter presents the background for this thesis. The background was utilised to create the basic knowledge for all published papers. Two papers are closely related to this Chapter:

Semantic Interoperability in the Industry 4.0 Using the IEEE 1451 Standard

H. da Rocha, A. Espirito-Santo and R. Abrishambaf, *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, Singapore, 2020, pp. 5243-5248, doi: 10.1109/IECON43393.2020.9254274.

An Interoperable Digital Twin with IEEE 1451 Standards

H. da Rocha, J. Pereira, R. Abrishambaf, and A. Espirito Santo, *Sensors*, vol. 22, no. 19, p. 7590, Oct. 2022, doi: 10.3390/s22197590.

2.2 State of the Art

2.2.1 Internet of Things

The Internet of Things (IoT) concept is receiving increasing attention, not only from the academic community but also from society. It is already possible to find available commercial products and services. The expression "Internet of Things" was first employed in a presentation to the P&G group by Kevin Asheton in 1999 [1]. IoT will enable billions of intelligent devices used in everyday life to see, hear, think, and talk together, sharing information and coordinating decisions through the Internet [2]. A more immersive and pervasive Internet experience can be achieved by introducing the IoT concept in a diversity of areas presented by Zanella et al. [3] as home automation and industrial automation, mobile healthcare, intelligent energy management, and others. The investment in the IoT is expected to be billions of dollars in the following years [2], [4].

IoT has been amply studied. Al-Fuqaha et al. [2] developed a survey about the IoT. Its survey presents the integration between the vertical and the horizontal market. The vertical market is the domain-specific application (e.g., smart home, industry, agriculture, and healthcare). In contrast, the horizontal market is the application of the same domain-independent computing and analytical services to more than one specific domain without changing it. The survey develops the IoT components, the IoT economic impact, the IoT architectures, the IoT elements (i.e., identification, sensing, communication, hardware, software, service, and semantics), and the IoT standards.

Moreover, the authors Khan et al. [5] exposed IoT challenges in availability, reliability, mobility, performance, management, scalability, interoperability, security and privacy. The IoT architecture was presented with five layers (business, application, middleware, network, and perception layer). To enable possible future applications, the authors discussed the prediction of natural disasters, industry applications, water scarcity monitoring, design of smart homes, medical applications, agriculture applications, intelligent transport systems design, smart cities, smart metering and monitoring, and smart security. The authors addressed the challenges of naming and identity management, interoperability, standardisations, and information privacy.

Whitmore et al. [6] wrote a survey on topics and trends in IoT as topics and trends were discussed about hardware represented by Radio Frequency Identification (RFID), Near Field Communication (NFC), and Sensor Networks (SN). Software is discussed as a middleware to connect the variety of devices in the IoT and the searching/browsing. The architecture comprised hardware/network, software, process, and general/requirements for the IoT. Smart infrastructure, healthcare, supply chain/logistics, and social applications are application topics. The authors presented a challenge to security, privacy, and legal/accountability. It defines five essential IoT technologies (RFID, Wireless Sensor Network (WSN), middleware, cloud computing, and IoT application software). Lee et al. [7] presented the challenger in the IoT context, such as data management, data mining, privacy, security, and chaos (handling the complex context and communication), as presented by. A survey about the IoT was written by Atzori et al. [8]. It introduced the paradigms and technologies developed in identification, sensing and communication. Also, it introduced the communication protocols that can be used to provide data exchange in the IoT and the domains that IoT can be applied to (i.e., transportation and logistics, healthcare, smart environment, personal and social, and futuristic). Moreover, Atzori et al. presented standardised topics addressing networking, security, and privacy issues.

2.2.2 Industrial Internet of Things

Many areas are being developed in the IoT world, including the industry. The Industrial Internet of Things (IIoT) covers the Machine-to-Machine (M2M) domain and industrial communications technologies with autonomous applications. It enables a better understanding of the manufacturing process and promotes efficiency and sustainability in the production process [9]. Differences between the IoT and IIoT concepts are better understood after studying the concept of each one. IoT was defined by Ray and ITU as "A global infrastructure for the information society enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving, interoperable information and communication technologies" [10], [11]. Whereas the IIoT definition by Sisinni et al. [9] is "IIoT is about connecting all the industrial assets, including machines and controls systems, with the information systems and the business processes". Another definition was from GE in 2012, as IIoT entails the adoption of the IoT from the perspective of industry in general (manufacturing and nonmanufacturing) [12].

The authors Da Xu et al. [13] wrote a survey reviewing the current state of the art concerning the research progress achieved in IoT, technological advances, industrial applications, research trends and challenges. The authors presented a four-layer architecture for IoT based on the sensing, networking, service, and interface layers. To provide the communication link, the authors introduced standards that could be used to connect devices, such as RFID, NFC, IEEE 802.11 (Wi-Fi), IEEE 802.15.4 (ZigBee), IEEE 802.15.1 (Bluetooth), Wireless Sensors, Mesh Networks, Low power Wireless Personal Area Networks (6LoWPAN), and M2M. The concept that supports the IIoT architecture is introduced by Sisinni et al. [9]. The same authors address topics such as IIoT connectivity and standardisation and identify energy efficiency, real-time performance, coexisting and interoperability, and security and privacy as the main challenges to the development and acceptance of IIoT. The development of the Industrial Internet of Things and Industry 4.0 in fog computing was written by Aazam et al. [12]. Fog computing can be a middleware node performing specific industry or smart factory tasks. The authors also wrote about the IIoT components, such as the localisation of Wireless Sensor Networks (WSN) and Wireless Sensor Actuator/actor Networks (WSANs), to control and manage the Cyber-Physical Systems (CPS), industrial big data analytics, virtual sensing and virtual sensor networking, Web of Things (WoT) for the industry. Inside fog computing architecture, they present the concepts for data mining, smart grid and power, transportation, waste management, food, agriculture, advertisement, third-party delivery, and smart parking. As challenges in the IIoT, the author brings the discussion about energy consumption and management, interoperability of devices, Service Level Agreement (SLA) and interoperability of services, security and privacy of data and workers, context and semantics of aware service provisioning, fault detection and reconfiguration, user-friendliness in the product deployment and usage [12].

Gilchrist [14] wrote a book about Industry 4.0 and the Industrial Internet of Things. In this book, the author wrote that IIoT integrates machine sensors, middleware, software, backend cloud computing, and storage systems to increase efficiency and accelerate productivity, resulting in industrial process downtime and optimisation. The author presented use cases for IIoT in the healthcare, oil and gas industry, smart office, logistics, retail, IIoT references, designed of IIoT systems, networking technologies and protocols, and examination of the middleware transport protocols, the middleware software patterns (e.g., publish/subscribe), software design (e.g., Application Programming Interface – API, Web Services, SOAP, REST, others), IIoT platforms, paradigms, and challengers.

2.2.3 Industry 4.0

Industry 4.0 was developed inside the IIoT concept as “addressing the use of Internet technologies to improve production efficiency employing smart services in smart factories” [9]. For Aazam et al., the term Industry 4.0 (I4.0) “refers to the current fourth generation of industry focusing on the manufacturing industry scenario, only that is a subset of IIoT” [12]. The I4.0 component appears as a combination of objects from the physical world and the information world, offering dedicated functionality and flexibility inside and outside an I4.0 network. It can be materialised as a module, a device, or an entire system [15].

Ye and Hong [15] wrote about the Industry 4.0 components, insights and implementation of Asset Administration Shells (AAS). AAS is a virtual digital representation of the I4.0 concept to establish communication between I4.0 components. Qin et al. [16] wrote about current manufacturing systems' state of the art. It described the gap between the existing manufacturing systems and the I4.0 (i.e., single station automated cells, automated assembly systems, flexible manufacturing systems, computer-integrated manufacturing systems, and manufacturing systems). Moreover, a multi-layer framework was proposed composed of three levels (control, integration, and intelligence) for implementing the I4.0 structure. A review of intelligent manufacturing in the context of I4.0 describes concepts such as smart manufacturing, IoT manufacturing, and cloud manufacturing. The authors present a review of the technologies IoT, Cyber-Physical Production Systems (CPPS), cloud computing, Big Data Analyses (BDA), and Information and Communication Technology (ICT). Also, it is presented the efforts from different nationalities in intelligent manufacturing (e.g., European Union, United States, Japan, and China) developed by Zhong et al. [17]. The conceptual influence and importance of I4.0 and how it contributes to adding value to organisations and society. A theoretical framework of I4.0 based on three points (digitalisation of production, automation, and automatic data interchange) is also proposed. The fundamental concepts of I4.0 are presented in this same research (e.g., smart factory, new systems in developing products and services, self-organisation, smart product, new systems in distribution and procurement, adoption of human needs, CPS, smart city, and digital sustainability) was presented by Roblek et al. [18].

The design principles for I4.0 scenarios are based on quantitative and qualitative literature reviews. Resulting in design principles in interconnection, information transparency, decentralised decisions, and technical assistance written by Herman et al. [19]. A detailed survey of relevant articles related to Industry 4.0, including concepts and perspectives, CPS based on I4.0, interoperability of Industry 4.0 (operational, systematically, technical, and semantic), key technologies, smart facture and manufacturing, smart product, and the smart city was exposed by Lu [20].

The I4.0 comprises a smart manufacturing chain stack shown in Figure 2.1. The sensor and actuator at the physical layer acquire data or process an action. The second layer is responsible for controlling the physical devices. Supervisory Control and Data Acquisition (SCADA) is the software responsible for controlling the industrial process, equipment, and data used. The Human Machine Interface (HMI) shows the process running to the operator. The Manufacturing Execution System (MES) tracks and documents all the data and information inside the manufacturing process to help the decision maker to optimise and improve the production linked with Enterprise Resource Planning (ERP). The ERP is the system utilised by the enterprise to manage. It integrates its business areas (e.g., planning, purchasing, sales, marketing, and finance), sharing cross information with all the areas inside the enterprise [21], [22].

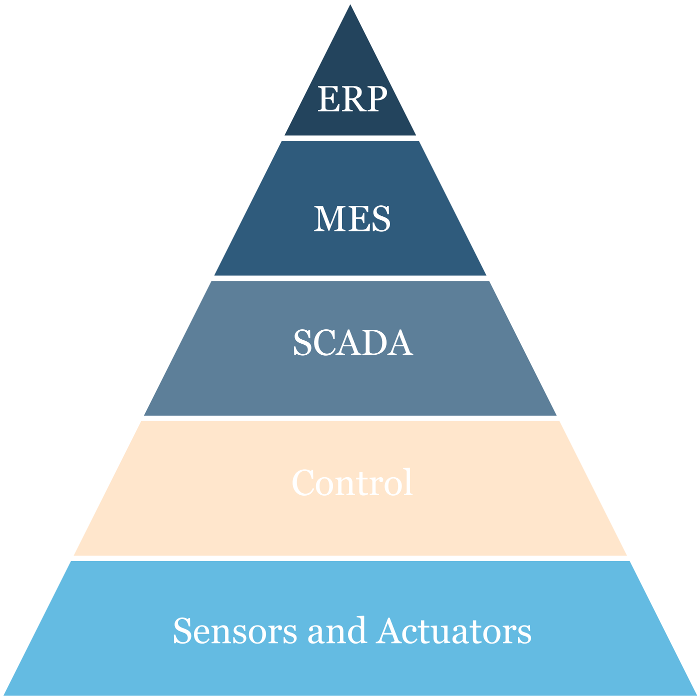


Figure 2.1 Classic smart manufacturing stack.

2.2.4 Cyber-Physical Systems

Industry 4.0 is mainly shaped by the CPS and the Internet of Things and Service concepts [23]. The CPS is the core foundation for the I4.0 [24], which promotes the integration of the physical world with its associated digital world [9], [23]–[26]. The term CPS was introduced in 2006, but the concept that turned into the CPS is from the 70s [27]. It is responsible for realising the intelligent cross-link between the applications, which operates in a self-organised and decentralised manner, used by smart factories to create added value [28]. The remote access and control of smart systems and devices in the physical world are promoted by the CPS and play an essential role in control, diagnostics, maintenance, and assistance in controlling machines, devices, sensors, and actuators, contributing to the industrial process [12]. Machines in I4.0 are integrated with Information and Communication Technologies (ICT), the basis for smart automation. Furthermore, the systems can be autonomous and make their own decision based on machine learning and real-time data capture, analytics results, and recorded past behaviours [29].

2.2.5 Cyber-Physical Production Systems

Cyber-Physical Production System (CPPS) is the implementation of I4.0 and CPS in manufacturing and production, crossing all production levels between the autonomous and cooperative elements, such as Smart Machines and sub-systems and Smart Factories [23], [30]. It is responsible for connecting the virtual space with the physical world, which allows the smart factories to be more intelligent, resulting in better and smarter production conditions, increasing productivity, production efficiency, and product quality [24], [31]. This can be combined or not with the human-machine interaction [31]. The CPPS found application in production design, logistics/supply chain, production process, maintenance services and recycling. Different architectures can be developed for different production lines (e.g., assembly, stamping, and machining lines) [32]. The main characteristics of a CPPS are their ability to acquire information and act autonomously intelligently, to set up and connect to other elements or systems, giving them the connectedness property, and the responsiveness towards internal and external changes [33].

The effects of I4.0 on the planning and scheduling process are presented and investigated by Hermann et al., [19]. Further, it was analysed that the main contribution is based on the core elements for improving production. Based on the literature review performed by the author, the core elements are connectedness, smart machine and products, decentralisation, big data, and cybersecurity. An implemented the CPPS in learning factories in manufacturing training and education was presented by Thiede et al. [31]. This kind of factory is an effective instrument for developing competence. A cyber-physical system architecture for machining production lines, a cloud-based cyber-physic architecture, was proposed by Herwan et al. [32]. The main objective was to provide an adaptive system for tool wear and breakage detection, composed of three layers of physical resources, a local server, and a cloud server.

In contrast, Stock and Schel [34] examined the security and secure identity for CPPS to construct a hybrid fingerprint for CPPS to increase security and build a specific identity for CPPS. Fingerprinting is defined by the Internet Engineering Task Force (IETF) as "The process of an observer or attacker uniquely identifying (with a sufficiently high probability) a device or application instance based on multiple information elements communicated to the observer or attacker" [35]. A "Digital Twin" concept was introduced by Uhlemann et al. [36], as its digital equivalent, proposes guidelines for implementing a Digital Twin in production systems for enterprises of small and medium sizes, promoting data quality based on multimodal data acquisition and contributing to developing a CPPS. Marseu et al. [37] presented an interdisciplinary engineering methodology for changeable CPPS with the focus on providing a guideline and support for engineers in the design process and based on a five-phase procedural model structure, subdividing the design process into manageable and limited phases, with transitions between the different levels happening seamlessly. In addition, it is recommended that tools and modelling languages support the procedural model. The roots (Intelligent Manufacturing Systems – IMS and the Biological Manufacturing Systems - BMS), expectations (i.e., robustness, self-organisation, self-maintenance, self-repair, safety, remote diagnosis, real-time control, autonomous navigation, transparency, predictability, and efficiency). The R&D challenges are context-adaptative and autonomous systems, cooperative production systems, identification and prediction of dynamic systems, robust scheduling, the fusion of real and virtual systems, and human-machine symbiosis for the CPPS by Monoston [38].

2.2.6 Digital Twin

A Digital Twin (DT) enables the representation of an asset from the physical world into the digital world. Grieves expressed the generic definition of digital in 2003 [39]. However, the concept of having a representation of an asset was started by NASA in the early 1960s for the Apollo mission [40]. In 2012 NASA presented a detailed definition of a DT [41]. Since the initial report was delivered, many authors have developed definitions, such as "the cyber layer of CPS, which evolves independently and keeps close integration with the physical layer" [42]. Negri, Fumagalli, and Macchi [43] defined DT as "a virtual and computerised counterpart of a physical system that can exploit a real-time synchronisation of the sensed data coming from the field and is deeply linked with Industry 4.0." and many others summarised in [40], [44] most of them the DT definition is related to CPS. However, it is difficult to classify what is or is not a digital twin because the definitions are ambiguous [39].

Three closer representations of a concept for communication of the DT are very similar: Digital Model, Digital Shadow, and Digital Twin. The Digital model does not have real-time communication between the physical and digital parts. Digital Shadow has one way of communication, physical to real-time digital communication. DT has bi-directional communication from the physical to digital and digital to real-time physical [45]–[47].

There is no consensus on DT development. Each author brings their concepts. The DT design and development occurs in stages: mirroring, monitoring, modelling and simulation, federation, and autonomous. Mirror the DT from physical to digital. Monitor and control the DT. Model and simulate the DT from the simulation result of data obtained. Federated the DT to optimise the complex objects and interoperated Federated DTs. Finally, to act autonomously to recognise and solve problems in the federated DTs [48].

Fuller et al. [49] define domains in developing DTs: application, middleware, networking, and object. The application Domain is the model architecture and visualisation, software and APIs, data collection and pre-processing. Middleware domain: stage technology; data processing. Network domains are communication technology and wireless communication. Object domains are the hardware platform and the sensor technology. In [40], Liu et al. conclude that a DT needs to be individualised, being a closer individual part of its physical representation, with high fidelity where a DT can simulate the behaviour if its physical position is as perfect as possible, updating itself as soon the physical part is updated, the communication needs to be real-time with low-latency, and controllable changes in one part needs to reflect in the other as quickly as possible. Tao et al. [50] defines five layers of DT modelling: physical and digital parts, data, connection, and service modelling.

The interaction between the concepts introduced previously is illustrated in Figure 2.2. The IoT is the concept of connecting all smart devices through the Internet. One area inside the IoT is the IIoT. With the focus on manufacturing inside the IIoT, industry 4.0 was developed. The three concepts can be combined with the CPS. Finally, the CPS, focusing on the production systems, is named the CPPS. CPS and CPPS have their representation of the physical to the digital world as Digital Twin.

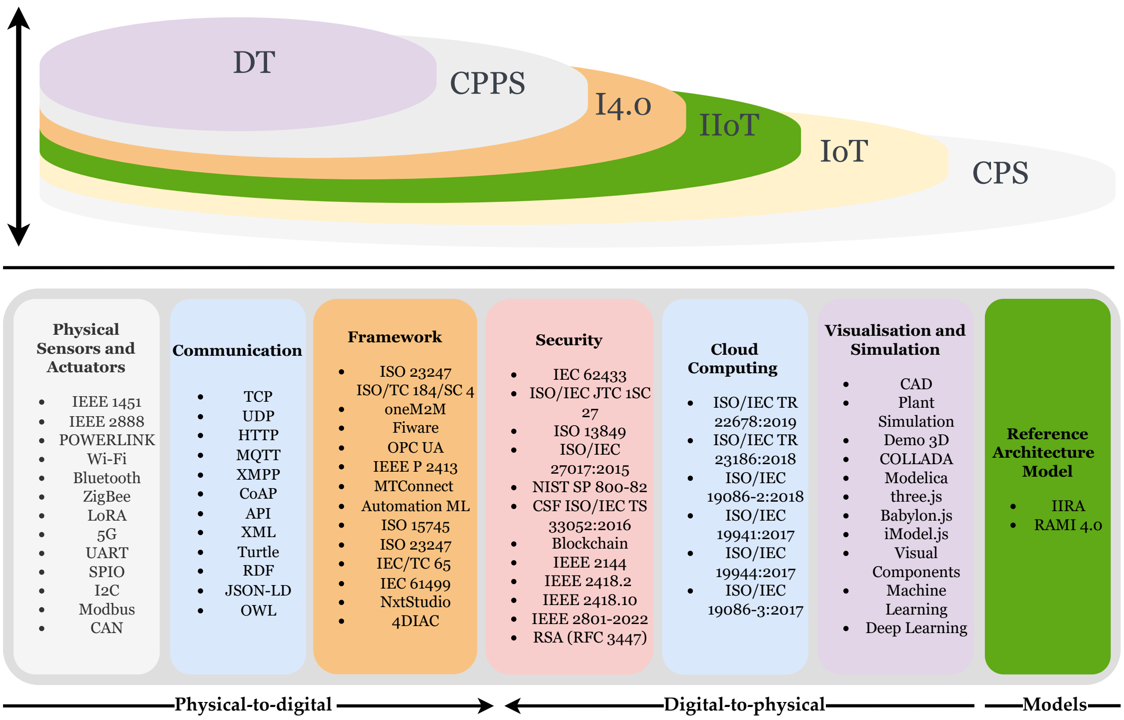


Figure 2.2 Concepts in the industrial world [51].

2.3 Interoperability

2.3.1 Reference Architecture Models

A reference architecture model provides guidelines for the interdisciplinary technologies used inside Industry 4.0. Architecture is defined as an organisational structure of a system or component, presenting relationships, principles, and guidelines that control its designs and evolution over time [52]. Whereas a reference model of architecture is a stable model that is commonly used and recommended from derivate specified and concrete architecture, also plays an essential role in the systems of an application area, describes the structure of the model and is a departure from developing tools. Also, it can provide a framework that contains a minimal set of unifying concepts, axioms, and relationships responsible for understanding the interactions between the entities inside of an environment [52].

This section presents two of the most common reference architecture models for Industry 4.0 and IIoT. Are they the Reference Architecture Model for Industry 4.0 (RAMI 4.0) and the Industrial Internet Reference Architecture (IIRA) with particular emphasis on communication because "network-based communication is the most important aspect of Industry 4.0" [53]. Moreover, there are other reference architecture models: the Smart Grid Architecture Model (SGAM) [54], the Internet of Things Strategic Research Agenda (IoT-SRA) [55], the IBM Industry 4.0 Architecture for Manufacturing [56], and the NIST Service-Oriented Architecture for Smart Manufacturing Project [57].

2.3.2 Industrial Internet Reference Architecture

The Industrial Internet Consortium (IIC) developed the IIRA in 2015 for IIoT systems. It provides implementation guidance for IIoT architectures, business leaders, and users of every level to optimise and establish IIoT systems, converging OT and IT to achieve economic benefits, based on the ISO/IEC/IEEE 42010 [58]. It specifies an Industrial Internet Architecture Framework (IIAF) for viewpoints and concerns in the development, documentation, and communication of the IIRA.

The IIRA is an open architecture that maximises the value chain by broad industry application driving interoperability, mapping applicable technologies, and guiding technology and standard development. It is a generic description and representation of high levels of abstraction supporting the industry's applicability by abstracting common characteristics, features, and patterns from use cases [58]. IIRA defined the business, usage, functional, and implementation viewpoints, as shown in Figure 2.3.

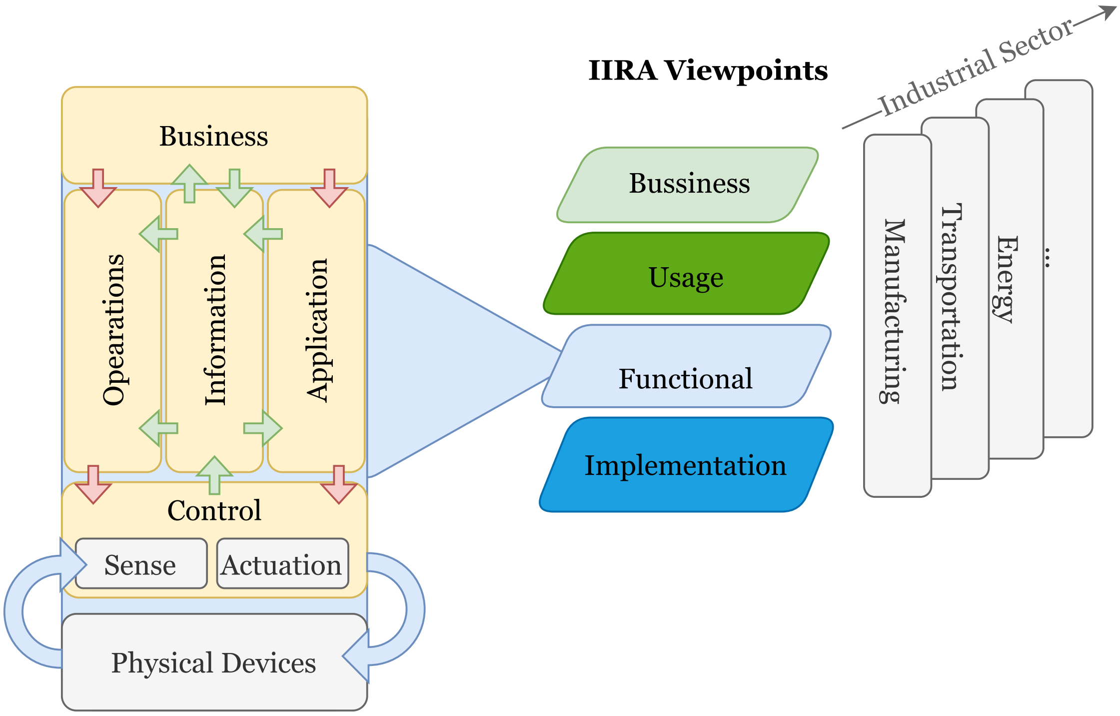


Figure 2.3 The relationship between IIRA Viewpoints, Functional Viewpoint Scope, and Industrial Sectors adapted from [58].

IIC developed the viewpoints based on various IIoT use cases. The business viewpoint corresponds to the concerns of the stakeholders' identification and their business vision, values, and objectives. Usage viewpoint corresponds to concerns of system usage, typically representing the sequence of activities that involves human or logical users that delivers functionality to achieve the fundamental system capabilities. These can be engineers, product managers, and other stokeholds. The functional viewpoint focuses on the functional components, their structures and interrelation, interfaces and interactions, relation, and interactions of the systems with external elements in the environment, to support the usage and activities of the overall system.

The implementation viewpoint exposes the technologies needed to implement the functional viewpoint, communications schemes, and life cycle procedures. This viewpoint is coordinated by the usage viewpoint and supports the business viewpoint [58]. The application scope represents the adoption of the general architecture framework as a reference architecture for real-world usage scenarios transforming and extending the abstract architecture concept and models in detailed architectures used in the industrial methods. The entire lifecycle from the IIoT system conception to the design and implementation.

The Functional viewpoint has five functional domains: control, operations, information, application, and business. The relevant aspect of the communication task is the Control domain which represents a collection of functions that can read data from sensors, apply logic rules, and execute control over the physical system through an actuator. Although the sensing function reads data from a sensor, the actuation function writes data and control signals to an actuator.

The communication channel connects sensors, actuators, controllers, gateways, and other systems. It can use network architectures and Quality of Service (QoS). The entity provides an abstraction of sensors, actuators, controllers, and systems and expresses their relationships. Modelling is responsible for understanding the systems' states, conditions, and behaviour. The Asset Management manages the control systems, including onboarding, configuration, policy, system, software/firmware updates, and other lifecycle management operations. The Executor executes logic controls to understand the states, conditions, and behaviour of the system under its control. It enables the Executor to control its objectives that can be programmed with static or dynamic configuration [58].

The implementation viewpoint is also essential for communication. It concerns the technical representation of an IIoT system, technologies, and system components requirements implementing the activities and functions described in the functional viewpoint. It describes the general architecture of an IIoT system which is the structure and distribution of components. The topology adopted a technical description of the components (interfaces, protocols, behaviours, and properties), an implementation map of activities identified in the usage viewpoint of the functional components and their implementation, and an implementation map for crucial system characteristics. The example architectures that can be implemented are the Three-tier architecture pattern, Gateway-Mediated Edge Connectivity and Management architecture pattern, and the layered Data bus pattern [58].

2.3.3 Reference Architecture Model for Industry 4.0

The German Electrical and Electronics Manufacturers' Association (ZVEI) developed RAMI 4.0 as a framework. It is composed of a three-dimensional layer model that includes the crucial elements of Industry 4.0. Smaller and simple clusters to reduce the complex interrelation [59]. The RAMI 4.0 three-dimensional structure divides into Life Cycle & Value Chain, Hierarchy Levels, and Layers. The Life Cycle & Value Chain is based on the IEC 62890 for life-cycle management, composed of types and instances. A type becomes an instance with the conclusion of its design, prototyping, and product manufacturing [59]. The Hierarchy Level is derived from the IEC 62264 developed for enterprise Information Technology (IT) and Control Systems. It represents the different functionalities between factories and facilities. The functionalities were expanded to include workpieces as products and connections to IoT and Services as Connected World [59]. At the Layer axe, six layers describe, layer by layer, the decomposition of a machine in its properties and structure. All crucial aspects of Industry 4.0 are mapped, enabling the classification of objects according to this model [59]. The RAMI 4.0 Layer axe is shown in Figure 2.4.

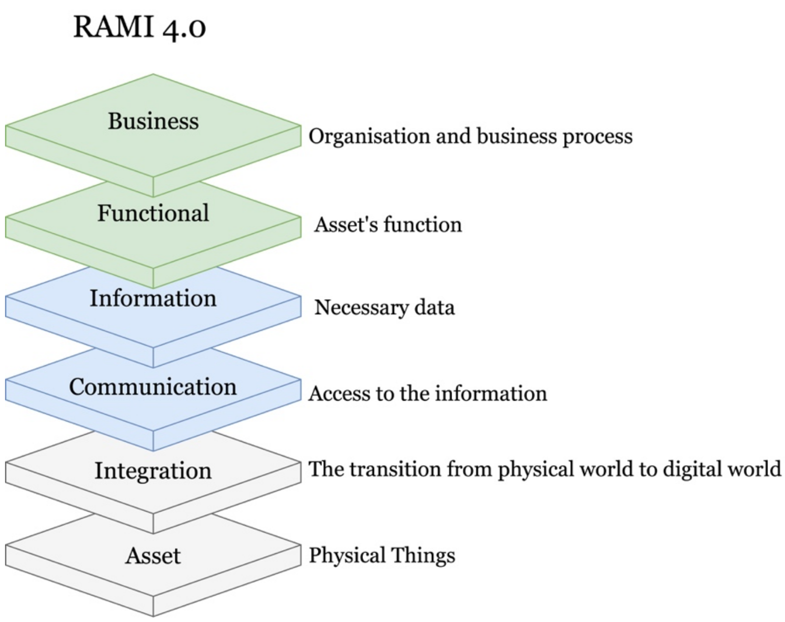


Figure 2.4 RAMI 4.0 Layer axe adapted from [59].

A document developed by the ZVEI specified how communication occurs in Industry 4.0 [53]. This reference presents a new concept of communication based on individual demands and a flexible structure. That will change over the lifetime in new hardware components and configurable software stacks for communications, protocols promise and require oriented communication with high bandwidths and fast response times, with a large number of widely distributed features developed in modest requirements. An increasing complexity and heterogeneity network is also presented, demanding a decoupling between the specified communication structure and the application. This is achieved by the Service-Oriented Architecture (SOA), which creates an abstraction from the user view and hides the communication details. Also, the process control will subscribe to data directly on the device using the publish/subscribe models [53]. The same document overviews the current situation in communication in the industrial environment and presents the industrial communication requirements based on defined scenarios.

The solution for communication in Industry 4.0 shall meet the requirements resulting from interactions. A concept used by the Asset Administration Shell (AAS) transforms physical assets, such as a device or machine, and intangible assets, such as plans or functions [53]. A communication relationship or an entire network can be an asset. The AAS covers the relevant aspects of an asset defined on RAMI 4.0. Whereas the AAS represents the properties of a network, the requirements for the process data in the engineering phase are defined and result in the Quality of Service (QoS) requirements implemented by a selection of suitable components. It also is used to develop service-level agreements between communication partners. The communication data process can occur in a non-Industry 4.0 compliant way. However, Industry 4.0 compliant communication is service-based and defines the communication layer in RAMI 4.0. It can use standardised models, such as OPC UA, which gained significance in the automation [53]. Its main point is to promote models' digitalisation and corresponding software to reduce the efforts, time, and cost of implementing the process in the various value chain based on Operational Technology (OT) and Information Technology (IT).

An implementation of RAMI 4.0 communication is described in [60]. The main contribution is the specification of the Administration Shell (AS), the asset, and the Industry 4.0 components between networked partners. Moreover, it stipulates a few structural principles of an Administration Shell but does not describe the technical interfaces of AS nor the exchange of information, protocols, or interaction partners. It focuses on transporting information from one partner to another in a value chain context. AS submodule and their structures identify the concept of access rights and roles. It can be used as a base for developments and input for discussion. The documents are composed of UML diagrams to describe the types and instances, XML files, Automation ML, OPC AU information models, and JSON or RDF to exchange the data that are represented as the data model, IRDI, URI, and URL are used for the identification of assets, AS or other properties and classifications.

2.3.4 Interoperability between IIRA and RAMI 4.0

The IIC developed a whitepaper to interoperate the RAMI 4.0 and the IIRA [61]. IIRA is focused on the IIoT across industries addressing commonality and interoperability, whereas RAMI 4.0 focuses on digitalisation and interoperability in manufacturing. Mapping the communication developed in the Functional viewpoint from IIRA to RAMI 4.0 can take advantage of common terms used in both references. An Asset in RAMI 4.0 defines anything (network, product, raw material, and software). In contrast, an Asset in IIRA refers to a physical thing monitored and controlled. Functional in RAMI 4.0 is the layer with standard functions and concrete applications. In comparison, IIRA Functional is a viewpoint that decomposes functionally into functional domains [61]. Mapping from IIRA functional domains to RAMI 4.0 Layers is represented in Table 2.1.

Table 2.1. Mapping IIRA Functional Domains to RAMI 4.0 Layers adapted from [61].

IIRA Functional Domains	**RAMI 4.0 Layers**
Physical Systems	Physical aspects of an Assets Layer
Control Domain	Integration Layer Functional Layer
Connectivity and Distributed Data	Communication Layer
Information Domain	Information Layer
Operations and Application Domain	Functional Layer
Business Domain	Business Layer

Some standards and protocols to provide communication capabilities, defined in the Industrial Internet Connectivity Framework (IICF) developed by the IIC, extend the IIRA to an open connectivity reference architecture. IICF is composed of five layers from bottom to top: Physical and Link "TSN/Ethernet (802.1, 802.3), Wireless PAN (802.15), Wireless LAN (802.11 Wi-Fi), Wireless 2G/3G/LTE (3GPP), Wireless Wide Area (802.16)", Network "Internet Protocol (IP)", Transport "TCP, UDP, DDSI-RTPS, CoAP, MQTT, HTTP, OPC UA Bin", Framework "DDS, oneM2M, Web Services, OPC UA" [62]. IICF presents connectivity interoperability device-to-device, device-to-application, and application-to-application.

Mapping RAMI 4.0 communication layer to the ISO/OSI layer, the Application, Presentation, and Session can be achieved by using the HTTP(S), MQTT, or OPC UA (Server-Client and Pub/Sub) [63]. It is possible to conclude that even HTTP, MQTT, and OPC AU can be used to communicate in RAMI 4.0 and IIRA. Also, these standards can be used in Industry 4.0 communication. RAMI 4.0 communication layer and IICF from IIRA are shown in Figure 2.5.

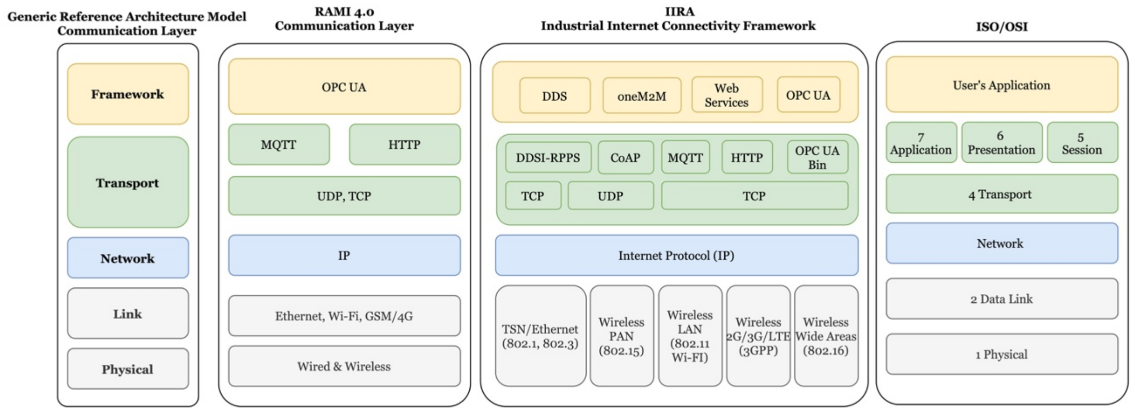


Figure 2.5 RAMI 4.0 communication layer mapped to ISO/OSI layers and Framework and Transport layer of IICF used in IIRA.

2.4 Levels of Interoperability

To interconnect all these concepts, interoperability is necessary to integrate devices and systems inside the IoT, IIoT, Industry 4.0, CPS, and CPPS. The IEEE defines this property as the "degree to which two or more systems, products or components can exchange information and use the information that has been exchanged" [64], i.e., the capacity of two or more systems to exchange the data and use it, implying that two systems understand each other. Interoperability is a fundamental issue in the IoT world. It is estimated that without interoperability, 40% of the benefits of the IoT cannot be obtained [65], [66]. To overcome this issue industry is addressing the IoT interoperability challenges through the standardisation [65].

Interoperability can be divided into levels with different arrangements, depending on the author. In [65] and [67], the authors divided them into five categories:

- Device interoperability is the interoperability between a smart device, such as a Raspberry Pi and a smartphone, enabling the integration and interoperability of heterogeneous devices by a diversity of communication protocols and standards supported by those devices.
- Network interoperability: It is the interoperability of the mechanisms to enable seamless message exchange between different networks. It handles addressing, routing, resource optimisation, security QoS, and mobility support.
- Syntactical interoperability is between formats and data structures that exchange messages between heterogeneous systems. It is necessary to provide an interface

for each schema, such as REST APIs. The message content needs to be serialised to be sent in a format, such as XML or JSON. The sender encodes the message, and the receiver decodes the message. The issue happens when the receiver decoding rules are incompatible with the sender encoding rules.

- Semantic interoperability: It is the interoperability that enables different agents, services, and applications to exchange information unambiguously. By the employment of ontology, the Research Description Framework (RDF), Web Ontology Language (OWL), and JSON-Linked Data (JSON-LD). It can receive data from an API compatible with the data formats, such as JSON, XML, or CSV.
- Platform interoperability: It is the interoperability between the diverse operational systems, programming languages, data structures, architecture access mechanisms and data. Developers need to understand the different APIs and information models for each platform and adapt their applications for each one of these platforms. It enables the cross-domain between the other platforms with heterogenous domains (e.g., health, home, and transport).

Figure 2.6 shows the pyramid based on the interoperability levels [51].

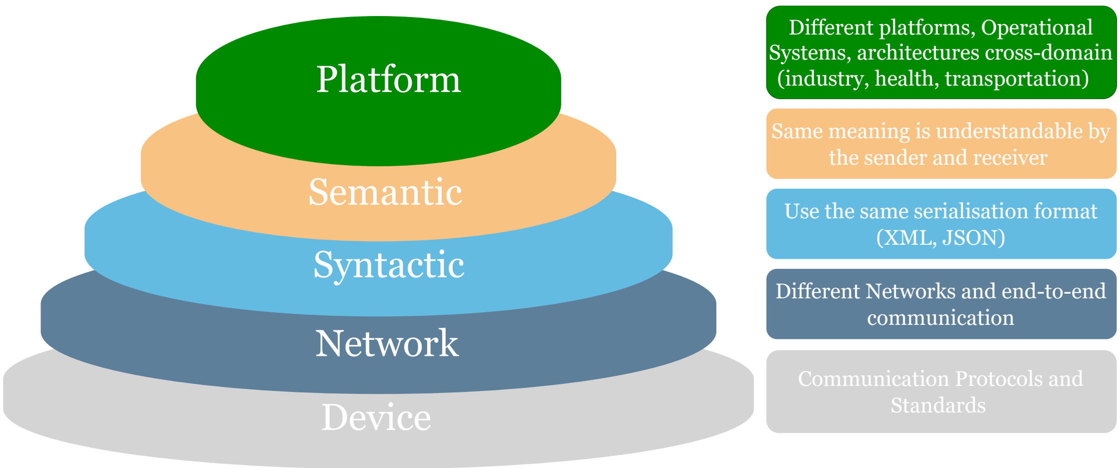


Figure 2.6 Interoperability levels inside Industry 4.0 [51].

Authors in [68], [69] proposed four levels of interoperability. The same syntactical and semantic levels are presented in the paper [65]. The difference is:

Technical interoperability is between physical and software compatibility, including device communication protocols.

Organisational interoperability: It is a requirement for industries to communicate and transfer information between the different systems, infrastructures, or geographic regions and cultures.

Levels of interoperability are derived from a Levels of Conceptual Interoperability Model (LCIM) [69]. Level 0 represents no interoperability, level 1 represents technical interoperability, level 2 represents syntactic interoperability, levels 3 represent semantic interoperability, level 4 represents pragmatic interoperability, level 5 represents dynamic interoperability, and level 6 defines conceptual interoperability.

A solution, such as oneM2M, can provide the semantic level of interoperability inside the IoT and M2M. oneM2M addresses technical specifications by observing the needs of a standard M2M Service Layer, reading various embedded hardware and software connecting a large number of devices in the field of M2M. It is composed of more than 200 companies from different fields, such as industrial automation, and it has the responsibility to approve and maintain the set of technical specifications and technical reports for many areas (e.g. protocols, APIs, standards objects, security and privacy aspects, and others) [70]. The Atlas communication framework is proposed by Khaled and Helal [71]. It provided interoperability with the things that work similarly to different IoT communication languages. The frameworks use the protocol translation that can operate in a cloud or in the thing itself, enabling seamless communication between the devices through well-defined interfaces. The protocols that can be used are the Constrained Application Protocol (CoAP), Representational State Transfer (REST) over HTTP, and Message Queueing Telemetry Transport (MQTT).

Lee et al. [72] proposed the interoperability for IoT protocols based on a bi-directional communication framework that supports different application-layer network protocols (e.g., HTTP, CoAP, and MQTT), integrating the Software-Defined Network (SDN) in the proposed framework. A review of interoperability approaches in the application layer of IoT, based on publishing standards (IPSO, ETSI, IoT-A, AllJoyn, LwM2M, SenML, Weave, Thread, CoAP, and Message Brokers), reference architectures, and frameworks (oneM2m. Vorto, Ponte, OpenIoT, and IoTVity) were presented by Tayur and Suchithra [73]. A semantic gateway to provide interoperability based on a service architecture. The proposed gateway promotes interoperability by translating message protocols (e.g., XMPP, CoAP, and MQTT) and uses the W3Cs Semantic Sensor Network (SSN) ontology for semantic interoperability. It was developed by Desai et al. [74].

The syntactic level of interoperability in the Industrial Internet of Things under the Industry 4.0 context can be possible using the Message Queueing Message Protocol (MQTT). The MQTT was created in 1999 by Andy Stanford-Clark and Arlen Nipper [75]. At this moment, the MQTT is in version five (MQTTv5). It is a publish/subscribe messaging protocol developed to be lightweight, open, and simple with easy implementation [76]. The MQTT is a good option for M2M and IoT contexts with these characteristics.

In [2], Al-Fuqaha et al. alleged that "the MQTT protocol represents an ideal messaging protocol for IoT and M2M communication and can provide routing for small, cheap, low power and low memory devices in vulnerable and low bandwidth networks". MQTT comprises three components a publisher, a subscriber, and a broker. The subscriber subscript into a topic in the broker. Every time the publisher publishes a message to the broker, the broker sends the message to the subscriber. This communication occurs through three Quality of Service (QoS): QoS 0 (at most once), QoS 1 (at least once), and QoS 2 (exactly once).

In the interoperability context, Naik [77] wrote that the MQTT is the most used for M2M and IoT and has the best performance for reliability and QoS. However, It was classified in the last position for interoperability compared with the other protocols (HTTP, CoAP, and AMQP). Interoperability is the biggest issue among all IoT protocols [77]. The resource to provide interoperability will help the communication inside the IoT, IIoT, and Industry 4.0, including CPS and CPPS.

The IEEE 1451 family of standards will be an excellent option for collecting and sharing data from the transducers from the CPPS to the Digital Twin. Composed mainly by the IEEE 1451.0-2007 [78], defines the Transducer Electronic Sheet (TEDS), which describes a transducer stored in some form of electronically readable memory. The TEDS is managed by a Transducer Interface Module (TIM), providing the Interoperability of transducers and systems built to this standard [79]. This standard also defines the communication to access the transducer inside a network employing an API. To collect the data will be used the standard IEEE 21451-2 defines the digital interface for connecting transducers (sensors or actuators) to microprocessors. Standard IEEE 21451-1 was developed to access the network data. This standard presents the Network Capable Application Processor (NCAP), an interface to manage the TIM and the TEDS accessing by these components the physical world employing the transducer [80]. The connection between the NCAP and the TIMs can be wired or wireless, specified by the standard IEEE 1415-5 [81]. IEEE 21451-001 is a recommended practice that defines signal processing algorithms and data structure to define a standardised and universal framework that allows the smart transducer to extract features from signals being generated and measured [82].

There are studies about the IEEE 1451 family standards in the industry context. The standardisation that can be used in the IoT and IIoT infrastructures was presented by Huang et al. [83]. Monte et al. gave an overview of the ISO/IEC/IEEE P2151-001 standard, focusing on the signal processing [84]. Velez et al. proposed methods using the IEEE 1451.1-6 to provide a standard network service over the MQTT [85]. In another paper, Velez et al. [78] developed a compatible IoT network using readily available development platforms. Finally, state-of-the-art about the IEEE 1451 family of sensors with architectures for STIMs and architectures for the NCAPs was presented by Kumar et al. [86]. The IEEE 1451 family of standards has been used to monitor the environment [87], the indoor environment [88], [89], the biomedical sensors and instrumentation [90], the air pollution and gases [91], [92], the smart grid [93], and the water quality [94].

There are tools developed especially for IoT, IIoT, and Industry 4.0. These tools are being developed to perform tests, validate, and promote data exchange between the devices and systems. The tools can be divided into four categories: platform (Table 2.2), middleware and framework (Table 2.3), operational systems (Table 2.4), and hardware (Table 2.5). Figure 2.7 shows all the components categorised.

Table 2.2 Platforms.

Platforms	**Available at**
AWS IoT	https://aws.amazon.com/iot/
Azure IoT	https://azure.microsoft.com/pt-pt/overview/iot/
Cisco IoT Connect	https://www.cisco.com/c/en/us/solutions/internet-of-things/overview.html
DeviceHIVE	https://devicehive.com
Eclipse IoT	https://iot.eclipse.org
Emoncms.org	https://emoncms.org
GE Predix	https://www.ge.com/digital/iiot-platform
IBM Watson	https://www.ibm.com/watson
Inter-IoT	https://inter-iot.eu
Kaa	https://www.kaaproject.org/
M2Mlabs	http://www.m2mlabs.com
macchina.io	https://macchina.io
Node-RED	https://nodered.org
Open Connectivity Foundation	https://openconnectivity.org
open remote	https://www.openremote.io
openHAB	https://www.openhab.org
OpenSCADA	http://oscada.org
Oracle IoT	https://www.oracle.com/internet-of-things/
PlatformIO	https://platformio.org/
rti	https://www.iotone.com/supplier/rti/v49
Shodan	https://www.shodan.io
Siemens MindSphere	https://siemens.mindsphere.io/en
The Things Network (TTN)	https://www.thethingsnetwork.org/
Thinger.io	https://thinger.io
ThingSpeak	https://thingspeak.com/
ThingWorx	https://developer.thingworx.com/en
zetta	https://www.zettajs.org

Table 2.3 Middleware and framework.

Middleware/Framework	Available at
Conera	https://v2com.com/conera-v2com-en/
Fiware	https://www.fiware.org
Global Sensor Network (GSN)	http://gsn.epfl.ch
HiveMQ	https://www.hivemq.com
iotsys	https://code.google.com/archive/p/iotsys/
LinkSmart	https://linksmart.eu/#02-why
oneM2m	https://www.onem2m.org
OPC UA	https://opcfoundation.org/about/opc-technologies/opc-ua/
Ponte	https://www.eclipse.org/projects/archives.php
SIRENA	[95]
SMEPP	[96]
Socrades	http://www.socrades.net/Project/Presentation/default.html
SODA	[97]
ubiROAD	[98]
ubiSOAP	[99]
Virtus	[100]
Vorto	https://www.eclipse.org/vorto/
Wherex	[101]

Table 2.4 Operational Systems.

Operational Systems	Available at
Android Things	https://developer.android.com/things
Apache Mynewt	https://mynewt.apache.org
Azure RTOS	https://azure.microsoft.com/en-us/products/rtos/
Contiki NG	https://www.contiki-ng.org
Linux Lite OS	https://www.linuxliteos.com
Mbed OS 5	https://os.mbed.com
MicroEJ	https://www.microej.com
Mongoose OS	https://mongoose-os.com
Nucleus	https://www.plm.automation.siemens.com/global/en/products/embedded/nucleus-rtos.html
RIOT	https://www.riot-os.org
Tiny OS	http://www.tinyos.net
Ubuntu Core	https://ubuntu.com/core
VxWorks	https://www.windriver.com/products/vxworks
Windows 10 IoT	https://developer.microsoft.com/pt-pt/windows/iot/
Zephyr	https://www.zephyrproject.org
µC/OS	https://www.silabs.com/developers/micrium

Table 2.5 Hardware.

Hardware	Available at
Raspberry Pi	https://www.raspberrypi.org
Adafruit	https://www.arduino.cc
BeagleBone	https://beagleboard.org/bone
Tessel 2	https://tessel.io
Mbed	https://os.mbed.com
NodeMcu	https://www.nodemcu.com/index_en.html
Silicon Labs	https://www.silabs.com
Texas Instruments	https://www.ti.com
Espressif Systems	https://www.espressif.com/en
SparkFun	https://www.sparkfun.com
Particle	https://www.particle.io
Intel boards and Kits	https://www.intel.com/content/www/us/en/products/details/nuc.html
CubieBoard	http://cubieboard.org
Phidgets	https://www.phidgets.com
UDOO	https://www.udoo.org
Siemens SIMATIC	https://new.siemens.com/global/en/products/automation/pc-based.html

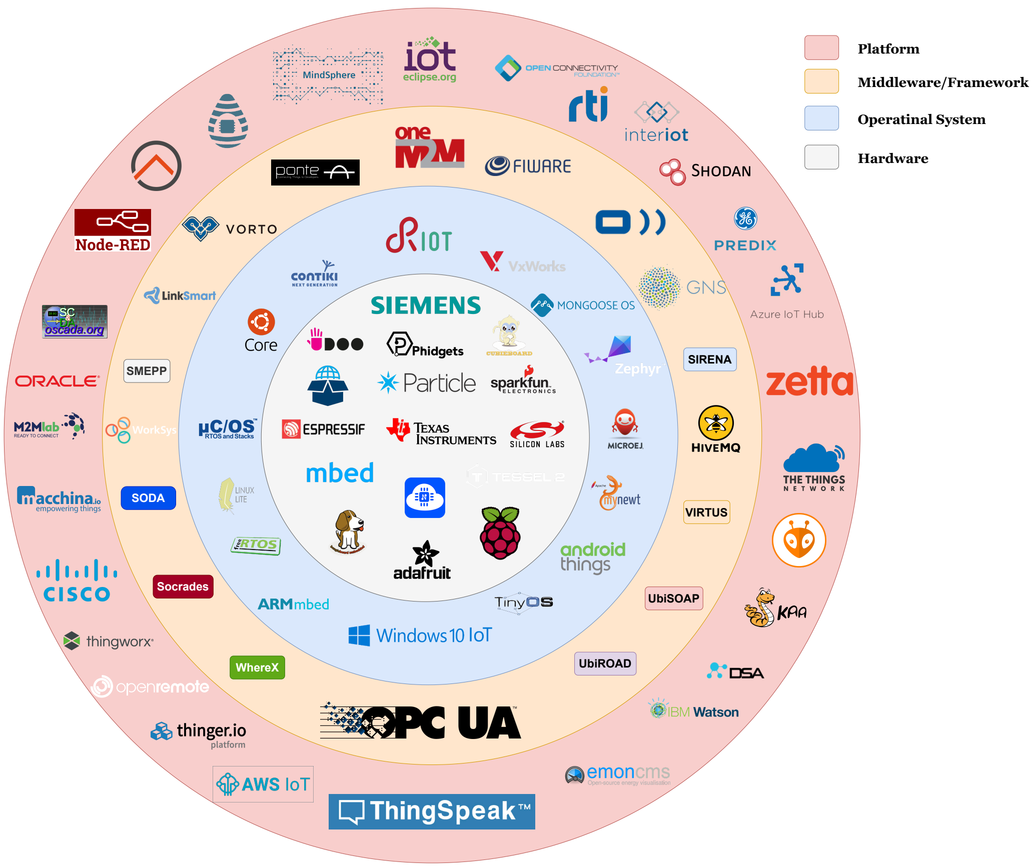


Figure 2.7 Overall development components inside IoT and IIoT world.

2.5 Conclusion

This section presented a survey of the concepts from the IoT until Industry 4.0. Many platforms, middleware, standards, protocols, and components can be utilised inside an industrial context.

It presented the levels of interoperability utilised for the communication required for the industry are the syntactic and semantic levels. The semantic level for communication inside smart manufacturing and Industry 4.0. The reference architecture models utilised inside the industry required semantic levels, normally achieved using a framework, such as OPC UA.

Based on the concept review and requirements presented in this Chapter, the IEEE 1451 will become a better communication choice in the Industry 4.0 environment. It needs to reach the semantic level during the data transmission.

The following Chapters present how the IEEE 1451 reaches the syntactical level of interoperability and present a proposal for the IEEE 1451 standards to reach a semantic level.

References

[1] K. Ashton and others, "That 'internet of things' thing," *RFID J.*, vol. 22, no. 7, pp. 97–114, 2009.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[3] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, 2014.

[4] F. Wortmann and K. Flüchter, "Internet of things," *Bus. Inf. Syst. Eng.*, vol. 57, no. 3, pp. 221–224, 2015.

[5] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: the internet of things architecture, possible applications and key challenges," 2012, pp. 257–260.

[6] A. Whitmore, A. Agarwal, and L. Da Xu, "The Internet of Things—A survey of

topics and trends," *Inf. Syst. Front.*, vol. 17, no. 2, pp. 261–274, 2015.

[7] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," *Bus. Horiz.*, vol. 58, no. 4, pp. 431–440, 2015.

[8] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Comput. networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[9] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, Opportunities, and Directions," *IEEE Trans. Ind. Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[10] P. P. Ray, "A survey on Internet of Things architectures," *J. King Saud Univ. Inf. Sci.*, vol. 30, no. 3, pp. 291–319, 2018.

[11] I. Peña-López and others, "ITU Internet report 2005: the internet of things," 2005.

[12] M. Aazam, S. Zeadally, and K. A. Harras, "Deploying Fog Computing in Industrial Internet of Things and Industry 4.0," *IEEE Trans. Ind. Informatics*, vol. 14, no. 10, pp. 4674–4682, 2018.

[13] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Trans. Ind. informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[14] A. Gilchrist, *Industry 4.0: the industrial internet of things*. Apress, 2016.

[15] X. Ye and S. H. Hong, "Toward Industry 4.0 Components: Insights Into and Implementation of Asset Administration Shells," *IEEE Ind. Electron. Mag.*, vol. 13, no. 1, pp. 13–25, 2019.

[16] J. Qin, Y. Liu, and R. Grosvenor, "A categorical framework of manufacturing for industry 4.0 and beyond," *Procedia Cirp*, vol. 52, pp. 173–178, 2016.

[17] R. Y. Zhong, X. Xu, E. Klotz, and S. T. Newman, "Intelligent manufacturing in the context of industry 4.0: a review," *Engineering*, vol. 3, no. 5, pp. 616–630, 2017.

[18] V. Roblek, M. Meško, and A. Krapež, "A complex view of industry 4.0," *Sage Open*, vol. 6, no. 2, p. 2158244016653987, 2016.

[19] M. Hermann, T. Pentek, and B. Otto, "Design principles for industrie 4.0 scenarios," 2016, pp. 3928–3937.

[20] Y. Lu, "Industry 4.0: A survey on technologies, applications and open research issues," *J. Ind. Inf. Integr.*, vol. 6, pp. 1–10, 2017.

[21] R. M. Pereira, A. L. Szejka, and O. C. Junior, "Towards an information semantic interoperability in smart manufacturing systems: contributions, limitations and applications," *Int. J. Comput. Integr. Manuf.*, vol. 34, no. 4, pp. 422–439, 2021, doi: 10.1080/0951192X.2021.1891571.

[22] "MES - Manufacturing Execution System." http://mescenter.org/en/articles/108-mes-manufacturing-execution-system (accessed Aug. 12, 2022).

[23] A. C. Pereira and F. Romero, "A review of the meanings and the implications of the Industry 4.0 concept," *Procedia Manuf.*, vol. 13, pp. 1206–1214, 2017.

[24] K. Zhou, T. Liu, and L. Zhou, "Industry 4.0: Towards future industrial opportunities and challenges," 2015, pp. 2147–2152.

[25] S. Vaidya, P. Ambad, and S. Bhosle, "Industry 4.0–a glimpse," *Procedia Manuf.*, vol. 20, pp. 233–238, 2018.

[26] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart factory of industry 4.0: Key technologies, application case, and challenges," *IEEE Access*, vol. 6, pp. 6505–6519, 2017.

[27] A. A. Letichevsky, O. O. Letychevskyi, V. G. Skobelev, and V. A. Volkov, "Cyber-Physical Systems," *Cybern. Syst. Anal.*, vol. 53, no. 6, pp. 821–834, 2017, doi: 10.1007/s10559-017-9984-9.

[28] T. Stock and G. Seliger, "Opportunities of sustainable manufacturing in industry 4.0," *Procedia Cirp*, vol. 40, pp. 536–541, 2016.

[29] A. Rojko, "Industry 4.0 concept: background and overview," *Int. J. Interact. Mob. Technol.*, vol. 11, no. 5, pp. 77–90, 2017.

[30] P. Hehenberger, B. Vogel-Heuser, D. Bradley, B. Eynard, T. Tomiyama, and S. Achiche, "Design, Modelling, Simulation and Integration of Cyber Physical

Systems: Methods and Applications," *Comput. Ind.*, vol. 82, pp. 273–289, 2016.

[31] S. Thiede, M. Juraschek, and C. Herrmann, "Implementing cyber-physical production systems in learning factories," *Procedia Cirp*, vol. 54, pp. 7–12, 2016.

[32] J. Herwan, S. Kano, R. Oleg, H. Sawada, and N. Kasashima, "Cyber-physical system architecture for machining production line," 2018, pp. 387–391.

[33] L. Monostori *et al.*, "Cyber-physical systems in manufacturing," *Cirp Ann.*, vol. 65, no. 2, pp. 621–641, 2016.

[34] D. Stock and D. Schel, "Cyber-Physical Production System Fingerprinting," *Procedia CIRP*, vol. 81, pp. 393–398, 2019.

[35] A. Cooper *et al.*, "Privacy considerations for internet protocols," *Internet Archit. Board*, 2013.

[36] T. H.-J. Uhlemann, C. Lehmann, and R. Steinhilper, "The digital twin: Realizing the cyber-physical production system for industry 4.0," *Procedia Cirp*, vol. 61, pp. 335–340, 2017.

[37] E. Marseu, D. Kolberg, M. Birtel, and D. Zühlke, "Interdisciplinary engineering methodology for changeable cyber-physical production systems," *IFAC-PapersOnLine*, vol. 49, no. 31, pp. 85–90, 2016.

[38] L. Monostori, "Cyber-physical production systems: Roots, expectations and R&D challenges," *Procedia Cirp*, vol. 17, pp. 9–13, 2014.

[39] A. Rasheed, O. San, and T. Kvamsdal, "Digital Twin: Values, Challenges and Enablers From a Modeling Perspective," *IEEE Access*, vol. 8, pp. 21980–22012, 2020, doi: 10.1109/ACCESS.2020.2970143.

[40] M. Liu, S. Fang, H. Dong, and C. Xu, "Review of digital twin about concepts, technologies, and industrial applications," *J. Manuf. Syst.*, vol. 58, pp. 346–361, 2021, doi: https://doi.org/10.1016/j.jmsy.2020.06.017.

[41] M. Shafto *et al.*, "Modeling, simulation, information technology \& processing roadmap," *Natl. Aeronaut. Sp. Adm.*, vol. 32, no. 2012, pp. 1–38, 2012.

[42] K. M. Alam and A. El Saddik, "C2PS: A digital twin architecture reference model

for the cloud-based cyber-physical systems," *IEEE Access*, vol. 5, pp. 2050 – 2062, 2017, doi: 10.1109/ACCESS.2017.2657006.

[43] E. Negri, L. Fumagalli, and M. Macchi, "A Review of the Roles of Digital Twin in CPS-based Production Systems," *Procedia Manuf.*, vol. 11, pp. 939–948, 2017, doi: https://doi.org/10.1016/j.promfg.2017.07.198.

[44] C. Semeraro, M. Lezoche, H. Panetto, and M. Dassisti, "Digital twin paradigm: A systematic literature review," *Comput. Ind.*, vol. 130, p. 103469, 2021, doi: https://doi.org/10.1016/j.compind.2021.103469.

[45] I. Errandonea, S. Beltrán, and S. Arrizabalaga, "Digital Twin for maintenance: A literature review," *Comput. Ind.*, vol. 123, p. 103316, 2020, doi: https://doi.org/10.1016/j.compind.2020.103316.

[46] S. Aheleroff, X. Xu, R. Y. Zhong, and Y. Lu, "Digital Twin as a Service (DTaaS) in Industry 4.0: An Architecture Reference Model," *Adv. Eng. Informatics*, vol. 47, p. 101225, 2021, doi: https://doi.org/10.1016/j.aei.2020.101225.

[47] S. R. Newrzella, D. W. Franklin, and S. Haider, "5-Dimension cross-industry digital twin applications model and analysis of digital twin classification terms and models," *IEEE Access*, vol. 9, pp. 131306–131321, 2021.

[48] D.-Y. Jeong *et al.*, "Digital Twin: Technology Evolution Stages and Implementation Layers With Technology Elements," *IEEE Access*, vol. 10, pp. 52609–52620, 2022, doi: 10.1109/ACCESS.2022.3174220.

[49] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital Twin: Enabling Technologies, Challenges and Open Research," *IEEE Access*, vol. 8, pp. 108952–108971, 2020, doi: 10.1109/ACCESS.2020.2998358.

[50] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital Twin in Industry: State-of-the-Art," *IEEE Trans. Ind. Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019, doi: 10.1109/TII.2018.2873186.

[51] H. da Rocha, J. Pereira, R. Abrishambaf, and A. Espirito Santo, "An Interoperable Digital Twin with the IEEE 1451 Standards," *Sensors*, vol. 22, no. 19, 2022, doi: 10.3390/s22197590.

[52] M. A. Pisching, M. A. O. Pessoa, F. Junqueira, D. J. dos Santos Filho, and P. E.

Miyagi, “An Architecture Based on RAMI 4.0 to Discover Equipment to Process Operations Required by Products,” *Comput. Ind. Eng.*, vol. 125, pp. 574–591, 2018.

[53] M. Wollschlaeger *et al.*, “Communication in the Context of Industrie 4.0,” 2019. [Online]. Available: https://www.zvei.org/en/press-media/publikationen/communication-in-the-context-of-industrie-40/.

[54] M. Uslar *et al.*, “Applying the smart grid architecture model for designing and validating system-of-systems in the power and energy domain: A European perspective,” *Energies*, vol. 12, no. 2, p. 258, 2019.

[55] S. Tarkoma, “Internet of Things Strategic Research Agenda (IoT---SRA) Finnish Strategic Centre for Science, Technology, and Innovation: For Information and Communications (ICT) Services, businesses, and technologies,” 2011.

[56] “Industry 4.0 architecture for manufacturing: Reference diagram - IBM Cloud Architecture Center.” https://www.ibm.com/cloud/architecture/architectures/industry-40/reference-architecture (accessed Aug. 03, 2022).

[57] NIST, “Service-Oriented Architectures for Smart Manufacturing Project | NIST.” https://www.nist.gov/programs-projects/service-oriented-architectures-smart-manufacturing-project (accessed Aug. 03, 2022).

[58] Industrial Internet Consortium, “The Industrial Internet of Things Volume G1: Reference Architecture,” 2019. Accessed: Dec. 07, 2019. [Online]. Available: https://www.iiconsortium.org/pdf/IIRA-v1.9.pdf.

[59] M. Hankel and B. Rexroth, “The Reference Architectural Model Industrie 4.0 (RAMI 4.0),” *ZVEI, April*, vol. 410, 2015.

[60] E. Barnstedt *et al.*, “Details of the Asset Administration Shell. Part 1 – The exchange of information between partners in the value chain of Industrie 4.0 (Version 1.0).” 2018.

[61] U. Löwen, R. Neubert, G. Bachmann, S. M. IIC, and H. Munz, “Architecture Alignment and Interoperability,” 2017, [Online]. Available: https://www.iiconsortium.org/pdf/JTG2_Whitepaper_final_20171205.pdf.

[62] B. Gessel *et al.*, "The Industrial Internet of Things Connectivity Framework," 2018. [Online]. Available: https://www.iiconsortium.org/IICF.htm.

[63] "Plattform Industrie 4.0 - Details of the Asset Administration Shell." https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/2018-details-of-the-asset-administration-shell.html (accessed Dec. 07, 2019).

[64] I. S. O. Iso, "IEC/IEEE International Standard-Systems and Software Engineering–Vocabulary," ISO/IEC/IEEE 24765: 2017 (E), 2017.

[65] M. Noura, M. Atiquzzaman, and M. Gaedke, "Interoperability in Internet of Things: Taxonomies and Open Challenges," *Mob. Networks Appl.*, vol. 24, no. 3, pp. 796–809, 2019.

[66] D. C. Y. Vargas and C. E. P. Salvador, "Smart IoT gateway for heterogeneous devices interoperability," *IEEE Lat. Am. Trans.*, vol. 14, no. 8, pp. 3900–3906, 2016.

[67] M. Noura, M. Atiquzzaman, and M. Gaedke, "Interoperability in Internet of Things infrastructure: classification, challenges, and future work," 2017, pp. 11–18.

[68] V. R. Konduru and M. R. Bharamagoudra, "Challenges and solutions of interoperability on IoT: How far have we come in resolving the IoT interoperability issues," 2017, pp. 572–576.

[69] B. Arne *et al.*, "Advancing IoT Platforms Interoperability," 2018.

[70] O. Alliance, *Onem2m: Standards for M2M and the Internet of Things*. 2014.

[71] A. E. Khaled and S. Helal, "Interoperable communication framework for bridging RESTful and topic-based communication in IoT," *Futur. Gener. Comput. Syst.*, vol. 92, pp. 628–643, 2019.

[72] C.-H. Lee, Y.-W. Chang, C.-C. Chuang, and Y. H. Lai, "Interoperability enhancement for Internet of Things protocols based on software-defined network," 2016, pp. 1–2.

[73] V. M. Tayur and R. Suchithra, "Review of interoperability approaches in

application layer of Internet of Things," 2017, pp. 322–326.

[74] P. Desai, A. Sheth, and P. Anantharam, "Semantic Gateway as a Service Architecture for IoT Interoperability," 2015, pp. 313–319.

[75] A. Banks and R. Gupta, "MQTT Version 3.1. 1," *OASIS Stand.*, vol. 29, 2014.

[76] A. Banks, E. Briggs, K. Borgendale, and R. Gupta, "MQTT Version 5.0," *OASIS Stand.*, 2019, [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html.

[77] N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," 2017, pp. 1–7.

[78] I. S. IEEE 1451.0, "IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats," *IEEE Std 1451.0-2007*, pp. 1–335, 2007, doi: 10.1109/IEEESTD.2007.4338161.

[79] "ISO/IEC/IEEE Standard for Information technology – Smart transducer interface for sensors and actuators – Part 2: Transducer to microprocessor communication protocols and Transducer Electronic Data Sheet (TEDS) formats," *ISO/IEC/IEEE 21451-2:2010(E)*, pp. 1–130, 2010, doi: 10.1109/IEEESTD.2010.5668463.

[80] I. 2010 IEEE 21451-1:, "ISO/IEC/IEEE Information technology – Smart transducer interface for sensors and actuators – Part 1: Network Capable Application Processor (NCAP) information model," *ISO/IEC/IEEE 21451-1:2010(E)*, pp. 1–356, 2010, doi: 10.1109/IEEESTD.2010.5668469.

[81] "IEEE Standard for a Smart Transducer Interface for Sensors and Actuators Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats," *IEEE Std 1451.5-2007*, pp. C1-236, 2007, doi: 10.1109/IEEESTD.2007.4346346.

[82] "IEEE Recommended Practice for Signal Treatment Applied to Smart Transducers," *IEEE Std 21451-001-2017*, pp. 1–109, 2017, doi: 10.1109/IEEESTD.2017.8227039.

[83] V. K. L. Huang *et al.*, "Past, present and future trends in industrial electronics

standardization," 2017, pp. 6171–6178.

[84] G. Monte, Z. Liu, F. Abate, V. Paciello, A. Pietrosanto, and V. Huang, "Normalizing transducer signals: An overview of a proposed standard," 2014, pp. 614–619.

[85] J. Velez, R. Trafford, M. Pierce, B. Thomson, E. Jastrzebski, and B. Lau, "Ieee 1451-1-6: Providing common network services over mqtt," 2018, pp. 1–6.

[86] A. Kumar, V. Srivastava, M. K. Singh, and G. P. Hancke, "Current status of the IEEE 1451 standard-based sensor applications," *IEEE Sens. J.*, vol. 15, no. 5, pp. 2505–2513, 2014.

[87] A. Kumar, H. Kim, and G. P. Hancke, "Environmental monitoring systems: A review," *IEEE Sens. J.*, vol. 13, no. 4, pp. 1329–1339, 2012.

[88] A. Kumar, I. P. Singh, and S. K. Sud, "Energy efficient and low-cost indoor environment monitoring system based on the IEEE 1451 standard," *IEEE Sens. J.*, vol. 11, no. 10, pp. 2598–2610, 2011.

[89] A. Kumar and G. P. Hancke, "An energy-efficient smart comfort sensing system based on the IEEE 1451 standard for green buildings," *IEEE Sens. J.*, vol. 14, no. 12, pp. 4245–4252, 2014.

[90] R. Morello, "Use of TEDS to improve performances of smart biomedical sensors and instrumentation," *IEEE Sens. J.*, vol. 15, no. 5, pp. 2497–2504, 2014.

[91] N. Kularatna and B. H. Sudantha, "An environmental air pollution monitoring system based on the IEEE 1451 standard for low cost requirements," *IEEE Sens. J.*, vol. 8, no. 4, pp. 415–422, 2008.

[92] A. Kumar and G. P. Hancke, "Energy efficient environment monitoring system based on the IEEE 802.15. 4 standard for low cost requirements," *IEEE Sens. J.*, vol. 14, no. 8, pp. 2557–2566, 2014.

[93] V. Paciello, A. Pietrosanto, and P. Sommella, "Smart sensors for demand response," *IEEE Sens. J.*, vol. 17, no. 23, pp. 7611–7620, 2017.

[94] F. Adamo, F. Attivissimo, C. G. C. Carducci, and A. M. L. Lanzolla, "A smart sensor network for sea water quality monitoring," *IEEE Sens. J.*, vol. 15, no. 5, pp. 2514–2522, 2014.

[95] H. Bohn, A. Bobek, and F. Golatowski, “SIRENA - Service Infrastructure for Real-time Embedded Networked Devices: A service oriented framework for different domains,” in *IEEE ICN/ICONS/MCL*, 2006, vol. 23, p. 43, doi: 10.1109/ICNICONSMCL.2006.196.

[96] R. Caro, D. Garrido, and P. TRON, “SMEPP: A Secure Middleware For Embedded P2P,” 2009.

[97] M. Wagner, D. Zöbel, and A. Meroth, “SODA: Service-Oriented Architecture for Runtime Adaptive Driver Assistance Systems,” in *2014 IEEE 17th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing*, Jun. 2014, pp. 150–157, doi: 10.1109/ISORC.2014.15.

[98] V. Terziyan, O. Kaykova, and D. Zhovtobryukh, “UbiRoad: Semantic Middleware for Context-Aware Smart Road Environments,” in *2010 Fifth International Conference on Internet and Web Applications and Services*, May 2010, pp. 295–302, doi: 10.1109/ICIW.2010.50.

[99] P. Raverdy, V. Issarny, and M. Caporuscio, “ubiSOAP: A Service-Oriented Middleware for Ubiquitous Networking,” *IEEE Trans. Serv. Comput.*, vol. 5, no. 01, pp. 86–98, Jan. 2012, doi: 10.1109/TSC.2010.60.

[100] D. Conzon, T. Bolognesi, P. Brizzi, A. Lotito, R. Tomasi, and M. A. Spirito, “The VIRTUS Middleware: An XMPP Based Architecture for Secure IoT Communications,” in *21st International Conference on Computer Communications and Networks (ICCCN)*, 2012, pp. 1–6, doi: 10.1109/ICCCN.2012.6289309.

[101] A. Puliafito *et al.*, “Making the Internet of Things a Reality: The WhereX Solution,” 2010, pp. 99–108.

Chapter 3

Syntactic Interoperable Model for the Industrial Internet of Things and Industry 4.0

3.1 Introduction

This Chapter presents the syntactic level of interoperability between two standards, the IEEE 1451 family of standards and the IEC 61499 standard. Two papers are directly related to this Chapter:

Integrating the IEEE 1451 and IEC 61499 Standards with the Industrial Internet Reference Architecture

H. da Rocha, R. Abrishambaf, J. Pereira, and A. Espirito Santo, *Sensors*, vol. 22, no. 4, p. 1495, Feb. 2022, doi: 10.3390/s22041495.

IEC 61499 and IEEE 1451 for Distributed Control and Measurement Systems

R. Abrishambaf, H. Da Rocha and A. Espírito-Santo, *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, Toronto, ON, Canada, 2021, pp. 1-6, doi: 10.1109/IECON48115.2021.9589997.

3.2 Syntactic Level of Interoperability

Interoperability is one of the biggest challenges to be addressed inside the Industrial Internet of Things and Industry 4.0 [1]. There are many definitions of interoperability. The IEEE defines it as the "degree to which two or more systems, products or components can exchange information and use the information that has been exchanged" [2]. The standardisation can address the industry's interoperability issue [3]. Inside the industry, the challenge is interconnecting heterogeneous devices and communication standards through web systems. It is increased when the devices start generating a quantity of random and dump data inside the shop floor [4].

Syntactic interoperability is reached using the same format and data structure to exchange messages between heterogeneous systems. It is necessary to provide an interface for each schema, such as REST APIs. The message content needs to be serialised to be sent in a format (e.g., XML or JSON) through a message protocol (e.g., MQTT, HTTP, XMPP). The sender encodes the message, and the receiver decodes the message. The issue happens when the receiver decoding rules are incompatible with the sender encoding rules [3], [4]. Syntactic interoperability can be achieved using the publish/subscribe pattern implemented in the MQTT. This messaging protocol was developed to be lightweight, open, and simple with easy implementation [5]. The MQTT is a good option for M2M and IoT contexts with these characteristics. In [6], the author alleged that "the MQTT protocol represents an ideal messaging protocol for IoT and M2M communication and can provide routing for small, cheap, low power and low memory devices in vulnerable and low bandwidth networks".

As described in IEEE 21451.1 [7], the "communication models define the syntax and semantics of the software interface between application objects and a communication network". However, it also refers that "The IEEE 1451.1 standard does not specify any network transfer syntax or protocol" [7]. However, at this point, the IEEE 1451.1 standard achieves the syntactic interoperability [8], [9].

Saito and Nishi developed a conversion method that translates messages between the MQTT, CoAP, XMPP, and SMTP using the IEEE 1451 standards [10]. Cruz et al. developed a systematic review of middleware used inside the IoT and proposed a new reference architecture for the IoT environment [11]. Gleim et al. [12] presented a conceptual data layer model that enables interoperability across domains, organisations and enterprises focused on the Internet of Production (IoP). A syntactical level of interoperability was achieved using the IEEE 1451 and IEC 61499 standards in [13], [14].

Roffia et al. developed using the publish/subscribe paradigm, a semantic model inspired by the Smart-M3 concept at the information interoperability level [15]. A service-orientated protocol that works on-demand as a translator between the protocols CoAP, HTTP, and MQTT was developed by Derhamy et al. [16]. An architecture to allow interoperability between multiple platforms and standards was introduced by An et al. [17]. It promotes the interoperability between FIWARE and oneM2M.

Syntactic and semantic levels of interoperability were implemented in a framework described in [18]. This framework focuses on device discovery and interaction. Another middleware proposed by Žarko et al. in [19], SymbIoT, was developed to achieve syntactic and semantic levels of interoperability.

To promote the syntactic level of interoperability inside an industrial environment was chosen to interoperate the IEEE 1451 family of standards with the IEC 61499 standard.

3.3 The IEEE 1451 Family of Standards

The IEEE 1451 started in 1997 as IEEE 1451.2-1997 IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats. The goal of this standard was to create a digital interface to connect transducers to microprocessors. Enabling plug-and-play at the transducer (sensors and actuators) level enables and simplifies the creation of smart transducer networks and supports multiple networks. It introduces the TEDS concept of storing information about the transducer. The electronic interface to read and write a function to access the TEDS is developed in the Smart Transducer Interface Module (STIM). This standard does not standardise the Network Capable Application Processor (NCAP) [20].

The NCAP information model was standardised in the IEEE 1451.1-1999 [21]. First, it is connected via wire to the STIM using the Transducer Independent Interface (TII) to read the sensor using a trigger event and write to an actuator [20]. The NCAP defines an object model based on a network-neutral interface for communicating the network, sensor, and actuators.

In 2003 was introduced the IEEE 1451.3-2003 Digital Communication and TEDS Formats for Distributed Multidrop Systems. This standard provided a digital interface to connect multiple physically separated transducers to a processor by a pair of wires. More TEDS formats were introduced, synchronous and asynchronous communication was supported, and channel identification and time synchronisation protocols were presented using the base from the IEEE 1451.1 ad IEEE 1451.2 standards [22].

The IEEE 1451.4-2004 was developed in 2004 to define the protocol and interface allowing digital communication between the analogue transducers and the NCAP [23]. It was built to simplify the smart mixed-mode transducer development by independently defining hardware and software blocks inside a control network. It describes the Mixed-Mode Interface (MMI) and the TEDS. MMI is the master-slave, multidrop, serial connection containing the mechanism to detect the transducer's hot-swap, separate digital and analogue connections, and access the TEDS. The Template Description Language (TDL) inside the Transducer Block is defined in the standard. The TDL is a scripted and tagged language used to describe the functionality (e.g., read, write, configure, and check status) of the IEEE 1451.4 Transducer placed inside the NCAP to access, encode, and decode TEDS. The communication specified in this standard was the 2-conductor signal.

To handle the wireless interface inside the IEEE 1451 was developed the IEEE 1451.5-2007 Wireless Communication Protocols and TEDS formats [24]. It defines the wireless TIM (WTIM) communication radio-specific protocols and the TEDS for these radios. The WTIM is connected to the NCAP by the supported protocols IEEE 802.11 (Wi-Fi), IEEE 802.15.4 (LR-WPAN), Bluetooth, and ZigBee. The WTIM contain the signal conditioning, analogue-to-digital and/or digital-to-analogue and can have the transducer placed on it. It defines the PHY TEDS as a mandatory TEDS on the IEEE 1451.0 family of standards [25].

In 2007, the IEEE 1451 standards were revised, and some standards were restructured. The IEEE 1451.0 Common Functions, Communication Protocols, and TEDS formats have become the leading standard for developing IEEE 1451. However, it does not fully comply with the IEEE 1451.1, IEEE 1451.2, and IEEE 1451.3 [25]. In this standard, the Transducer Interface Module (TIM) and the NCAP were introduced to the transducers of the Network. A TIM has the interface, signal conditioning as the IEEE 1451.5, and signal conversion. A TIM can have many transducers connected to it, attached to the NCAP, with the hardware and software acting as a gateway between the TIM and the user network. A new transducer was introduced the event sensor. An event sensor detects the state change in the physical world and acts on its measurement. The focus of this standard is to provide the basic functions to control and manage smart transducers, communication protocols, and TEDS formats. An implementation-independent Application Programming Interface (API) was created to allow communication between the NCAP and the user network or the Internet. Specified an implementation-independent Application Programming Interface (API) was made.

The Network Capable Network Processor (NCAP), standardised by IEEE 21451-2010, gives the network access to the TIM. The NCAP controls one or several TIMs and works as a gateway between the user network and a TIM. The services provided by the NCAP allow the discovery of new Transducer Interface Modules (TIMs), notify events detected by sensors and manage transducers. The structure of the IEEE 1451 family standard is described by standards IEEE 1451.0 and IEEE 21451.1. Communication with TIMs can be achieved wirelessly, complying with the IEEE 1451.5 standard, using one of the following protocols: Wi-Fi, 6LoWPAN, Zigbee, and Bluetooth [24]. The IEEE 1451.2

standard defines wired connections. Various protocols support the NCAP's connection to the user's network: TCP/UDP, HTTP, XMPP, SNMP, and MQTT [7].

Following the IEEE 1451, in 2010 was developed the IEEE 1451.7-2010 Transducer to Radio Frequency Identification (RFID) Systems Communication Protocols and TEDS formats [26]. It describes the communication methods, data formats, and TEDS for sensing networks to work with RFID, using the communication protocols, command structures, TEDS, and data transfer defined in IEEE 1451.0 standard. There is no specific air interface protocol defined. The standard is agnostic on this. The following year, it was revised in 2011 as [27].

To define the signal processing and data structured inside a TIM was proposed the IEEE 1451-001-2017 Recommended Practice for Signal Treatment Applied to Smart Transducers [28]. It focuses on sharing and inferring signal and state information of an instrumentation or control system by the transducer signal treatment services definition, allowing the transducer to extract features from the signal being generated and measured to generate knowledge based on the transducer's information about the signal acquisition and processing by the algorithms introduced in the standard.

Outside this scope, the IEEE p1451.99 Standard for Harmonisation of Internet of Things (IoT) Devices and Systems [29]. It defines a method for data sharing, interoperability, and security over a network based on the XMPP protocol. It relies on the infrastructure components, or bridges, in a real-time standardised interface to convert IoT protocols used in the industrial environment using the IEEE 1451 family of standards.

To complete the IEEE 1451 family of standards, there are standards in development inside the NCAP: IEEE P21451.1-1 to handle TCP/UDP communications, IEEE P21451.1-2 to define HTTP communication, IEEE P2151.1-3 for web services, IEEE P21451.1-4 XMPP protocol, IEEE P2141.5 SNMP protocol, and the IEEE P1451.1-6 for the MQTT communication. The IEEE P1451-002 Recommended Practice for Low-Power Smart Transducer Application is being developed for the TIM. It is important to mention that IEEE 1451.0 is under review as IEEE P1451.0 [30]. Figure 3.1 shows all the standards that compose the IEEE 1451 family of standards [31].

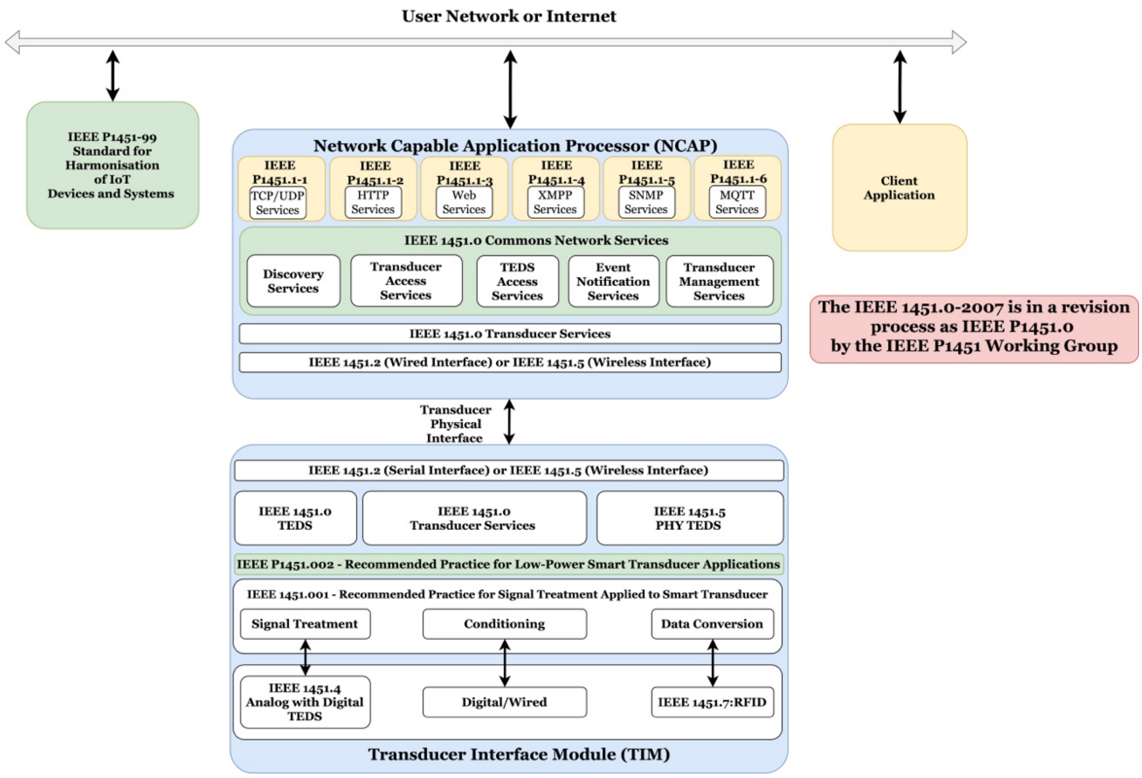


Figure 3.1 The IEEE 1451 family of standards.

The IEEE 1451 family of standards offers a set of characteristics that meet the Industry 4.0 requirements [32]. This family of standards aims to ensure the access of transducers to the communication network to support the exchange of information between the network elements and allows the manufacturers to build elements for an interoperable system.

This thesis unifies the IEEE 1451.0, the IEEE 21451.1, and the IEEE 1451.5 standards to provide syntactic and semantic levels of interoperability. The IEEE 1451.0 TIM development is outside of the scope of this thesis. The focus is on the interoperability provided by the NCAP. The next Section will discuss IEEE 1451.0 and IEEE 21451.1 standards in more detail. The IEEE 1451.0-2007 was used to develop this thesis with a compliant version of the IEEE P1451.1-2010.

3.3.1 The IEEE 1451.0 Standard

The IEEE 1451.0 was developed utilising the concepts presented by the standards published before this version. Focusing on the management of transducers defining the interfaces for signal conversion and communication through a defined API. This standard is defined the functional specification, data types, message structure, and commands between a TIM and the NCAP.

The functional specifications for the IEEE 1415 reference model are the user's network, the network access model through the NCAP, the transducer services interfaces, the NCAP IEEE 1451.0 services, the module communication interface, the NCAP communication module, the physical interface, the TIM communication module, the TIM services, transducer API, the signal conditioner and data conversion functions, the transducer analogue interface, the transducer, and the TEDS definition.

The IEEE 1451 family of standards define its own datatypes they are Int8, Int16, Int32, UInt8, UInt16, UInt32, Float32, Float64, Boolean, UnsignedByte, Octet, String, Int8Array, Int16Array, Int32Array, UInt8Array, UInt16Array, UInt32Array, Float32Array, Float64Array, OctetArray, BooleanArray, StringArray, TimeInstateType (composed by secs in Uint32, nsecons in Uint32), TimeDurantionType (composed by secs in Uint32, nsecons in Uint32, and epoch in Uint8), TimeInstateArray, and TimeDurantionArray.

The message structure defined by the standard is based on an octets array illustrated in Figure 3.2. The TransducerChannel number for the destination message has 16-bit, the Command Class and Command Function are defined inside the command structure, the Length corresponds to the number of octets that the command-dependent will utilise, and the command-dependent has the information specified by the command.

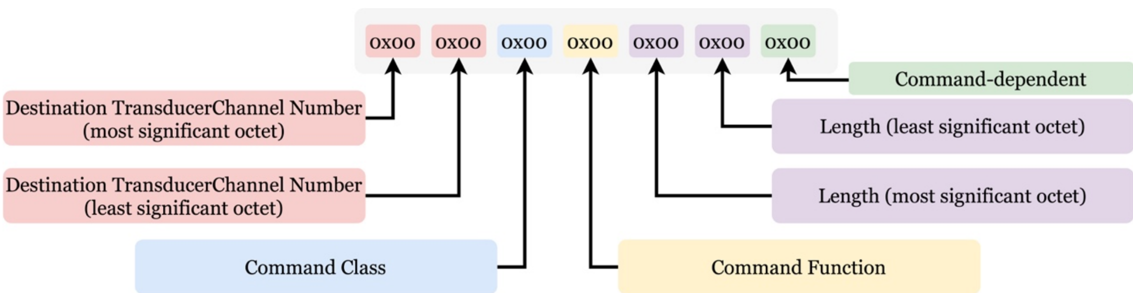


Figure 3.2 Command message structure.

The replay of the command is shown in Figure 3.3. The first octet refers to the success or failure during the execution of the command, zero means fail, and one means success. The length corresponds to the length in octets for the message, and replay-dependent octets contain the content answer to the command.

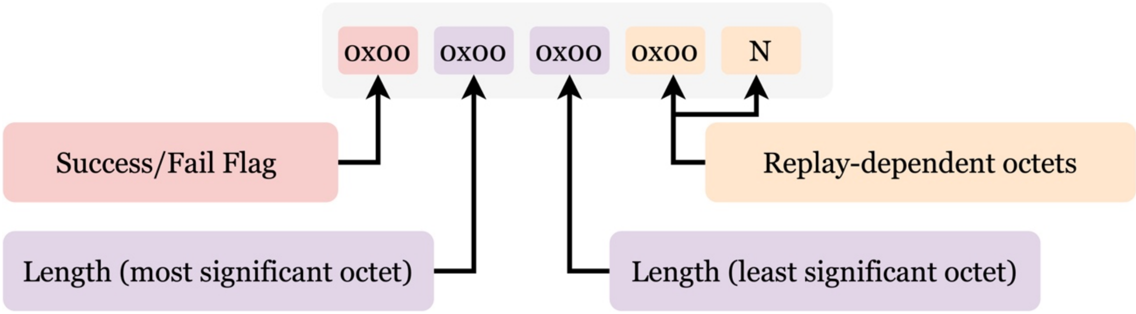
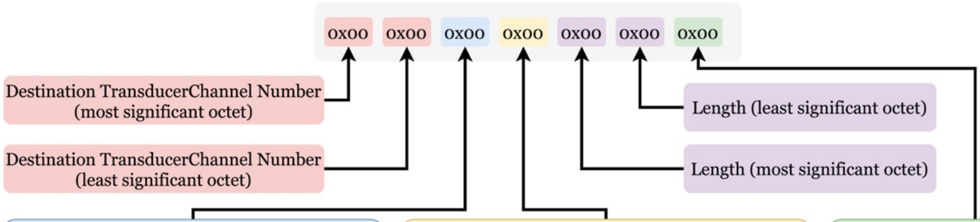


Figure 3.3 Replay message structure.

Inside the message are the commands. The commands are divided into standard and manufacture-dependent, where the most significant octet define the class, and the least defines the specific command. The TIM replays the command when the replay is required or if the status event is enabled. The common commands are shown in Figure 3.4.



Command Classes		
CmdClassId	Attribute name	Category
0	Reserved	Reserved
1	CommondCmd	Commands common to the TIM and Transducer
2	XdcrIdle	Transducer idle state
3	XdcrOperate	Transducer operating state
4	XdcrEither	Transducer either idle or operating state
5	TIMsleep	Sleep state
6	TIMActive	TIM active state commands
7	AnyState	Any State
8-127	ReservedClass	Reserved
128-255	ClassN	Open to manufacturers - N = class number

Common Commands		
CmdFunctionId	Command	Replay expected
0	Reserved	---
1	Query TEDS	Yes
2	Read TEDS segment	Yes
3	Write TEDS segment	No
4	Update TEDS	Yes
5	Run self-test	No
6	Write service request mask	No
7	Read service request mask	Yes
8	Read status-event register	Yes
9	Read status-condition register	Yes
10	Clear status-event register	No
11	Write status-event protocol state	No
12	Read status-event protocol state	Yes
13-127	Reserved	---
128-255	Open for manufacturers	---

TEDS Access Codes		
TEDS Access Code	TEDS Name Attribute	TEDS
0	---	Reseved
1	MetaTEDS	Meta TEDS
2	MetaIdTEDS	Meta Identification TEDS
3	ChanTEDS	TransducerChannel TEDS
4	ChanIdTEDS	TransducerChannel Identification TEDS
5	CalTEDS	Calibration TEDS
6	CalIdTEDS	Calibration Identification TEDS
7	EUASTEDS	End User's Application Specific TEDS
8	FreqRespTEDS	Frequency Response TEDS
9	TransferTEDS	Transfer Function TEDS
10	CommandTEDS	Command TEDS
11	TitleTEDS	Location and Title TEDS
12	XdcrNameTEDS	User's Transducer Name TEDS
13	PHYTEDS	Physical TEDS
14	GeoLocTEDS	Geographic Location TEDS
15	Units Extension TEDS	Units Extent TEDS
16-127	---	Reserved
128-255	---	Manufacturer defined TEDS

Figure 3.4 Common commands to the TIM and TransducerChannel.

The Query TEDS command is utilised to request information to read and write content in the TEDS. The TEDS segment read the TEDS content to the NCAP, returning the raw TEDS block. Write TEDS segment writes part of the TEDS by a raw TEDS block, and the

Update TEDS copies the information in a written TEDS into the non-volatile memory. These are the most used command to work with the TEDS and was utilised in this thesis.

The main contribution of the IEEE 1451.0 standard is the TEDS. TEDS are blocks of information developed to store in non-volatile memory inside of a TIM or outside the "virtual" (VTEDS). It is generally immutable after the user or the manufacturer it provided. However, it can support the Transducer Channels to change the TEDS contents during its operation. The TEDS describes the TIM, the transducer, the signal conditioner, and the data converters. The TEDS has a general format based on the TEDS Length, Data Block, and Checksum. The TEDS Length represents the total octets in the TEDS Data Block sum with the two octets for the Checksum. The TEDS Data Block contains the information defined by each specific TEDS. It utilises the Type/Length/Value (TLV) format. The field Type identifies the field of the TEDS, the Length field is the number of octets representing the Value length, and the Values field has the information about the TEDS, as shown in Figure 3.5.

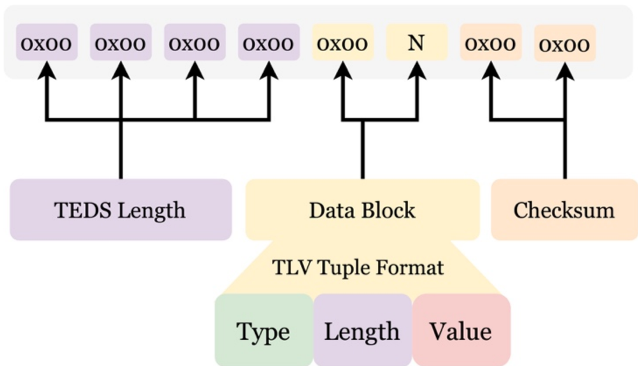
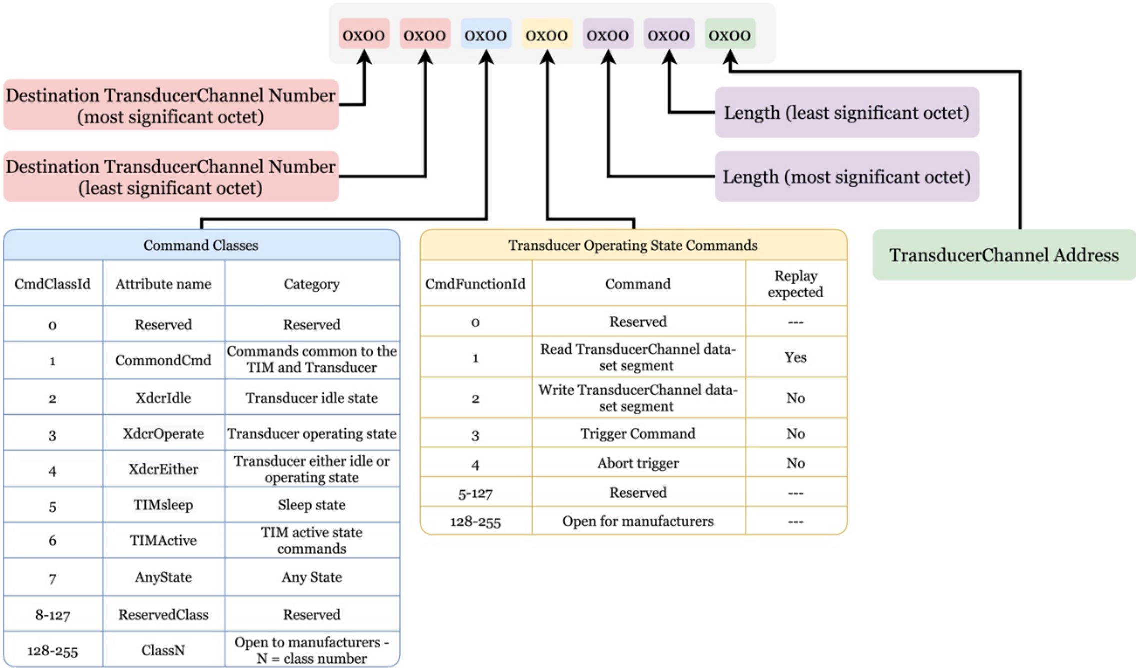


Figure 3.5 General format for TEDS.

The command to access the TransducerChannel is presented in Figure 3.6. The commands are sent for the TransducerChannel address destination when the TIM and the TransducerChannel are in the operation state. The Read TransducerChannel data segment reads the data set inside a transducer using the same octet format as the TEDS. The replay is the same as presented in Figure 3.3.



Command Classes

CmdClassId	Attribute name	Category
0	Reserved	Reserved
1	CommondCmd	Commands common to the TIM and Transducer
2	XdcrIdle	Transducer idle state
3	XdcrOperate	Transducer operating state
4	XdcrEither	Transducer either idle or operating state
5	TIMsleep	Sleep state
6	TIMActive	TIM active state commands
7	AnyState	Any State
8-127	ReservedClass	Reserved
128-255	ClassN	Open to manufacturers - N = class number

Transducer Operating State Commands

CmdFunctionId	Command	Replay expected
0	Reserved	---
1	Read TransducerChannel data-set segment	Yes
2	Write TransducerChannel data-set segment	No
3	Trigger Command	No
4	Abort trigger	No
5-127	Reserved	---
128-255	Open for manufacturers	---

Figure 3.6 Transducer Operating State Commands.

Four mandatory TEDS are defined in the IEEE 1451.0: Meta TEDS, TransducerChannel TEDS, User’s Transducer Name TEDS, and PHY TEDS.

The Meta TEDS defines the communication timing parameter (e.g., time-out values) to communicate the TIM with the NCAP and the relationship between the transducers connected to the TIM. The Meta TEDS Data Block is presented in Figure 3.7.

Meta TEDS Data Block			
Field type	Field Name	Description	Data Type
---		Length	UInt32
0-2	---	Reserved	--
3	TEDSID	TEDS Identification Hearder	UInt8
4	UUID	Globally Unique Identifier is unique and placed inside the Meta TEDS	UUID
5-9	---	Reserved	---
10	OholfOff	Operational time-out is the time in seconds after the operation to present a failure if it is achieved to advise the NCAP for the time-out	Float32
11	SholOff	Slow-access time-out is the time in seconds that represents a long time-out to advise the NCAP about the timeout a a long process	Float32
12	TestTime	Self-Test time in seconds	Float32
13	MaxChan	Number of implemented TransducerChannels starting in one	UInt16
14	CGroup	Identify transducers that operate other transducers	---
15	VGroup	Identifies the relationship between the data sets and a transducer model	---
16	GeoLoc	Implements for dynamic location information	---
17	Proxies	Used to combine the outputs of multiples sensors of the input to multiple actuators in a single structure	---
18-19	---	Reserved	---
25-127	---	Reserved	---
128-255	---	Open to manufacturers	---
---		Checksum	---

Control Group and Vector Group Data Block			
Field type	Field Name	Description	Data Type
20	GrpType	Enumeration for Control Groups or Vector Groups	UInt32
21	MenList	List of TransducerChannels numbers inside the ControlGroup or Vector Group	UInt16Array

Geographic Location Data Block			
Field type	Field Name	Description	Data Type
24	LocEnum	Enumeration for the geographic location types	UInt8
20	GrpType	Enumeration for Control Groups or Vector Groups	UInt32
21	MenList	List of TransducerChannels numbers inside the ControlGroup	UInt16Array

Proxies Data Block			
Field type	Field Name	Description	Data Type
22	ChanNum	TransducerChannel number to be addressed by the proxy	UInt16
23	Organiz	Enumeration to combine the transducer's data set	UInt8
21	MenList	List of TransducerChannels numbers inside the Proxies	UInt16Array

Figure 3.7 Meta TEDS Data Block.

The information about a specific transducer is stored in the TransducerChannel TEDS, such as physical parameters, the range of transducer operation, characteristics of the input and output operational modes, and the taming information, as shown in Figure 3.8 and Figure 3.9.

TransducerChannel TEDS Data Block			
Field type	Field Name	Description	Data Type
---		Length	UInt32
0-2	---	Reserved	---
3	TEDSID	TEDS Identification Hearder	UInt8
4-9	---	Reserved	---
10	CalKey	Calibration Key	UInt8
11	ChanType	TransducerChannel type key (sensor, actuator, or event sensor)	UInt8
12	PhyUnits	Physical Units enumeration	UNITS
13	LowLimit	Design operational low range limit interpreted as physical units	Float32
14	HiLimit	Design operational high range limit interpreted as physical units	Float32
15	OError	Worts case uncertain	Float32
16	SelfTest	Self-test key	UInt8
17	MRange	Multi-range capability identifies if the transducer can operate in different ranges	UInt8
18	Sample	For reading and writing TransducerChannels data-set commands	---
19	DataSet	For retrieving the data-set information	---
20	UpdateT	TransducerChannel update time between the trigger event and the first data-set sample get from the TransducerChannel	Float32
21	WSetupT	TransducerChannel write setup time for end the write data-set and start a trigger	Float32
22	RSetupT	TransducerChannel read setup time is the maximum time to receive a trigger and make the data available for a read	Float32
23	SPeriod	TransducerChannel sampling mode to read or write	Float32
24	WarmUpT	TransducerChannel warm-up time defines the TransducerChannel stabilises its performance	Float32
25	RDelayT	TransducerChannel read delay time defines the minimum time between a read TransducerChannel data segment and the state of data transmission	Float32
26	TestTIme	TransducerChannel self-test time requirements	Float32
27	TestSrc	Source for the time sample based on the time of day that the sample was measured	UInt8
28	InPropDl	Incoming propagation delay through the data transport logic is the time delay from the physical layer and the communication protocols for a sample acquisition	Float32
29	OutPropD	Outcoming propagation delay through the data transport logic is the time delay between the last request from the communication protocols and the message transmitted	Float32
30	TSError	Trigger-to-sample delay uncertain is the delay between the trigger and the sample acquisition	Float32
31	Sampling	Sampling attribute	---
32	DataXmi	Data transmission attribute	UInt8
33	Buffered	Buffer information enumeration	UInt8
34	EnfOfSet	End-of-data-set operation attribute enumeration	UInt8
35	EdgeRpt	Edge-to-report attribute enumeration	UInt8
36	ActHalt	Actuator-halt attribute enumeration determines the actuator's behaviour after it receives a command	UInt8
37	Direction	Sensitive direction measures the physical phenomena in tree-dimensional spatial properties	Float32
38	DAngles	Direction Angles determine the measurement of the reference plane and direction on the TransducerChannel by the manufacturer	Float32
39	ESOption	Event Sensor options	UInt8
61-217	---	Reserved	---
128-255	---	Open to manufactures	---
---	---	Checksum	---

Figure 3.8 TransducerChannel TEDS Data Block part 1.

Physical Units Data Block			
Field type	Field Name	Description	Data Type
50	UnitType	Physical units interpretation enumeration	UInt8
51	Radians	The exponent for Radians	UInt8
52	SterRad	The exponent for Steradians	UInt8
53	Meters	The exponent for Meters	UInt8
54	Kilogram	The exponent for Kilogram	UInt8
55	Seconds	The exponent for Seconds	UInt8
56	Amperes	The exponent for Amperes	UInt8
57	Kelvins	The exponent for Kelvins	UInt8
58	Modes	The exponent for Modes	UInt8
59	Candelas	The exponent for Candelas	UInt8
60	UnitsExt	TEDS access code for units is text-based extension	UInt8

Sample Data Block			
Field type	Field Name	Description	Data Type
40	DatModel	The data model used for a read and write Transducer data-set segment commands	UInt8
41	ModLenth	Data model length	UInt8
42	SigBits	Model significant bits	UInt16

Data Set Data Block			
Field type	Field Name	Description	Data Type
43	Repeats	Maximum data repetitions for a trigger	UInt16
44	SOrigin	Series origin for the case that the data repetition counter is greater than zero	Float32
45	StepSize	Series increment determines the minimum space between the values measured	Float32
46	SUnits	Series units utilise the Physical Units Data Block	UNITS
47	PreTrigg	Maximum pre-trigger samples repetitions for TransducerChannel values sampled and stored	UInt16

Sample Data Block			
Field type	Field Name	Description	Data Type
48	SampMode	Sampling mode capability enumeration	UInt8
49	SDefault	Default sampling mode enumeration	UInt8

Figure 3.9 TransducerChannel TEDS Data Block part 2.

The User's Transducer Name TEDS is where the user can store the name of the transducer used in the systems. The user or the manufacturer defines the content of this TEDS, as shown in Figure 3.10.

User's Transducer Name TEDS Data Block			
Field type	Field Name	Description	Data Type
---		Length	UInt32
0-2	---	Reserved	---
3	TEDSID	TEDS Identification Header	UInt8
4	Format	Format description of this TEDS	UInt8
5	TCName	TIM or TransducerChannel Name	---
---	---	Checksum	---

Figure 3.10 User's Transducer Name TEDS Data Block.

Defined in the IEEE 1451.5, the PHY TEDS defines the physical connection between the NCAP and the TIM.

Physical TEDS Data Block			
Field type	Field Name	Description	Data Type
---		Length	UInt32
0-2	---	Reserved	---
3	TEDSID	TEDS Identification Header	UInt8
4-9	---	Reserved	---
10	Radio	Radio type	UInt8
11	MaxBPS	Max data throughput in bits per second	UInt32
12	MaxCDDev	Max connected devices simultaneously	UInt16
13	MaxrDev	Max registered devices simultaneously	UInt16
14	Encrypt	Encryption types and keys length	UInt16
15	Authent	Authentication support TRUE or FALSE	Boolean
16	MinKeyL	Minimum key length for security functions	UInt16
17	MaxKeyL	Maximum key length for security functions	UInt16
18	MaxSDU	Maximum payload size for data transfer between devices	UInt16
19	MinALat	Minimum access latency for the first transmission to unconnected devices	UInt32
20	MinTLat	Maximum transferring latency for data transmission	UInt8
21	MaxXact	Maximum simultaneous transaction latency for data transmission	UInt8
22	Battery	Indicates the battery power	UInt8
23	RadioVer	Radio version	UInt16
24	MaxRetry	Maximum retries before disconnect	UInt16
25-31	---	Reserved	---
32-41	---	Bluetooth protocol	---
42-47	---	Reserved	---
48-54	---	802.11 Radios	---
55-63	---	Reserved	---
64-80	---	ZigBee Protocol	---
81-95	---	Reserved	---
96-103	---	6LowPAN Protocol	---
104-127	---	Reserved	---
128-255	---	Open to manufacturers	---
---	---	Checksum	---

Figure 3.11 PHY TEDS Data Block.

It allows the communication between the TIM and TransducerChannels to the outside of the network by the API defined in the IEEE 1451.0 standard. This API focus on being a middleware between the end application that measures and controls the NCAP and TIM. The services provided are the TIM Discovery to discover new TIMs. Transducer access is based on transducer identification. Transducer Management to read and write data on the transducers, and TEDS Management to read, write, and update information on the TEDS. The API is implemented inside the NCAP and uses one of the supported standards (e.g., MQTT, HTTP) to communicate with external applications.

3.3.2 The IEEE 1451.1 Standard

The internal structure of an NCAP is divided by object models with a network-neutral interface. It contains blocks, services, and components that specify the interaction with the transducers [7]. It has two views: physical and logical. The physical view relates to the physical components and considers the sensors and the actuators to make a transducer. This transducer connects over an interface that interfaces with the network, a microprocessor, or a controller. The hardware part of an NCAP includes the microprocessor and the implementation of a physical layer that connects the transducer element. The logical view remains for the logical component that may be grouped into an application to support the NCAP's features. These support components are the operating system, the network protocol, and the transducer's firmware. The application is modelled as Function Blocks combined with Components and Services as defined in IEEE 1451.1. The blocks are divided into NCAP, Function, and Base Transducer. The NCAP Block provides the software interfaces to support network communication and system configuration. The Base Transducer Block provides the software interfaces between the transducer and application functions. The Function Block presented in Figure 3.12 is responsible for encapsulating the application-specific functionalities [7]. An object class defines the mode's implementation.

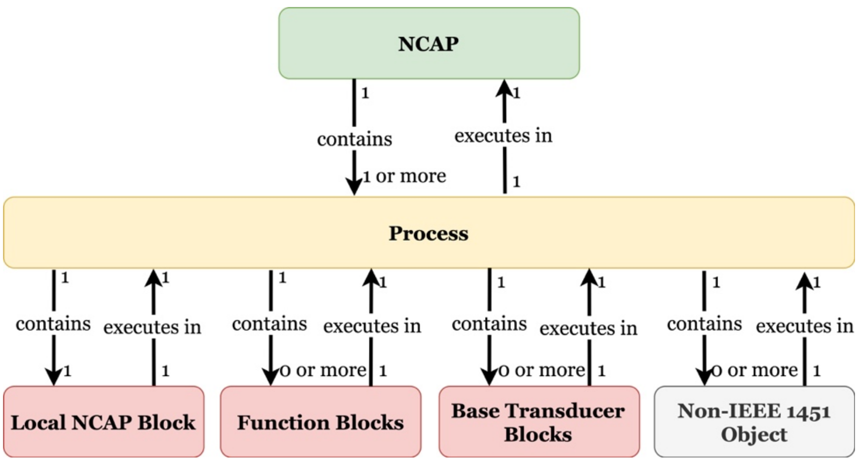


Figure 3.12 IEEE 1451.1 Top Level Relationship between NCAP, Process, and Block Objects [7].

An object class defines the mode's implementation. The IEEE 1451.1 class hierarchy is shown in Figure 3.13. The IEEE1451_NCAPBlock is the main block connecting all the components and services. The NCAP defined in the thesis development was the identification store inside the NCAPBlock (Block Manufacture ID, Block Manufacture Model, Block Software Version, NCAP Manufacture ID, NCAP Model Number, NCAP serial Number, NCAP OS Version) to be compliant with the IEEE 1451.0. The IEEE 1451.1 class hierarchy is presented in Figure 3.13.

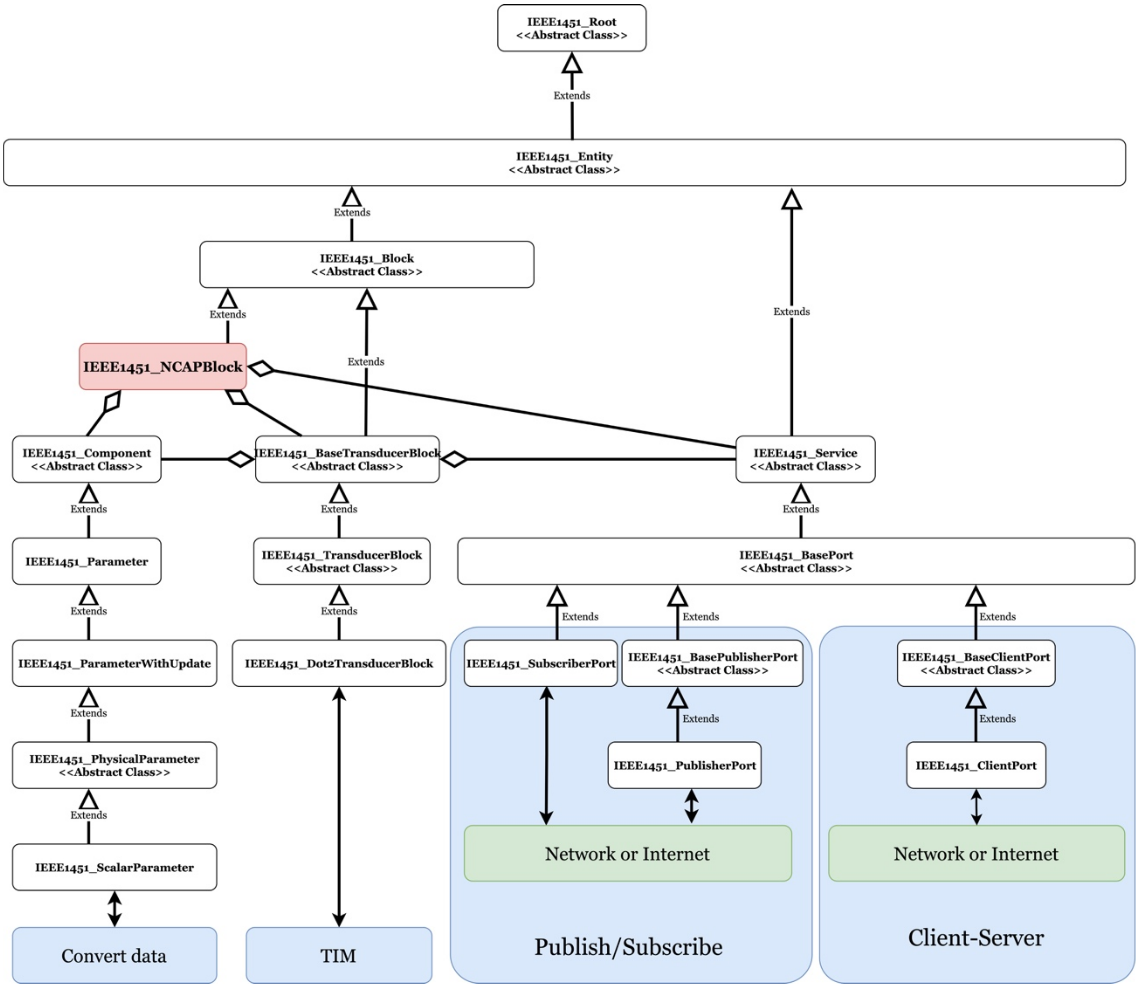


Figure 3.13 Basic NCAP class hierarchy.

3.4 IEC 61499 Standard

IEC 61499 was developed to define a standard architecture and to provide recommendations for the Function Block (FB) usage in Distributed Industrial Process Measurement and Control Systems (IPMCSs) [33]. It presents a generic, domain-independent extensible use of FB. FB was employed previously by the IEC 61131 standard to program the Programmable Logic Controller (PLC). After that, the IEC 61131 was improved to overcome the deficiency in flexibility and reusability, and it became the bases and functional software for the IEC 61499. It has been used to develop hardware-independent applications to improve device interoperability between different companies. The IEC 61499-1 addresses the architectural model, the System, the Devices, and the communication network, as shown in Figure 3.14.

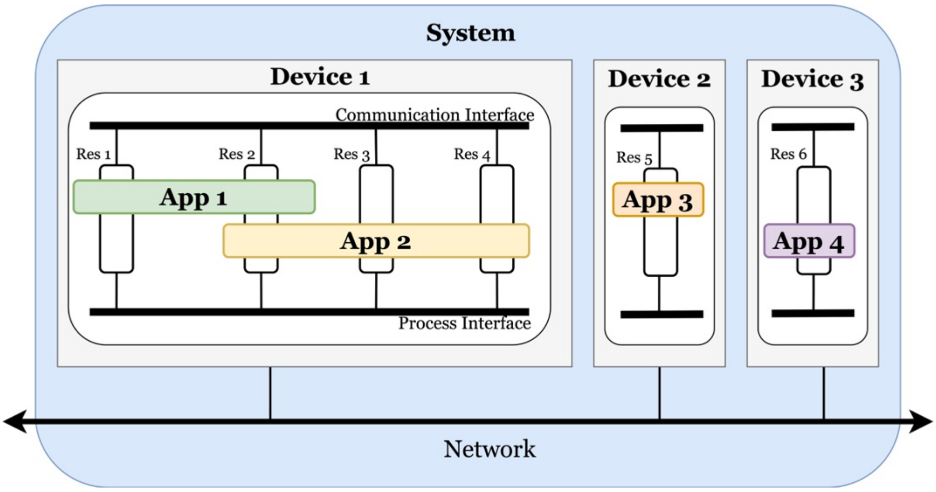


Figure 3.14 IEC 61499 Architecture: System, Device, and Resource adapted from [33].

FB is a software developed to be the bases for the IEC 61499. A FB comprises inputs and outputs events and inputs and outputs data [33]. A basic FB contains Execution Control (EC) and one or more algorithms. When an event arises in an input, the algorithm runs, consuming the input data, and the output data will be obtained. After its processing, the equivalent output event will be subsequently delivered. IEC 61499 is used to implement the control system. The Eclipse 4diac was developed based on the IEC 61499 as an open-source infrastructure for distributed industrial measurement and control. This infrastructure will be discussed in more detail in the implementation section.

3.5 Message Queuing Telemetry Transport and Hypertext Transfer Protocol

The Message Queuing Telemetry Transport (MQTT) implements the publish/subscribe paradigm. The subscriber subscribes to a middleware named broker. When a publisher publishes the corresponding topic subscript by the subscriber, the broker leads the message to the subscriber. The MQTT was selected because the 4diac implements it, and the IEEE 1451.1-6 working group is working on implementing MQTT in the NCAP.

The MQTT was designed as a message protocol that uses the publish/subscribe paradigm to exchange messages between different users [5]. The communication process occurs based on three Quality of Service (QoS) levels. With QoS 0, “At most once” messages are sent to the subscriber and are not transmitted again if the message is lost. Using the QoS 1 “At least once” ensured that at least one message arrives to the subscriber. In case the message is lost, it will be sent again. However, several messages can arrive to the subscriber while the publisher waits for the message confirmation. QoS 2 “Exactly once” ensures that only one message will arrive to the subscriber. The publisher waits for two confirmation messages from the subscriber. This increases the overhead message [5]. MQTT has been employed in the context of the industrial IoT [13]. Other protocols are being developed and implemented in the IoT fields. Each protocol has its way of communicating.

The Hypertext Transfer Protocol (HTTP) was introduced in 1995 as an application protocol for distributed collaborative and hypermedia information exchange. HTTP implements the client-server paradigm. Nowadays, it is widely used for communication over the internet, implementing the request and response message exchange by employing the Uniform Resource Identifier (URI) as the Uniform Resource Locator (URL) to indicate the resource location. Since 1999, HTTP has been in its version HTTP v1.1 [34].

Both standards support MQTT and HTTP. in the application layer from ISO/OSI. However, the 4diac only supports HTTP v1.0.

3.6 Interoperability between the IEC 61499 and the IEEE 1451 standards

The distributed scheme of IEEE 1451 and IEC 61499 standards has led the authors to propose the control and measurement system for distributed industrial automation systems. Moreover, the distributed method in both standards is presented for different levels of abstraction. For instance, IEEE 1451 benefits from TEDS and TIM at sensor and actuator levels, whereas IEC 61499 does not specify any guideline after the process interface in function blocks. However, IEC 61499 focuses on the control level, which IEEE 1451 lacks at the NCAP level. Plug-and-Play and configurability are two properties that both IEEE 1451 and IEC 61499 support at different levels of abstraction. For these reasons, IEEE 1451 and IEC 61499 can complement each other, at both measurement and control levels, by integrating the TEDS and TIMs into IEC 61499 to enhance their low-level configuration.

Both standards address other essential communications and protocols (e.g., MQTT and HTTP). They both support client-server and publish/subscribe protocols and have a common point that can facilitate the interoperability between the two standards.

The IEC 61499 allow Plug-and-Play capabilities in the distributed systems, based on the distributed nature of IEC 61499 Devices independent of the other Devices. The Service

Interface Function Blocks (SIFBs) add or removes an IEC 61499 Device in the real-time [33].

SIFBs are abstract hardware models provided by device vendors, similar to TEDS in IEEE 1451. A new Device can be added easily without compromising the system. On the other hand, IEEE 1451 introduces TEDS to store the transducer's information electronically, enabling the NCAP to easily recognise assets after discovering the TIM and integrating it into the other network. As a result, IEC 61499 supports Plug-and-Play at the control level, whereas IEEE 1451 covers it at the transducer level. Sensors and actuators are coupled or decoupled easily since IEEE 1451 supports the plug-and-play mechanism. Previous characteristics result in plug-and-play at the control and measurement levels and benefit from the (re)configuration at any level of abstraction.

The IEEE 1451 and the IEC 61499 standards address a uniform design modelling to develop a control or measurement system. The standards explicitly mentioned that the designed system must communicate, interoperate, and be network-independent. The user application must be designed regardless of the communication network used and be platform-independent. They also address that reconfigurability and portability should be achieved independently of the underlying infrastructure and communication network adopted. Time management, uniform models and schemes for data and events, and concurrency management must also be supported. As a result, an application developed based on either or both standards must be able to be mapped (on NCAP or Device) and communicate regardless of the platform and network, respectively. Device and NCAP structures are shown in Figure 3.15.

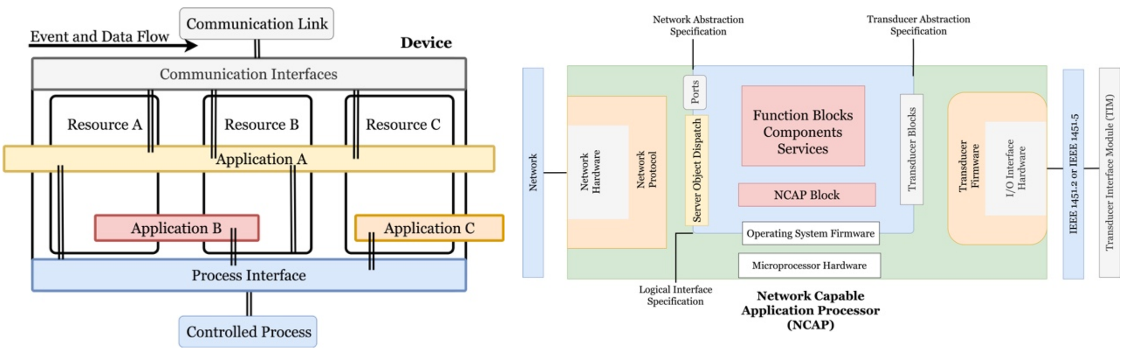


Figure 3.15 Device model adapted from [35] and NCAP model adapted from [7].

The Inter-NCAP concurrency is not supported in the IEEE 1451. Although the IEC 61499 provides Inter-Device concurrence, resources inside a Device perform tasks simultaneously. The NCAP does not support multiple operations, whereas the IEC 61499 supports it. State Machine Control is also introduced in both standards at different levels.

For instance, the function block's Execution Control Chart (ECC) is designed based on the state machine since it is an event-driven control scheme [33]. In the NCAP, the state machine can develop the block life cycle. A distributed control and measurement addresses interoperability, configurability, and portability achieved by employing both standards.

Table 3.1 demonstrates the properties and characteristics of the IEC 61499 Device and the IEEE 1451 NCAP share [13].

Table 3.1 Device and NCAP comparison and shared characteristics from the standards [13].

Property	IEC 61499 Device	IEEE 1451 NCAP	Shared Characteristics
Inter-Concurrency	Inter-Device	No Inter-NCAP	Uniform Design Modelling
Intra-Concurrency	Resource (FB) Concurrency	No Process Concurrency	Network Independent
Multiple Operations	Yes (at any level)	Within a Process Only	Interoperability
Client/Server	Yes	Yes	Portability
Publish/Subscriber	Yes	Yes	Configurability
State Machine Control	Yes (FB Level)	Yes (Block Level)	Platform Independent Application Design
Mapping	Applications onto Devices	Applications onto Object Types	Plug-n-Play
Application Distribution	Over Multiple Devices	Over Multiple NCAPs	Application Distribution

The conceptual method in Figure 3.16 illustrates how the user interacts with the system utilising the publish/subscribe paradigm. First, the user subscribes to the topic of interest. Publishing methods *GET* and *SET* allow for publishing on the topics. The device subscribes to the topic and waits for the user's command. When the user publishes the topic, the device receives the message and translates it into a command. This command is sent from the device to the transducer to collect the required data. The transducer replies to the device afterwards. Therefore, the device publishes the information on the topic subscribed by the user. The results obtained with the proposed method confirm syntactic interoperability between the standards presented in the next section.

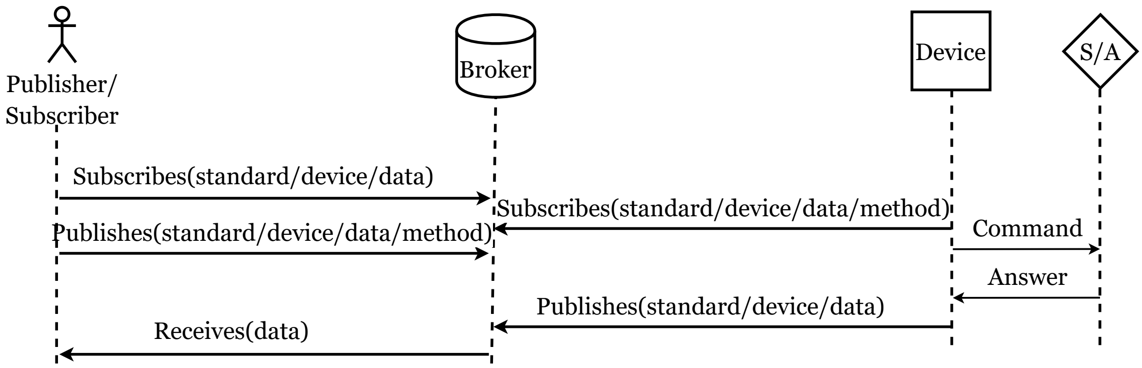


Figure 3.16 User Interaction Method.

3.6.1 First Interoperable Scenario Implementation

The proposed scenario validates the interoperability of a distributed control and measurement system. A pH sensor measures the pH value of a water tank. This node can also trigger an alarm whenever the pH value exceeds programable alkaline and acidic thresholds. This sub-system is compliant with IEEE 1451. The application level is an IEC 61499-compliant system that controls the pH value and issues the alarms in a dashboard. The user is also allowed to set the pH alarm levels. The data exchange uses the MQTT network protocol.

The Eclipse 4diac FORTE software runs on Raspberry Pi B to allow the development of FB in real-time, interacting with the control application running on a computer via Wi-Fi. Publish/subscribe FB is used to communicate with the MQTT broker. Upon initialisation, Eclipse 4diac publisher FB publishes several messages to set the Alkaline and Acidic pH alarm levels. NCAP, which has already subscribed to those topics, will receive the alarm levels for further processing. 4diac also subscribes to the generated alarms and the pH level to implement the control system.

The IEEE 1451 standards cover all aspects of pH smart sensor implementation, from physical to application. An NCAP was programmed in Python on a Raspberry Pi 3 B+. The NCAP communicate with the TIMs sending commands using the UART serial connection. The first command is the “Read Meta TEDSs command” to get the information about the system and the Transducer Channels, as shown in Figure 3.17.

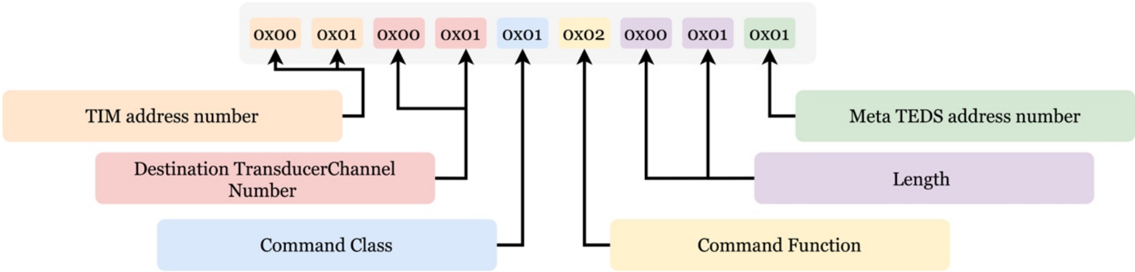


Figure 3.17 Read Meta TEDS command.

The TIM answer the NCAP with the Meta TEDS information, as presented in Figure 3.18. The Meta TEDS is stored in the NCAP, and by the TransducerChannel number (MaxChan), the NCAP reads the other mandatory TEDS: TransducerChannel TEDS for each channel, User's Transducer Name TEDS and PHY TEDS. All the TEDS are stored inside the NCAP memory and available for request from the user application.

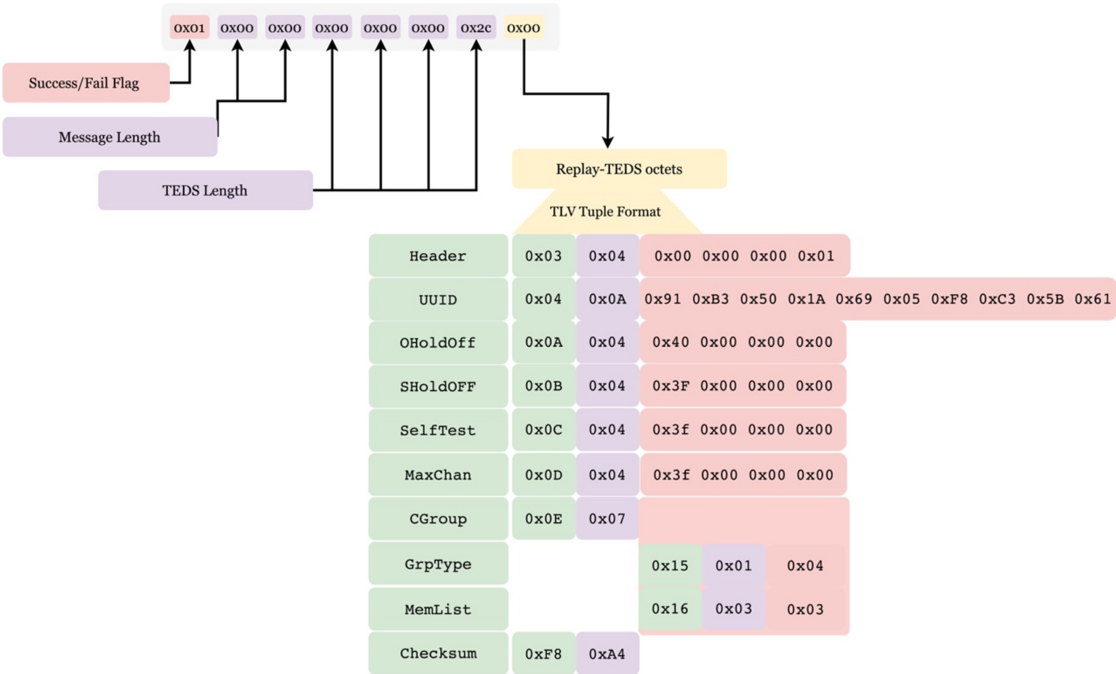


Figure 3.18 Replay Read Meta TEDS.

Inside the Meta, TEDS was defined as three TransducerChannels. At this moment, the NCAP reads the sensor's information by its TransducerChannel address. In the MQTT broker, there are specific topics to which NCAP subscribes. When the NCAP receives a message from the broker, it interprets it to generate a command that is sent using UART from the NCAP to the TIM. The TIM reads the pH value from the sensor and converts its output value from analogue to digital whenever it receives the command to read the *TransducerChannel*. The acquired data is sent using a UART serial connection to the NCAP. To conclude the communication, the value is converted from a float value, returned from the TIM to the float format representative of the pH value and published to the related topic (e.g., response topic) in the broker by the NCAP, as shown in Figure 3.19.

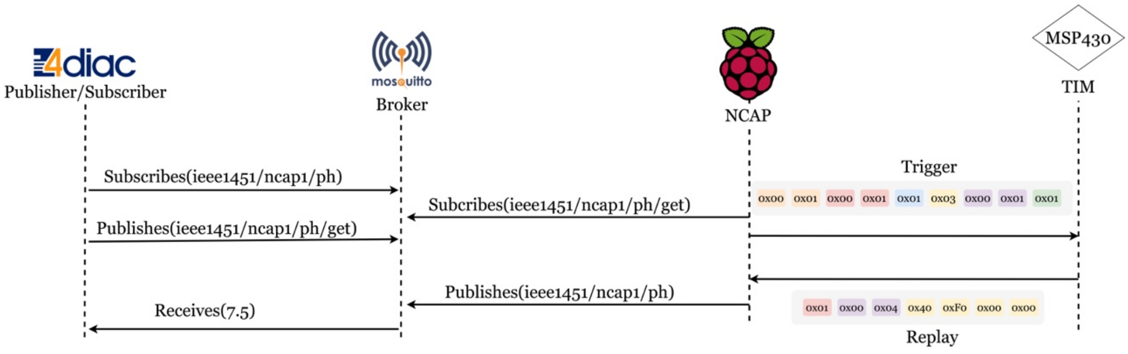


Figure 3.19 pH measurement communication.

The smart sensor is implemented in a low-power TI MSP430F5529 MCU. Two analogue event sensors detect changes in the water pH value. An alarm triggers when the pH value of water crosses the alkaline threshold, and another alarm triggers after the pH value surpasses the acidic threshold. The upper and lower limits define the gap for the hysteresis, with a size greater than or equal to zero. There are two types of transitions. The rising transition occurs when the pH value surpasses the alkaline threshold value, setting the event sensor output to 1. In contrast, the falling transition happens when the pH value overpasses the acidic limit value, setting the event sensor output to 1. The value between the acid and the alkaline is set as 0, meaning that the pH value is good.

The IEEE 1451.0 standard defines the analogue event sensor using four transducer channels joined in a Control Group. An analogue event sensor is the principal transducer channel to sense threshold crossing. The analogue input sensor converts the signal to digital and sets the output state. A *TransducerChannel* and the embedded actuators define an upper threshold and the hysteresis interval, respectively. These two last transducer channels control the operation of the Control Group.

The NCAP requests the TIM water's pH value using the "TransducerChannels Data-Set Segment Read" command. Inside a TIM, the input sensor converts the pH value from analogue to digital and returns it to the NCAP. An analogue event sensor monitors the buffer and states for an input sensor. Sending a trigger command to the Control Group makes the "*TransducerChannel*" put a single sample into the dataset. Moreover, the event sensor data buffer is updated when an event is created by the pH value crossing the alkaline or acidic thresholds. The event value, 0 or 1, is sent to the NCAP. This answer is generated without needing a "*Read TransducerChannels Data-Set Segment*". The embedded actuators inside the TIM operating in a continuous sampling mode set the upper threshold and upper hysteresis values. When the Control Group receives a trigger, pH value acquisition starts [36]. A method based on the MQTT topics was developed to promote syntactic interoperability. This method was presented in Distributed Control and Measurement System Section and considered the existence of an NCAP compliant with IEEE 1451.1 that interprets the received commands. An implementation of this method is presented in Figure 3.20. The user application subscribes to the four topics representing water pH thresholds. The user application publishes the limits of the pH values, which are considered safe water. The user application receives an alert about the corresponding event whenever the limit is exceeded. Also, the user application can request the current value of the pH based on the commands described in Figure 3.20. The broker used to communicate is the Mosquitto [37] installed in a Raspberry Pi 3 B+.

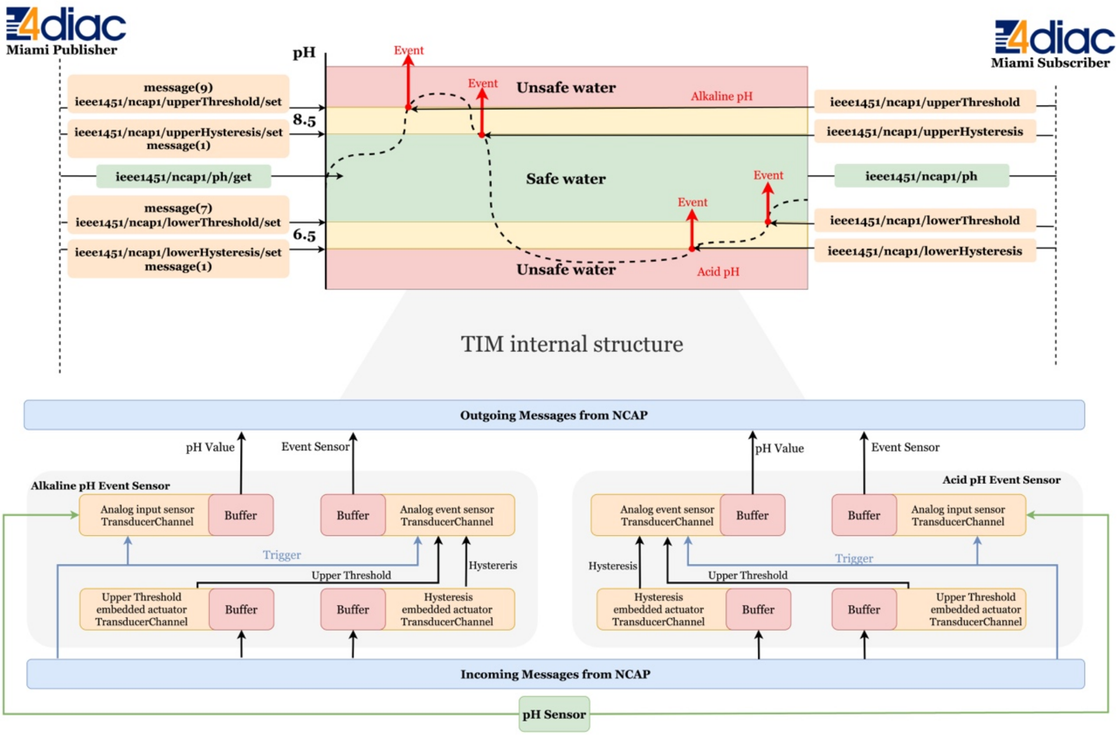


Figure 3.20 Implementation of the pH monitoring system for test and validation.

3.6.2 Evaluation Platform

4diac control application publishes/subscribes to four topics, as shown in Figure 3.21. Four publish function blocks set the alkaline and acidic thresholds and upper and lower hysteresis values. The values are sent to publish FB employing STRING2STRING converters. At the same time, the IEC 61499 control application also monitors the threshold, hysteresis, and pH values in real-time by using Subscribe function blocks. The values then go through a string to an integer converter and are compared with pre-defined values using F_EQ function blocks. The results are demonstrated on Raspberry Pi by LEDs as alarms. QX function blocks are outputs that show the values at the pins of the Raspberry Pi. An analogue pH sensor from the Gravity module measured the pH value. The ADC converts the output of this module from the TI MSP430F5529 MCU.

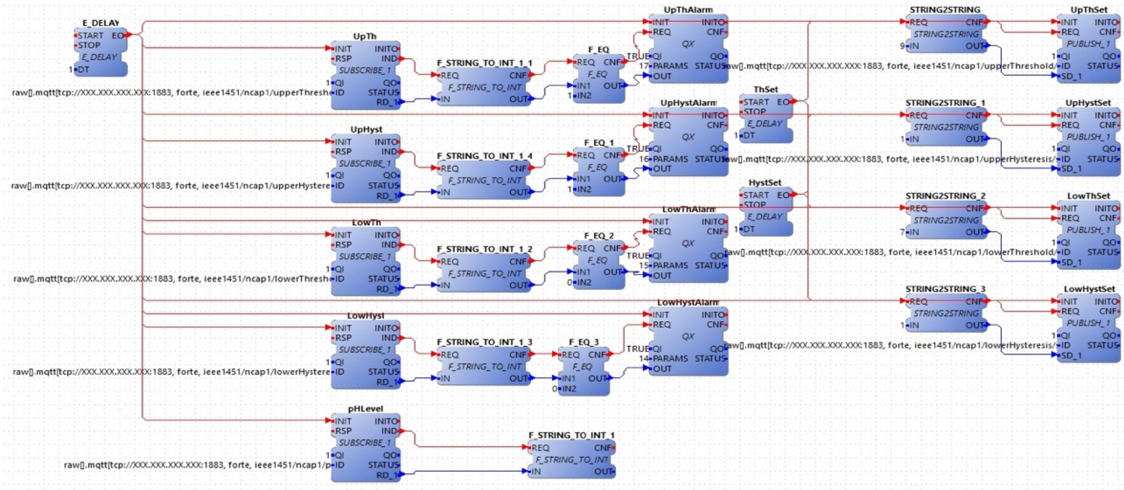


Figure 3.21 4diac implementation of the pH control system.

3.6.3 Test Cases

Test Case 1 (TC1): The pH value was periodically requested from the IEC 61499 side. The maximum packet size was measured, specifically to the time taken to deliver a message. Three rounds of 100 messages were exchanged using MQTT QoS 0. No messages were lost in the communication process, as shown in Table 3.2. The packet size published from 4diac was 94 bytes, and the packet size published from the IEEE 1451 was an average of 99.92 bytes. The average communication time for the three tests was 103.58 seconds, with an average network latency of 0.165 seconds (Table 3.3).

Table 3.2 Results from TC1.

TC1	Number of Messages		Packet Size Averages/Bytes		Time in seconds (s)	
Test	Publisher	Subscriber	Publisher	Subscriber	Network Latency	Communication Time
1	102	102	94	100	0.163	103.60
2	101	101	94	100	0.158	102.61
3	103	103	94	99.78	0.173	104.51
Average	102	102	94	99.92	0.165	103.58

Table 3.3 Results from TC2.

TC2	Number of Alarm Received				Packet Size Averages/Bytes		Time in seconds (s)	
Test	Upper Threshold	Upper Hysteresis	Low Threshold	Low Hysteresis	Publisher	Subscriber	Network Latency	Communication Time
1	9	10	10	10	103,5	100.5	0.133	259.75
2	9	10	9	10	104.5	100.5	0.132	291.01
3	10	10	10	10	104.5	100.5	0.134	272.93
Average	9.33	10	9.67	10	104.16	100.5	0.133	274.56

Test Case 2 (TC2): the IEC 61499 monitoring system performed smart sensor setup through the embedded actuators that define the alarm limits. The four parameters, two for each embedded actuator, were successively configured. Three tests were carried out where ten rounds of the four alarms (*UpperThresold*, *UpperHysteresis*, *LowerThreshold*, and *LowerHysteresis*) were tested. The IEC 61499 monitoring system first sets the threshold and hysteresis values using setting messages. The packet sizes to set the values (e.g., "*ieee1451/ncap1/upperThreshold/set*") have an average of 104.16 bytes for the four published topics. The packet size received from the alarms (e.g., "*ieee1451/ncap1/upperThreshold*") averages 101.5 bytes. The latency in the communication process has been calculated using the pings signal received by the MQTT broker from 4diac with a packet size of 68 bytes. The tests were performed using MQTT QoS 0. The average latency time for the three trials was 133 milliseconds, and the average communication time of the test was 274.56 seconds (Table 3.3). The TC1 has a latency time of 165 milliseconds, and the TC2 has 133 milliseconds. Both latency times are lower than 300 milliseconds on average latency time between two continents [38].

3.6.4 Evaluation

The developed experimental setup allowed to study of interoperability between the two standards. IEC 61499 and IEEE 1451 family of standards successfully communicated using MQTT by developing a distributed control and measurement system. IEC 61499 standard does not only monitor the measuring system but also controls it system by setting the hysteresis levels. The tests demonstrated that it is possible to define operating scenarios using bidirectional communication among components implemented on each side.

Latency is a critical characteristic of an industrial network. The quality of this property arrives from the metric defined to fulfil the network's requirements. For a monitoring application, latency should be lower than 100 milliseconds [39]. For a safety application, this parameter should be less than 10 milliseconds [40].

Data loss in an industrial network shows the system's reliability. A package can be lost or received with a delay making it obsolete, and both situations will have the same impact on the process. The obtained results prove that the proposed solution is highly reliable.

It should be noted that although the MQTT is the fastest and lightly messaging protocol for IIoT applications compared with HTTP [41], the broker can be a single failure point, potentially shutting down the whole network. However, MQTT supports the bridging mechanism by adding redundancy to the broker to improve reliability. It should also be mentioned that another alternative for publish/subscribe is the usage of client-server, which on the one hand, is supported by two standards and, on the other hand, requires more computational complexity. As a result, the proposed methodology can be modified based on the application and the overall system's requirements.

Another study was developed by the author where the IEEE 1451 and IEC 61499 standards are used inside the IIRA communication layer [41]. This study used the IEC 61499 to manage e control e the IEEE 1451 standards to acquire temperature information to simulate an industrial environment. The same NCAP and MSP 430 boards were utilised in the study. Also, they compared the communication using the MQTT and the HTTP protocols for message exchange at the syntactical interoperability level. The conceptual setup is presented in Figure 3.22.

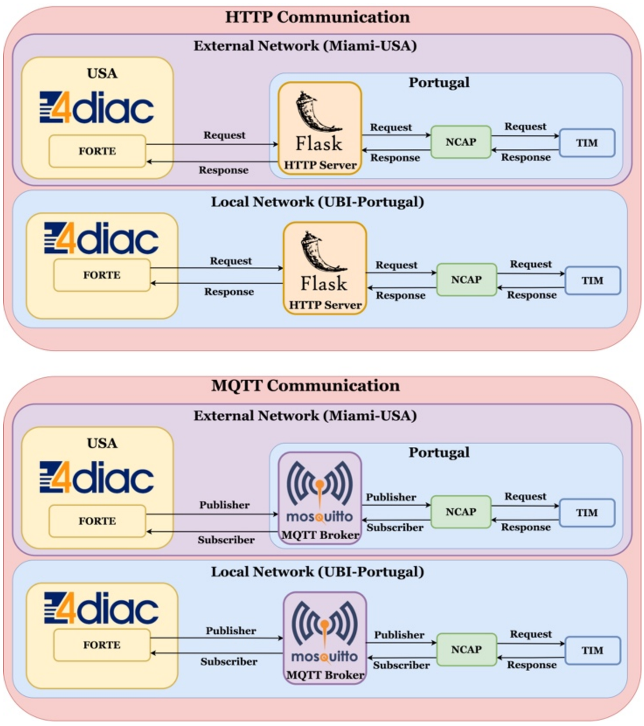


Figure 3.22 HTTP and MQTT communication [41].

This study had the focus on communication latency and packet loss. The tests were performed over five days, and more than forty thousand messages were exchanged. The Wireshark software was utilised to monitor the network on the UBI and Miami Universities sides. The latency communication time is shown in Figure 3.23. It is known that the temperature sensor takes 1000 milliseconds to acquire the data via UART. Decreasing the data acquisition time, the MQTT protocol latency at UBI is 43.78 milliseconds, and at Miami University, it was 41.92 milliseconds. Whereas utilising the HTTP protocols was 67.90 milliseconds and 265.03 milliseconds at UBI and Miami University, respectively. It concludes that the MQTT communication time was faster than HTTP, even in cross-world communication. Even the worst HTTP latency communication is lower than 300 milliseconds established by communication latency between continents [38]. Kalør et al. [39] established the latency for the scale reading information as 100 ms, enabling the MQTT and HTTP protocols in a local network, adequate for Industry 4.0 reading data. However, as shown in Figure 3.24, the packet loss for this faster communication was worst using MQTT than HTTP in an internal and external network. It is important to note that the Eclipse 4diac only implements the MQTT with QoS 0 “fire and forget” [42], which does not provide a confirmation message. The packet loss number could decrease using a higher QoS 1 or 2 that implements confirmation messages. The results comparing the latency time using the mean, standard deviation, median, minimum, and maximum for communication are presented in [41].

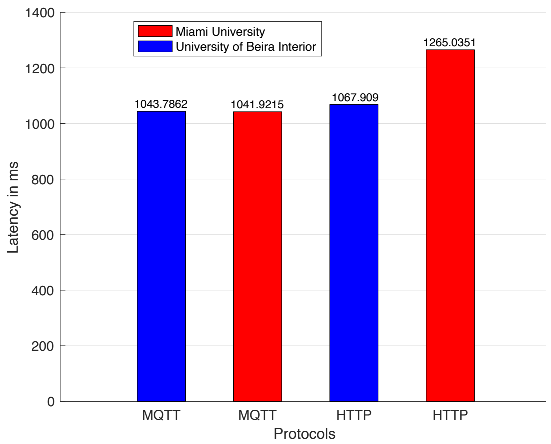


Figure 3.23 Latency time communication UBI and Miami.

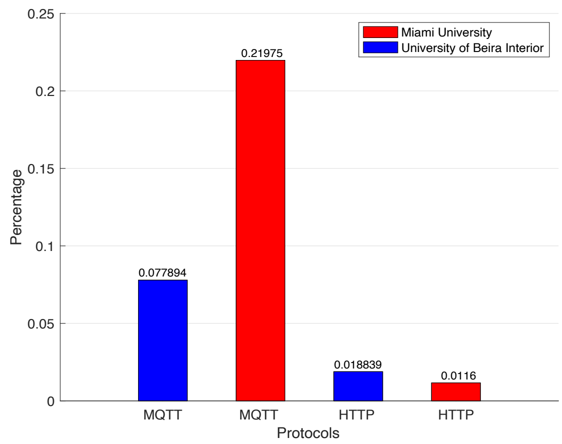


Figure 3.24 Packet loss percentage UBI and Miami.

3.6.5 Testing the Syntactic Level of Interoperability

The Standard and Interoperability Plugfest (INTEROP) event occurs every inside the Industrial Electronics Society (IES) conference. It focuses on the verification and validation platforms open to the community for the developer to test the systems and applications to ensure compliance and interoperability with the IES systems and field of interest. Industry partners can verify and validate their platforms' compliance with interoperability in the industrial context. Inside the event are presented prototypes and implementation of the interoperability and the standards.

The first event was inside and took place at the Annual Conference of the Industrial Electronics Society – IECON Conference 2018 in Washington, D.C. United States [43]. There were nine participants in the event from Portugal, Argentina, Japan, Hong Kong, India, and the United States that promoted the interoperability and hands-on demonstration focusing on promoting the IEEE 1451 standards for IoT, IIoT, and CPS.

In 2019 the INTEROP happened at the IECON in Lisbon [44]. This was the first attempt of the author of this thesis to the INTEROP. It promoted the interoperability between the IEEE 1451 standards at the UBI with other working groups, IEEE P1451.1-6 (MQTT), IEEE P1451.99 (IoT Harmonization), and an Austrian enterprise. The equipment from the UBI laboratory was utilised during the event for three days. The initial development hardware consists of the development board, MSP-EXP430f5529, from Texas Instruments, as a TIM and a Raspberry Pi as the NCAP. A serial communication through the UART port. In the second scenario, wireless communication was added. This connection allows communication with multiple TIMs [45]. The radio used is the CC2500 from Texas Instruments, as shown in Figure 3.25.

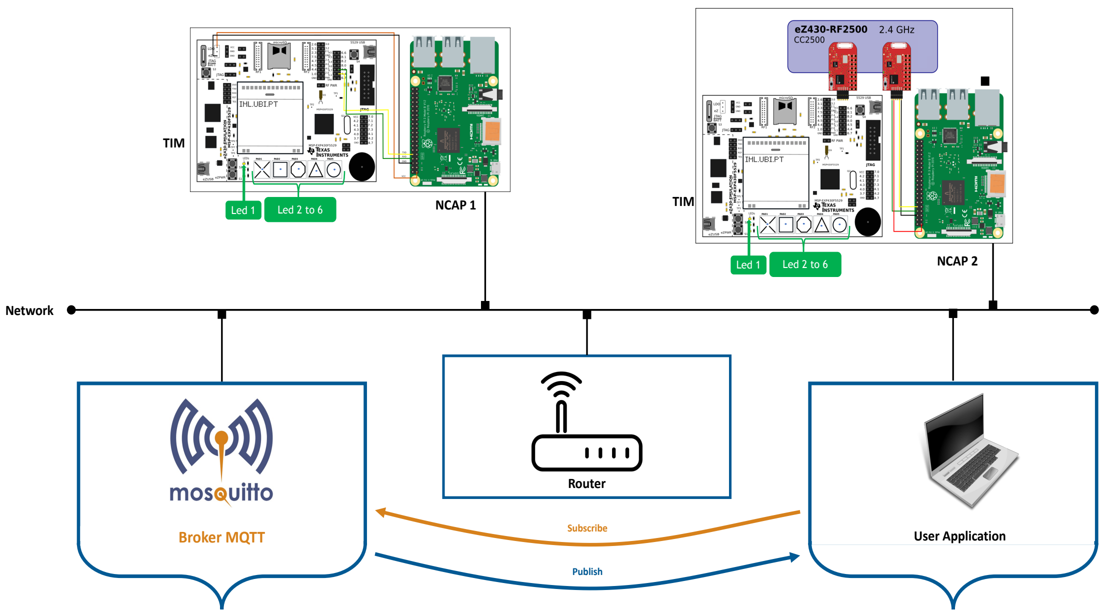


Figure 3.25 INTEROP scenarios.

The MQTT's topic structure changed to **/ieee1451/ncap1-2/:**

Topics to subscribe: **ieee1451/ncap1-2/temperature** and **ieee1451/ncap1-2/voltage**

To interact with the LEDs:

Publisher can turn on LEDs 1 to 6:

Topic: **ieee1451/ncap1-2/led1-6**

Message: **on**

Publisher can turn on or off all LEDs:

Topic**: ieee1451/ncap1-2/leds**

Message**: on** or **off**

Subscriber can see the status of each one of the LEDs (1 to 6) if it is on or off:

Topic: **ieee1451/ncap1-2/led1-6/status**

The message structures arrive through the topics from the user's application to the NCAP that was subscribed to the same topic. The NCAP converted it as a command and sent the command to the TIM using the octet format. The TIM answers the NCAP in the same format that converts the data using the information presented in the TransducerChannel TEDS. All the communication uses the same structure as in section 3.6.1, First Interoperable Scenario Implementation. With this implementation and testing at the event was possible to validate the development at the UBI laboratory and exchange information and knowledge with the other working groups. The hardware setup developed at the UBI laboratory is presented in Figure 3.26.

The level of interoperability achieved at this event was the syntactical level allowing to exchange of data using the same format with other groups. It also developed an application to control the turning on and off of the LEDs and get information about the temperature and voltage shown by the respective graphs, as shown in Figure 3.27 and the UBI workstation at IECON in Figure 3.28.

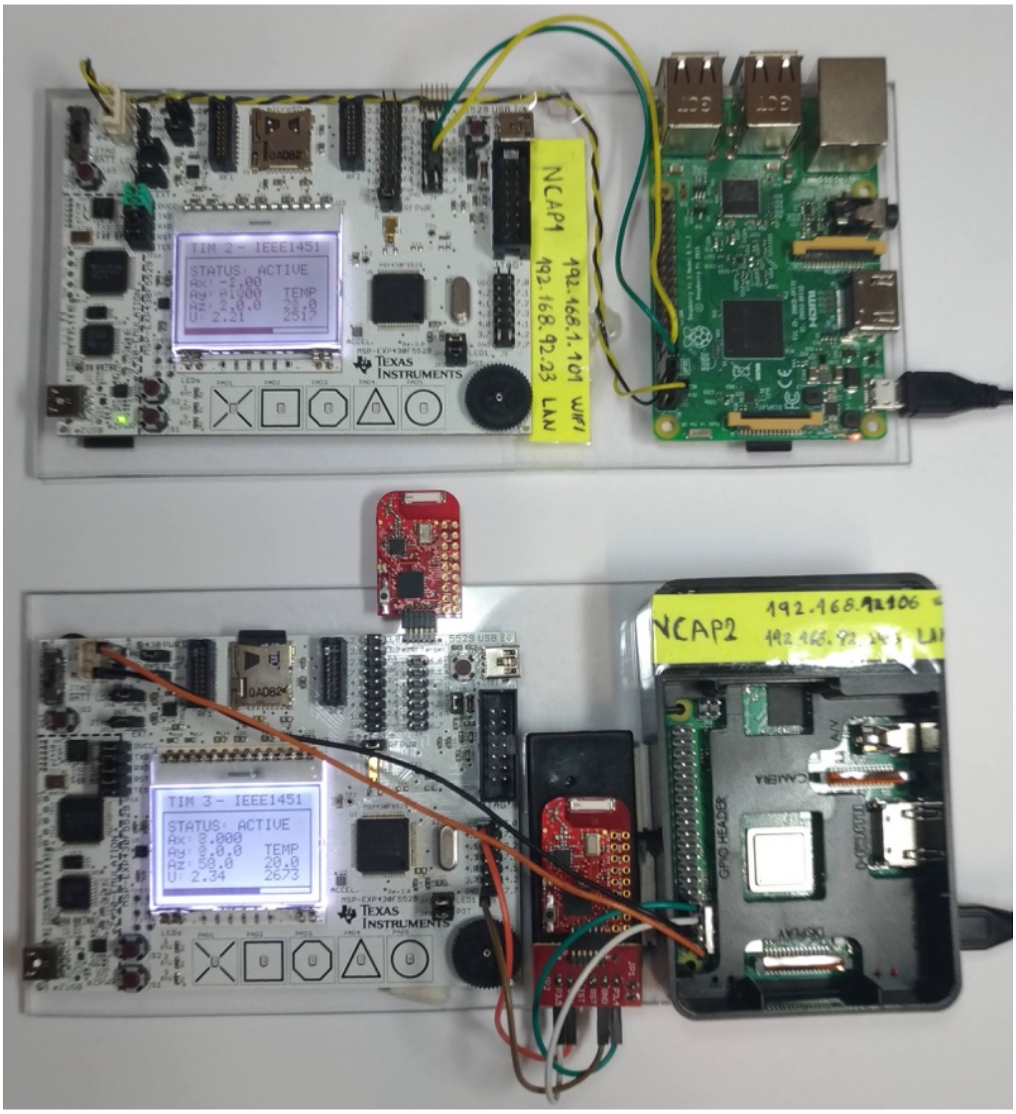


Figure 3.26 UBI hardware setup.

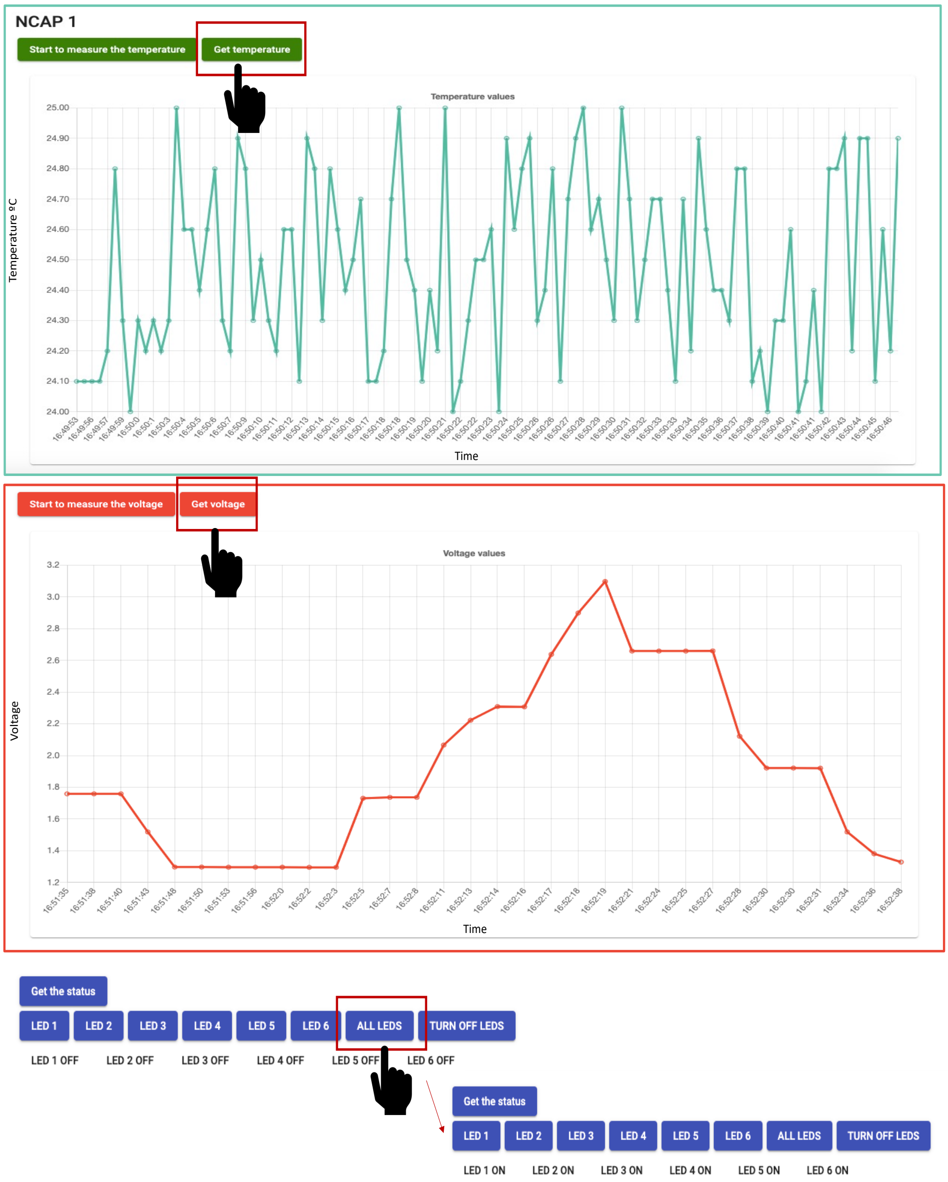


Figure 3.27 INTEROP application.

Figure 3.28 UBI workstation at INTEROP.

At the INTEROP event, the UBI provided data using the same setup as in 2019 to interoperate with more universities worldwide, as shown in Figure 3.29. For this event was create an HTTP server at UBI and was developed a Manager Information Base (MIB) for the SNMP protocol.

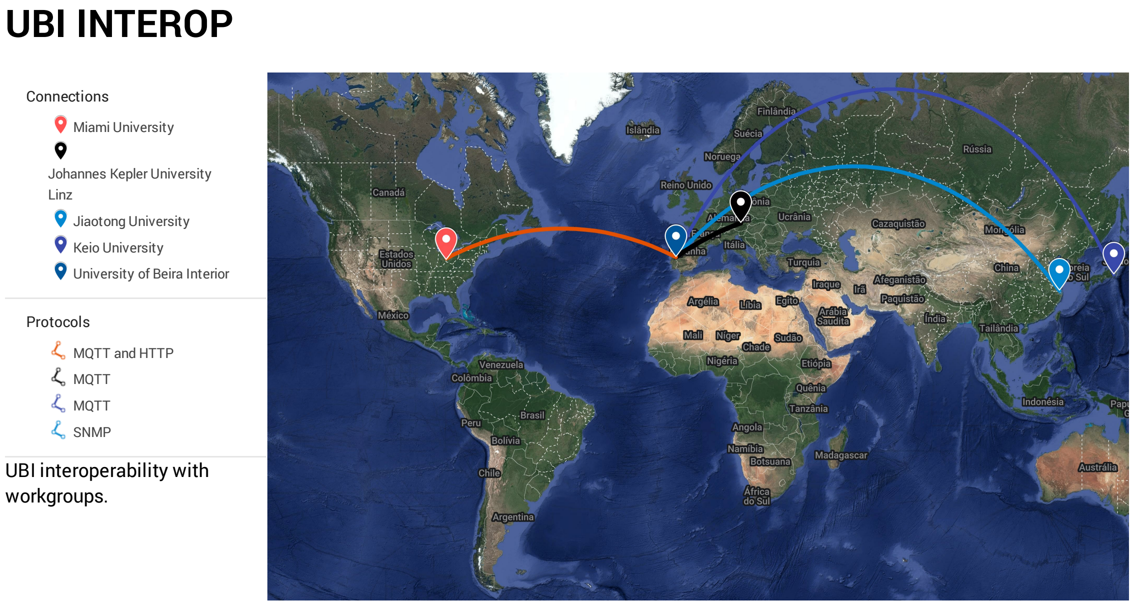


Figure 3.29 INTEROP event at IECON Singapore.

The INTEROP of 2020 was online by the restriction to travel held inside the IECON 2020 in Singapore. This year, the UBI has become a Center of Expertise (CoE), focusing on industry services and support sponsored by the IES to help develop and implement the standards. The CoE is a platform not only for standard development. It is "a basis for the interoperability test and verification" [46]. There are two active CoEs, CoE@UBI and CoE Japan (CoE@JP). At the CoE@UBI are developed and validated smart sensors with power and computational restrictions addressing the energy consumption, computational resources, interoperability, validation, and verification of IEEE 1451 implementations. The setup utilised in 2020 and 2021 during the restriction time is shown in Figure 3.30.

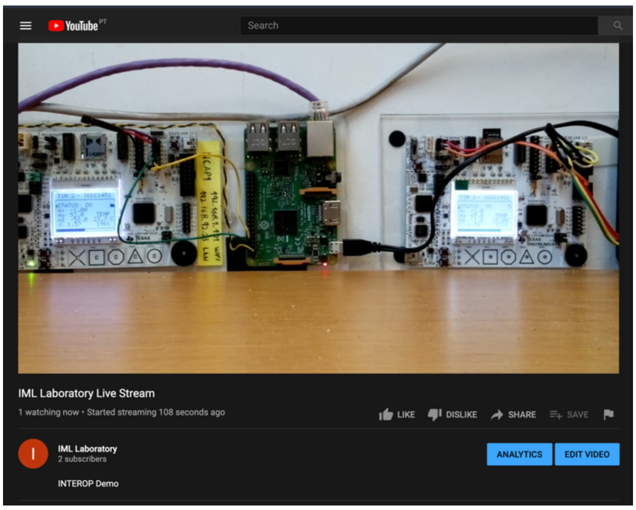


Figure 3.30 Setup Livestream on YouTube.

In 2021 INTEROP event took place, the first at Symposium on Industrial Electronics (ISIE) in Kyoto, Japan and the second at the IECON 2021 in Toronto, Canada [47]. The setup for these two events was the same as the previous one. At the first event, the interoperability was between the UBI and Johannes Kepler University, and the second was between the UBI and Miami University.

3.7 Conclusion

This Chapter presented the IEEE 1451 family of standards and how to reach the syntactic level of interoperability between two standards that can be used inside an IIoT and Industry 4.0 context.

The IEEE 1451 standard was developed to manage transducers connected to a TIM and communicate through an external network using an NCAP. The IEC 61499 was developed to automate and manage applications in a distributed network. Both standards have their characteristics but can interoperate and collaborate between them.

The INTEROP event inside the IECON conference allows the systems developed inside the UBI to interoperate with other protocols and applications. The level of interoperability reached in these contexts was the syntactic levels employing the same communication protocols.

The main limitation of the interoperability at this level was the dependence on using the same communication protocol and the same data serialisation format for the communication.

After the validation proved in this Chapter was the base for the next level of interoperability. The semantic level is the requirement for Industry 4.0. This level of interoperability is discussed and reached in the next Chapter.

References

[1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, Opportunities, and Directions," *IEEE Trans. Ind. Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[2] I. S. O. Iso, "IEC/IEEE International Standard-Systems and Software Engineering–Vocabulary," ISO/IEC/IEEE 24765: 2017 (E), 2017.

[3] M. Noura, M. Atiquzzaman, and M. Gaedke, “Interoperability in Internet of Things: Taxonomies and Open Challenges,” *Mob. Networks Appl.*, vol. 24, no. 3, pp. 796–809, 2019.

[4] A. Hazra, M. Adhikari, T. Amgoth, and S. N. Srirama, “A Comprehensive Survey on Interoperability for IIoT: Taxonomy, Standards, and Future Directions,” *ACM Comput. Surv.*, vol. 55, no. 1, Nov. 2021, doi: 10.1145/3485130.

[5] A. Banks, E. Briggs, K. Borgendale, and R. Gupta, “MQTT Version 5.0,” *OASIS Stand.*, 2019, [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html.

[6] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[7] I. 2010 IEEE 21451-1:, “ISO/IEC/IEEE Information technology – Smart transducer interface for sensors and actuators – Part 1: Network Capable Application Processor (NCAP) information model,” *ISO/IEC/IEEE 21451-1:2010(E)*, pp. 1–356, 2010, doi: 10.1109/IEEESTD.2010.5668469.

[8] N. Zaman, *Wireless Sensor Networks and Energy Efficiency: Protocols, Routing and Management: Protocols, Routing and Management*. Information Science Reference, 2012.

[9] J. Higuera and J. Polo, “Interoperability in wireless sensor networks based on IEEE 1451 standard,” in *Wireless Sensor Networks and Energy Efficiency: Protocols, Routing and Management*, 2012.

[10] K. Saito and H. Nishi, “Application Protocol Conversion Corresponding to Various IoT Protocols,” in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 2020, pp. 5219–5225, doi: 10.1109/IECON43393.2020.9255101.

[11] M. A. A. da Cruz, J. J. P. C. Rodrigues, J. Al-Muhtadi, V. V Korotaev, and V. H. C. de Albuquerque, “A Reference Model for Internet of Things Middleware,” *IEEE Internet Things J.*, vol. 5, no. 2, pp. 871–883, 2018, doi: 10.1109/JIOT.2018.2796561.

[12] L. Gleim *et al.*, “FactDAG: Formalizing Data Interoperability in an Internet of Production,” *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3243–3253, 2020, doi: 10.1109/JIOT.2020.2966402.

[13] R. Abrishambaf, H. Da Rocha, and A. Espirito-Santo, “IEC 61499 and IEEE 1451 for Distributed Control and Measurement Systems,” 2021.

[14] H. Da Rocha, A. Espirito-Santo, and R. Abrishambaf, “Semantic Interoperability in the Industry 4.0 Using the IEEE 1451 Standard,” 2020, doi: 10.1109/IECON43393.2020.9254274.

[15] L. Roffia *et al.*, “A Semantic Publish-Subscribe Architecture for the Internet of Things,” *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1274–1296, 2016, doi: 10.1109/JIOT.2016.2587380.

[16] H. Derhamy, J. Eliasson, and J. Delsing, “IoT Interoperability—On-Demand and Low Latency Transparent Multiprotocol Translator,” *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1754–1763, 2017, doi: 10.1109/JIOT.2017.2697718.

[17] J. An *et al.*, “Toward Global IoT-Enabled Smart Cities Interworking Using Adaptive Semantic Adapter,” *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5753–5765, 2019, doi: 10.1109/JIOT.2019.2905275.

[18] G. Xiao, J. Guo, L. Da Xu, and Z. Gong, “User interoperability with heterogeneous IoT devices through transformation,” *IEEE Trans. Ind. Informatics*, vol. 10, no. 2, pp. 1486–1496, 2014.

[19] I. P. Žarko *et al.*, “The symbIoTe Solution for Semantic and Syntactic Interoperability of Cloud-based IoT Platforms,” in *2019 Global IoT Summit (GIoTS)*, 2019, pp. 1–6, doi: 10.1109/GIOTS.2019.8766420.

[20] “IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats,” *IEEE Std 1451.2-1997*, pp. 1–120, 1998, doi: 10.1109/IEEESTD.1998.88285.

[21] “IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Network Capable Application Processor Information Model,” *IEEE Std 1451.1-1999*, pp. 1–480, 2000, doi: 10.1109/IEEESTD.2000.91313.

[22] “IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Digital Communication and Transducer Electronic Data Sheet (TEDS) Formats for Distributed Multidrop Systems,” *IEEE Std 1451.3-2003*, pp. 1–175, 2004, doi: 10.1109/IEEESTD.2004.94443.

[23] “IEEE Standard for A Smart Transducer Interface for Sensors and Actuators--Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats,” *IEEE Std 1451.4-2004*, pp. 1–454, 2004, doi: 10.1109/IEEESTD.2004.95745.

[24] “IEEE Standard for a Smart Transducer Interface for Sensors and Actuators Wireless

Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats," *IEEE Std 1451.5-2007*, pp. C1-236, 2007, doi: 10.1109/IEEESTD.2007.4346346.

[25] I. S. IEEE 1451.0, "IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats," *IEEE Std 1451.0-2007*, pp. 1–335, 2007, doi: 10.1109/IEEESTD.2007.4338161.

[26] "IEEE Standard for Smart Transducer Interface for Sensors and Actuators--Transducers to Radio Frequency Identification (RFID) Systems Communication Protocols and Transducer Electronic Data Sheet Formats," *IEEE Std 1451.7-2010*, pp. 1–99, 2010, doi: 10.1109/IEEESTD.2010.5494713.

[27] "Information technology--Smart transducer interface for sensors and actuators--Part 7: Transducers to radio frequency identification (RFID) systems communication protocols and transducer electronic data sheet (TEDS) formats," *ISO/IEC/IEEE 21451-7:2011(E) (Revision of 1451.7-2010)*, pp. 1–92, 2012, doi: 10.1109/IEEESTD.2012.6155051.

[28] "IEEE Recommended Practice for Signal Treatment Applied to Smart Transducers," *IEEE Std 21451-001-2017*, pp. 1–109, 2017, doi: 10.1109/IEEESTD.2017.8227039.

[29] IEEE, "IEEE P1451.99 Standard for Harmonization of Internet of Things (IoT) Devices and Systems," Sep. 24, 2020. https://standards.ieee.org/ieee/1451.99/10355/ (accessed Aug. 02, 2022).

[30] IEEE, "IEEE P1451.0 Standard for a Smart Transducer Interface for Sensors, Actuators, Devices, and Systems - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats," *2017*. https://standards.ieee.org/ieee/1451.0/7203/ (accessed Aug. 04, 2022).

[31] E. Y. Song and K. Lee, "Understanding IEEE 1451-Networked smart transducer interface standard - What is a smart transducer?," *IEEE Instrum. Meas. Mag.*, vol. 11, no. 2, pp. 11–17, 2008, doi: 10.1109/MIM.2008.4483728.

[32] R. Pinto, J. Pereira, H. da Rocha, R. I. Martin, and A. E. Santo, "A Discussion about the Implementation of a WSN to Industry 4.0 based on the IEEE 1451 Standard," 2019, pp. 22–25.

[33] R. Abrishambaf, M. Hashemipour, and M. Bal, "Structural modeling of industrial wireless sensor and actuator networks for reconfigurable mechatronic systems," *Int. J. Adv. Manuf. Technol.*, vol. 64, no. 5, pp. 793–811, 2013, doi: 10.1007/s00170-012-4070-y.

[34] R. Fielding *et al.*, "Hypertext transfer protocol--HTTP/1.1," 1999.

[35] International Electrotechnical Commission, *IEC 61499: Function Blocks*, 1º Edition. Geneva, 2005.

[36] A. Espírito-Santo, S. Ambrósio, P. M. D. Serra, and R. J. C. Pinto, "Water pH Monitoring with a Smart Sensor Powered by a Thermoelectric Generator with a Phase-Change Material," in *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, 2019, vol. 1, pp. 5550–5555, doi: 10.1109/IECON.2019.8927259.

[37] R. A. Light, "Mosquitto: server and client implementation of the MQTT protocol," *J. Open Source Softw.*, vol. 2, no. 13, 2017.

[38] P. Ferrari, E. Sisinni, D. Brandão, and M. Rocha, "Evaluation of communication latency in industrial IoT applications," in *2017 IEEE International Workshop on Measurement and Networking (M N)*, 2017, pp. 1–6, doi: 10.1109/IWMN.2017.8078359.

[39] A. E. Kalør, R. Guillaume, J. J. Nielsen, A. Mueller, and P. Popovski, "Network Slicing in Industry 4.0 Applications: Abstraction Methods and End-to-End Analysis," *IEEE Trans. Ind. Informatics*, vol. 14, no. 12, pp. 5419–5427, 2018, doi: 10.1109/TII.2018.2839721.

[40] R. Candell and M. Kashef, "Industrial wireless: Problem space, success considerations, technologies, and future direction," in *2017 Resilience Week (RWS)*, 2017, pp. 133–139, doi: 10.1109/RWEEK.2017.8088661.

[41] H. da Rocha, R. Abrishambaf, J. Pereira, and A. Espirito Santo, "Integrating the IEEE 1451 and IEC 61499 Standards with the Industrial Internet Reference Architecture," *Sensors*, vol. 22, no. 4, 2022, doi: 10.3390/s22041495.

[42] Eclipse, "Eclipse 4diac - The Open Source Environment for Distributed Industrial Automation and Control Systems." https://www.eclipse.org/4diac/ (accessed Oct. 30, 2019).

[43] V. Huang and E. Monmasson, "InterOp 1451 PlugFest Held at IECON 2018, 21?23 October, 2018 [Society News]," *IEEE Ind. Electron. Mag.*, vol. 13, no. 2, pp. 62–64, 2019, doi: 10.1109/MIE.2019.2910895.

[44] V. Huang, "The 2019 Standards and Interoperability PlugFest [Society News]," *IEEE Ind. Electron. Mag.*, vol. 14, no. 1, pp. 88–89, 2020, doi: 10.1109/MIE.2020.2970255.

[45] J. Pereira, H. da Rocha, and A. Espírito Santo, "A Platform for {IEEE} 1451 Standard's Education, Development and Validation for Industry {4.0}," May 2020.

[46] V. Huang, H. Nishi, A. Espírito-Santo, A. Chen, and D. Bruckner, "Standards and Interoperability in Industrial Electronics – A Trending View," in *2021 IEEE 19th*

International Conference on Industrial Informatics (INDIN), 2021, pp. 1–6, doi: 10.1109/INDIN45523.2021.9557461.

[47] "INTEROP 2021: Standards and Interoperability PlugFest at ISIE 2021 [Society News]," *IEEE Ind. Electron. Mag.*, vol. 15, no. 4, pp. 77–81, 2021, doi: 10.1109/MIE.2021.3117294.

Chapter 4

Proposed Semantic Interoperability Methodology for Industrial Internet of Things and Industry 4.0

4.1 Introduction

This Chapter presents the semantic level of the interoperability layer proposed to the IEEE 1451 family of standards. Two papers are directly related to this Chapter:

A Semantic Level of Interoperability by a Proposed IEEE 1451 Family of Standards Ontology

H. Da Rocha, A. Espírito-Santo and R. Abrishambaf, *IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society*, Brussels, Belgium, 2022, pp. 1-6, doi: 10.1109/IECON49645.2022.9968883.

An Interoperable Digital Twin with IEEE 1451 Standards

H. da Rocha, J. Pereira, R. Abrishambaf, and A. Espirito Santo, *Sensors*, vol. 22, no. 19, p. 7590, Oct. 2022, doi: 10.3390/s22197590.

4.2 Semantic Level of Interoperability

The semantic level of interoperability enables the message exchange unambiguously, i.e., "associated with the meaning of content and concerns the human rather than machine interpretation of the content. Thus, interoperability on this level means that there is a common understanding between people of the meaning of the content (information) being exchanged" [1]. This level of interoperability uses syntactic data formats (e.g., XML, JSON) to serialise the message. Presented in Chapter 3.

The semantic level of interoperability is the level of communication needed to achieve inside an industrial context [2], [3]. It has been addressed in the industry communication inside the CPS. Industrial communication should be able: at the physical level to send and receive messages by the agreement of physical and communication layers. At a syntactic level, it does the correct encoding and decoding of the messages, whereas, at the semantic level, it must understand the message by the interpretation and reasoning to parse information according to the context [4]. The information exchange can be wired or wireless. Wireless is used inside a subsystem, and wired is used inside a network infrastructure to ensure compatibility and real-time and reliable capabilities [5].

To address the gaps in implementing various CPS, the reference architecture models were developed by [6]. The industry utilises two references, architecture models. The Reference Architecture Model for Industry 4.0 (RAMI 4.0) and the Industrial Internet Reference Architecture (IIRA). Those architecture models interoperate by the mapping between them [7]. Both reference architecture models support the Open Platform Communication Unified Architecture (OPC UA) inside the communication layer. OPC UA presents the framework layer defined in the connectivity stack defined in the Industrial Connectivity Framework (IICF) [7].

One way to reach the semantic level of interoperability is by developing an ontology. However, it can become another interoperability problem when different ontologies are applied to communicate with each other [2]. Gruber defined ontology in 1993 as “an explicit specification of a conceptualisation” in the artificial intelligence context [8]. Where conceptualisation means an abstract and simplified view that will be represented for some purpose, being an ontology, a vocabulary based on class, relations, functions, and other objects created to share knowledge [8], [9], the IEEE defines ontology as the “logical structure of terms used to describe a domain of knowledge, including both the definitions of the applicable terms and relationships” [10]. Beden and Beckmann [11] developed Semantic Asset Administration Shells, using the Resource Description Framework (RDF) and Web Ontology Language (OWL) for the RAMI 4.0 to provide a vocabulary for the information exchange among the agents used in Industry 4.0. This work has focused on the Information Layer and Communication Layer of RAMI 4.0. Utilising the RAMI 4.0 ontology was developed the Standards Ontology (STO) for the Industry 4.0 [12]. The STO was designed to bring the standards that can be utilised inside Industry 4.0 and their relationship to add the relevant metadata for the industrial standards. A complete ontology for Industry 4.0 was developed using the RAMI 4.0, IIRA, standard, protocols, and reference frameworks developed by Bader et al. titled Industry 4.0 Knowledge Graph Ontology [13]. Gil and Madrigal [14] developed a semantic approach for automation systems named the Automation I4.0 Ontology model. Focusing on the higher, middle, and lower levels of conceptualisation of the automation systems in the I4.0 concept. A semantic model describes object properties, datatype properties, and class instance characteristics. The semantic model uses RDF, RDFS (RDF Schema), TTL/TURTLE (Terse RDF Triple Language), and OWL. In [15], the authors developed an autonomous system using the MQTT protocol and the oneM2M architecture. The system recognises the context by employing semantic web technology and reconfiguring the MQTT-based devices autonomously. As a result, it discovers and operates services and devices in a local network by employing a semantic engine.

There are ontologies based directly on the IEEE 1451 family of standards. Higuera and Polo [16] developed a small ontology based on the definition of the IEEE 1451 family of standards. It was developed inside a Wireless Sensors Network (WSN) containing sensors and actuators inside a wind monitoring system. The TEDS inside de transducers contain the metadata (node properties and localisation). This ontology also includes the remaining node energy e connectivity. To share information autonomously in a different context.

Eid et al. [17] developed a sensor network data prototype ontology based on the IEEE 1451.4 with the description vocabulary list and the language for the IEEE 1451 and the TEDS being the foundation for this ontology. In the paper [18], the authors combined two ontologies in a scenario. The first was an IEEE 1451 ontology based on the IEEE 1451.0 for the TIM and TEDS concept, followed by the sensor properties and observations and IEEE 1451.5 for the communication. The second ontology was the Open Geospatial Consortium (OGC) for the Sensor Observation Services (SOSs) standard built on the complete operation and service-oriented items and their relationship. Both ontologies are matched and applied to the soil temperature sensor, humidity sensor, and digital compass. The three ontologies presented in the previous paper are no longer available to the community by a link address. The new proposed ontology presented in this paper will be open, generic, and accessible at the laboratory link address for the ontology. This enables the community to utilise this ontology in newer projects using the IEEE 1451 standards.

Ontologies were developed to promote interoperability between sensors. The Sensor Semantic Network (SSN) was developed to describe sensors, actuators, procedures, and properties. It is integrated with a core to another ontology named Sensor, Observation, Sample, and Actuator (SOSA) classes and properties. SSN ontology can be applied in many cases, such as in scientific monitoring, industrial infrastructure, and social sensing [19], [20]. The OGC developed Sensor Model Language (SensorML) in 2007 to define processes related to measurement and observation, describe the sensors and the sensors systems, processing and analyse the sensors' observations, geolocation data, and characteristics, e.g., threshold and accuracy. It was built to be a framework for the process with transducer components that used the XML format for data transmission and needed vocabularies and ontologies on the web to promote the semantic interoperability [21]. Also, the OGC developed the Observations and Measurements (O&M), defines XML schemas for observations and sampling those observations for the information exchange [22]. The OGC standards (SensorML, O&M, SOS) are part of the Sensor Web Enablement (SWE) that defines interfaces and protocols that enables the "Sensor Web" applications and services to access sensors and their observations through the Internet.

4.3 IEEE 1451 Family of Standards Ontology Development

The knowledge models store information with meaning for people and computers. It is an effective way to organise and utilise dispersed knowledge in a reusable way [23]. It can be based on ontologies or non-base on ontologies. Ontology-based knowledge modelling use tools, such as the Protégé developed at Stanford University [24] and Web Ontology Language (OWL) developed and maintained by the World Wide Web Consortium (W3C). OWL was published in 2004 as a computational logic-based language created to represent complex knowledge about things, groups of things, and the relationship between the things, to verify the consistency of the knowledge and turn it from implicit to explicit [25]. The OWL is on its second version OWL 2, published in 2009, with its second edition published in 2012 [26]. The main goal of the proposed ontology is to provide a heavyweight ontology that can be used to develop other ontologies based on the IEEE 1451 and provide a semantic layer for the semantic communication that can be used inside Industry 4.0 and the Industrial Internet of Things. The semantic layer is shown in Figure 4.1.

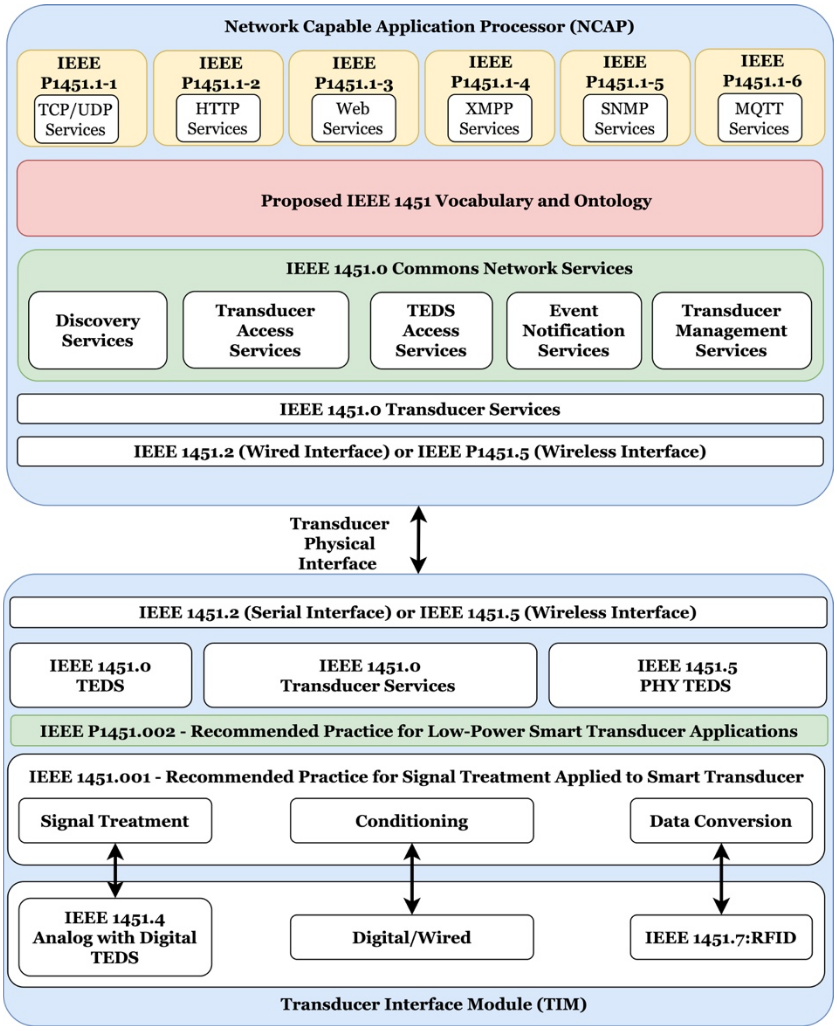


Figure 4.1 Proposed semantic layer for the IEEE 1451 [27].

Ontologies allow the reuse of domain knowledge. Ontologies are close related to information systems and artificial intelligence. Allowing the interoperability to cross information and meaning between systems. Uses of the ontologies are in domain knowledge representation, semantic data catalogues, Natural Language Processing (NPL), linked data, semantic interoperability, semantic resource, visual teaching assets, complexity management, and data management [28].

The development of ontology is based on the learning process. It is represented by pyramid data, information, and knowledge, as shown in Figure 4.2. Data represents raw data; its interpretation depends on the system that this data is used. Information is meaningful, valuable data giving meaning to raw data. Knowledge is understanding the information based on associated patterns based on collecting information and data [29]. This data, information, and knowledge model management contributed to the work and decisions in Industry 4.0 [30], [31]. Knowledge management has a life cycle of six phases: create, formalise, organise, distribute, apply, and evolve [23], [32]. Those six phases are closely related to the ontology development [33]–[35], as shown in Figure 4.3. However, there is no correct way to develop an ontology. It depends on the domain of the ontology used and its purpose. Ontology development is an interactive design process throughout the life cycles of the ontology [9].

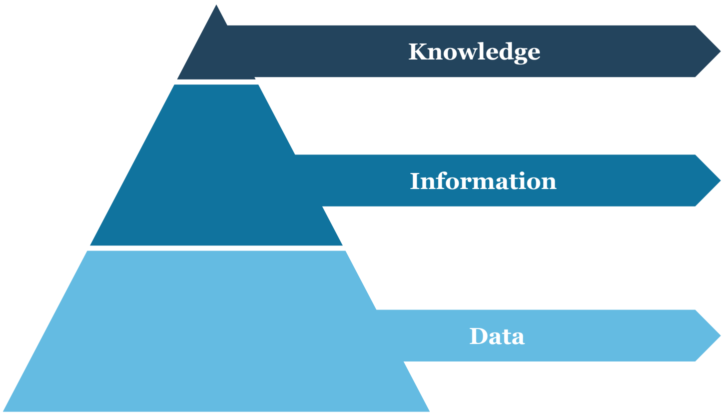


Figure 4.2 Knowledge management pyramid adapted from [29].

Figure 4.3 IEEE 1451 Ontology development process adapted from [32], [36].

4.3.1 Defining

The IEEE 1451 ontology aims to be a heavyweight ontology for smart sensor development based on the IEEE 1451 family of standards. Allowing the user to better understand the standards and allowing the user and machines to communicate semantically. Also, it serves as a base ontology for people who want to develop ontologies for their systems based on the IEEE 1451 standards. A vocabulary based on the IEEE 1451 will be developed to serve the information in the ontology.

4.3.2 Gathering

The methodology used the combine the information is shown in Figure 4.4. The subject is connected to the object by a predicate. The predicate attributes the characteristics to the subject by a linking verb. For example, a Class has Subclass. This structure is the base for the Resource Description Framework (RDF) that is organised inside as tiples [37]. The RDF is the framework that defines the language to describe resources on the web based on their name, property, and values. RDF is the base of the semantic web as the web represents decentralised information in documents that the computer can interpret. The semantic web uses the same base as the web, adding meaning to the representation of the information by its vocabulary defined as RDF Schema (RDFS) [38]. It is represented as a graph with a directional representing the relationships for a better view and understanding of the information by people.

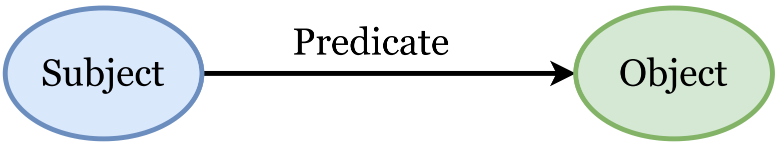


Figure 4.4 Subject, predicate, and object [37].

The IEEE 1451 family of standards is composed mainly of some standards. The IEEE 1451.0, IEEE 1451.1, IEEE 1451.2, and IEEE 1451.5 as shown in Figure 4.5. The predicate means the relationship between the entities. For example, the TIM has TEDS means the TIM is composed of TEDS. Also, the TIM is an IEEE 1451.0 means that the TIM is a type of IEEE 1451.0, and the NCAP communicates based on the commands from the NCAP to the TIM.

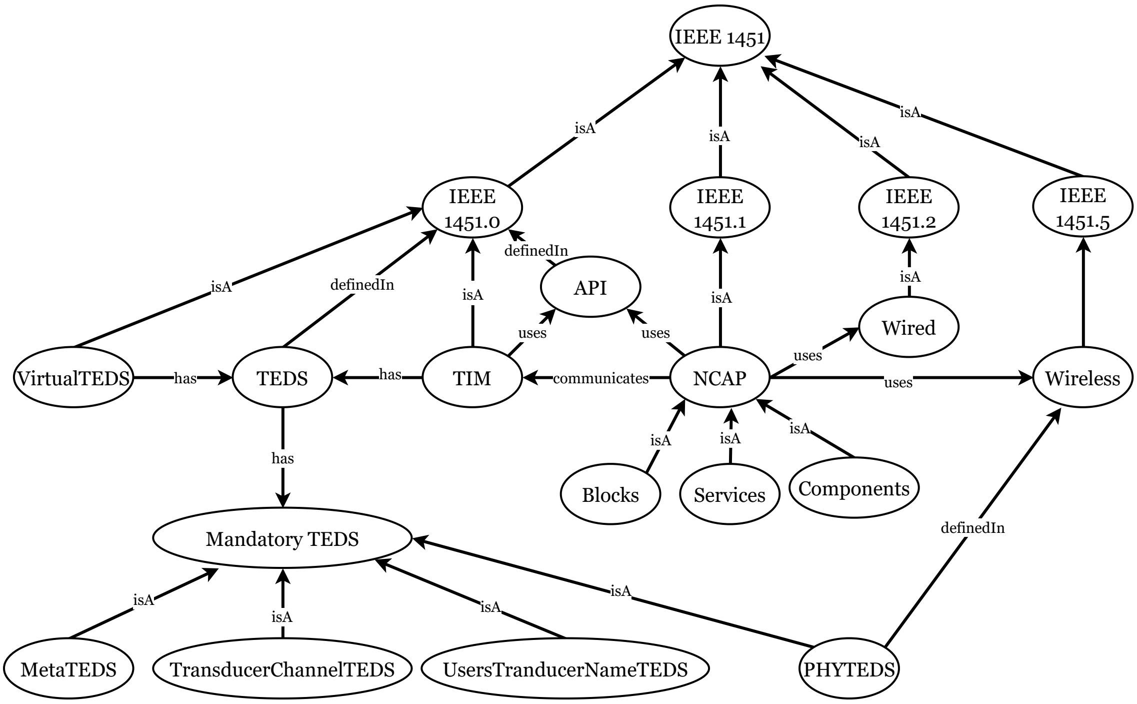


Figure 4.5 Basic IEEE 1451 Ontology.

IEEE 1451.0 describes the Transducer Interface Model (TIM). At the TIM are connected the sensors and actuators. The sensor also can be embedded inside a TIM. A TIM is described by a Transducer Electronic Data Sheet (TEDS). A TEDS can be stored inside a device that connects the transducers or stored outside a device as Virtual TEDS. There are four mandatory TEDS. A Meta TEDS stores information about the TEDS, its TEDS id, Universal Unique Identifier (UUID) from a TIM and the TEDS, the timeouts for request and response information from the TIM, the number of transducer channels implemented on the TIM, the geolocation of the TIM and for ensuring the information is correct is calculated a checksum to verify the information about the TIM after requesting the Meta TEDS. The following required TEDS is the TransducerChannel TEDS that stores information about the *TransducerChannels* (e.g., calibration, transducer channel type, low and high limits, error, sample), timing-related information (e.g., update, setup, period, warm up, delay), time of sample information (e.g., time source), attributes (e.g., sampling, sampling mode, data transmission, buffer mode). The Users Transducer Name TEDS is the third required TEDS and stores a transducer's system or end username. The fourth required TEDS is the PHY TEDS [39]. The PHY TEDS is defined in IEEE 1451.5. It is needed to approach any channel and store its information about the physical channels using wireless communication (e.g., radio throughput, connected devices, encryption, authentication, latency, simultaneous connections, battery, and radio version). It brings the Wireless TIM (WTIM) that can utilise one of the wireless connections (e.g., IEEE 802.11, Bluetooth, Zigbee, and

6LoWPAN) [40]. Whereas the IEEE 1451.2 handles the wired communication (e.g., SPI, UART) [41].

The second part of the IEEE 1451.0 standards presents the development of an Application Programming Interface (API). The API connects the TIM to the network through a Network Capable Application Processor (NCAP). The Transducer Service Interface focuses on measuring and controlling applications composed of NCAP and TIMs with methods to read and write *TransducerChannels* and TEDS, configure, control, and send commands to the TIM. The services provided by this API are TIM Discovery, Transducer Access, Transducer Management, and TEDS Management [39].

An NCAP is defined in the IEEE 1451.1 standard. It comprises objects divided into three types of classes: Block, Component, and Services, to allow access from the transducer connected to a TIM to a network providing a neutral application model [42]. An NCAP provides the connection through the network from an application to the API for accessing the transducers and TIMs by a supported application layer protocol, e.g., Hypertext Transfer Protocol (HTTP) for client-server and Message Queuing Telemetry Transport (MQTT) for publish/subscribe paradigm. An NCAP connects to the TIM by IEEE 1451.2 (wired) or the IEEE 1451.5 (wireless) standard.

A new ontology was based on the IEEE 1451.0, IEEE 1451.1, and IEEE 1451.5 standards since the other standards that compose the family of the standard are not compliant with the IEEE 1451.0.

The information gathering was designed as a diagram using the Unified Modelling Language (UML) [43], [44]. Representation of information or knowledge domains was done with UML. The scientific community using the UML language is very active and wide, ensuring the results will have a bigger impact. There are other languages to develop a graphic ontology model, such as the Integrated DEFinition Methods (IDEF5) ontology language [45]. The ontology development utilises the UML modelling languages for systems design. However, the UML terms change for the ontology development, such as the “association” becoming a “binary relation” term, as shown in Figure 4.6.

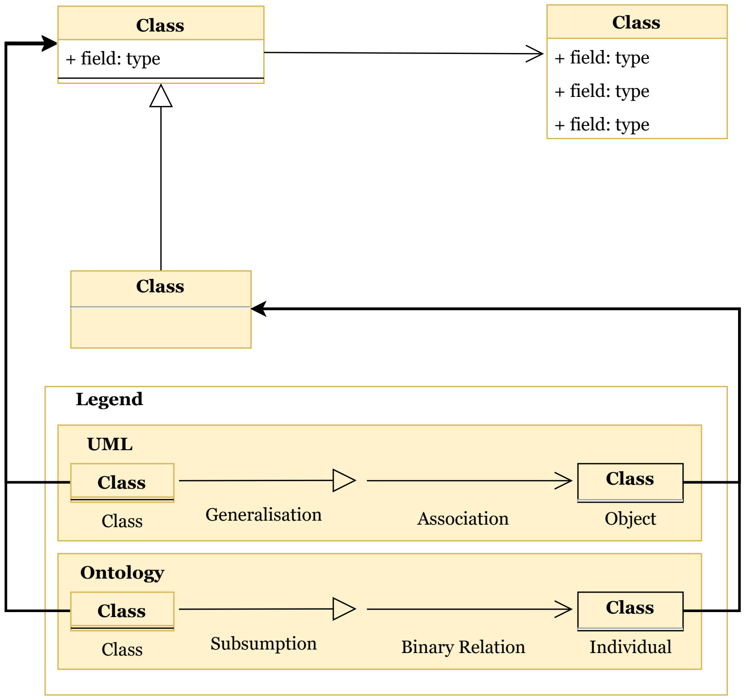


Figure 4.6 UML for ontology design.

A TIM defined in the IEEE 1451.0 is composed of state, transducers (sensors, event sensors, or actuators), the transducer has addresses to access them and a *TransducerChannel* responsible for the signal condition and conversion component related to the transducer, UUID, operating states, commands, and trigger for the *TransducerChannels* or TIMs. A *TransducerChannel* also has the operating state and status. The TIM can be placed in a device or be a Virtual TIM with the four mandatory TEDS. The TEDS are specified in Chapter 3. The IEEE 1451.0 defines the TEDS, TransducerChannels, the commands between an NCAP and a TIM, and the TIM internal states. The TEDS description in UML is shown in Figure 4.7. Other parts of the diagram are presented in Appendix A.

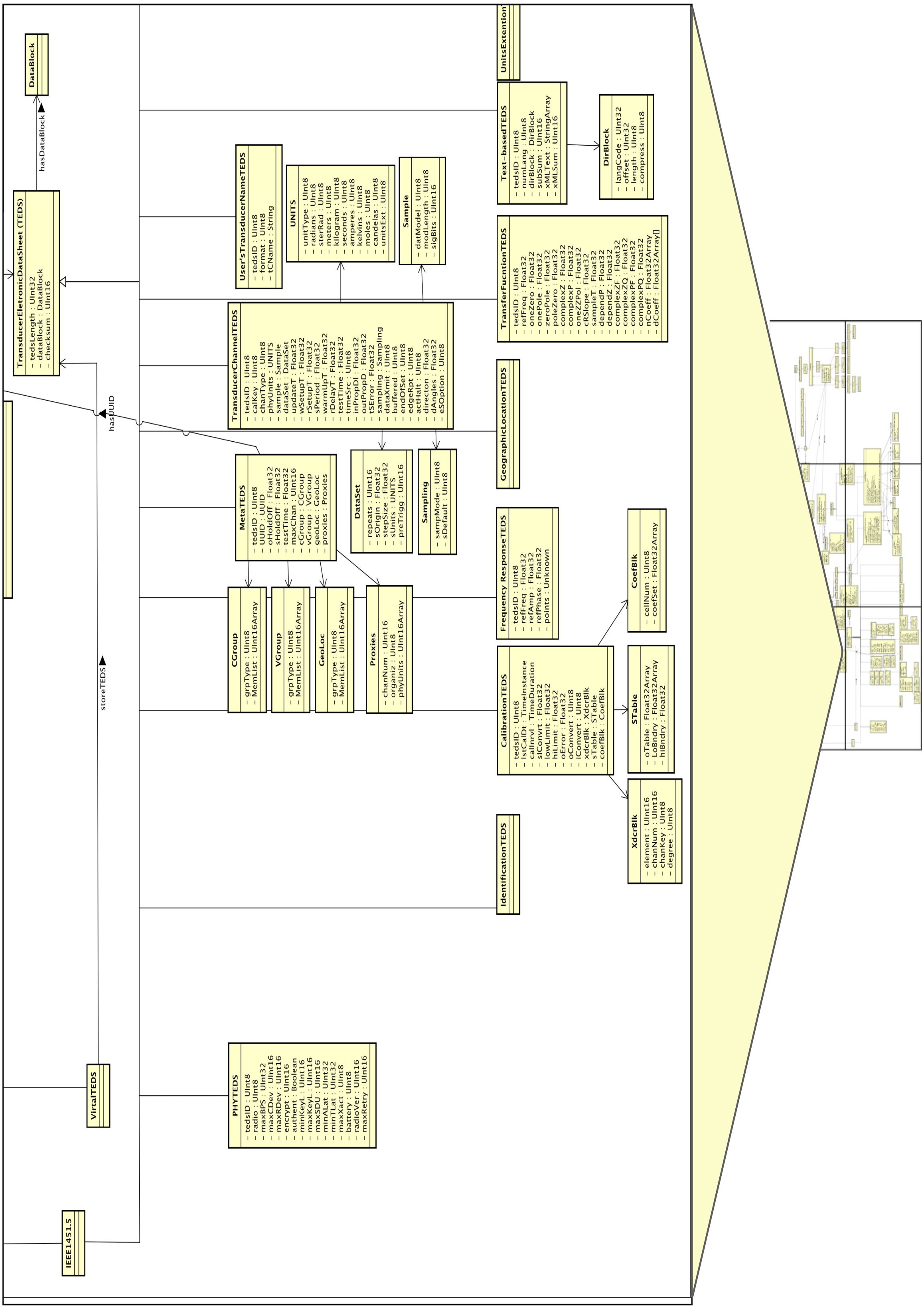


Figure 4.7 Knowledge modelling UML - TEDS

The NCAP communicate with the TIM by a message composed of the *TransducerChannel* address and commands (Command Class, Command Function and Command Independent). The message can or cannot have a replay. It depends on the type of command sent from the NCAP to the TIM. The NCAP is built using Object-Oriented Programming (OOP) with essential classes such as IEEE1451Root, IEEE1451Entity, IEEE1451Block, and IEEE1451NCAPBlock (the main class for the NCAP). The IEEE1451NCAPBlock can be or not be associated with the classes for conversion of the transducer data (IEEE 1451Component), communication with IEEE 1451.2 and IEEE 1451.5 (IEEE1451BaseTransducerBlock), and with the connection with the network by the (IEEE1451Services). The important attributes for the IEEE1451NCAPBlock are manufacture id, model number, software version, object properties, manufacture id, model number, serial number, and Operational System (OS) version. A complete implementation of an NCAP can be found in [46].

The NCAP receive the message, interprets it, and sends a command to the TIM through the IEEE 1451.0 API that promotes access from the network to the transducers. The API services allow the discovery and management of the TIMs and TEDSs presented connected with an NCAP. The answer message starts with the error code returned from the message request. There are two types of error codes. The code source represents an error in the communication, wired or wireless. The second is that the error code generated when accessing the TIM or the transducers is impossible.

4.3.3 Formalising

In the formalisation process is transferred the knowledge was modelled from the UML to using the OWL language and the Protégé software [24]. The IEEE 1451 ontology uses other ontologies, such as OWL for modelling, RDF, XML, and dcterms to store the data and permit the semantic relationship and annotation, and the XSD, that stores the default data types, as shown in Figure 4.8. Also, it is shown the metrics of the ontology. The focus of this ontology is to promote a base ontology for the user to apply to its use cases. Also, the ontology will be constant update.

Figure 4.8 IEEE 1451 ontology development.

The classes are inserted into the Protégé and create the class structure. The IEEE 1451.5 was removed since the start specification is defined inside the IEEE 1451.0. The semantic is added by the restriction given on the modelling process that allows the reasoner to infer about the data. A reasoner is a software that can infer logical sequences using the axioms employed inside an inference engine. The engine can utilise first-order logic using forward and backwards changing or probabilistic reasoners. There are many reasoners, such as Pallet, Hermit, FACT++, CEL, RACER, and ELK [47]. The reasoners are imported inside the Protégé as plugins. The Pallet was chosen for the ontology development since new data types do not significantly restrict it. Also, it is an open-source project. Pallet utilises Description Logic (DL) in its inference engine [48]. DL is a subset of first-order logic that models the relationship between concepts, roles, and individuals. The inference rules are the means of a description language [47]. The restrictions applied to the TIM are presented in Figure 4.9.

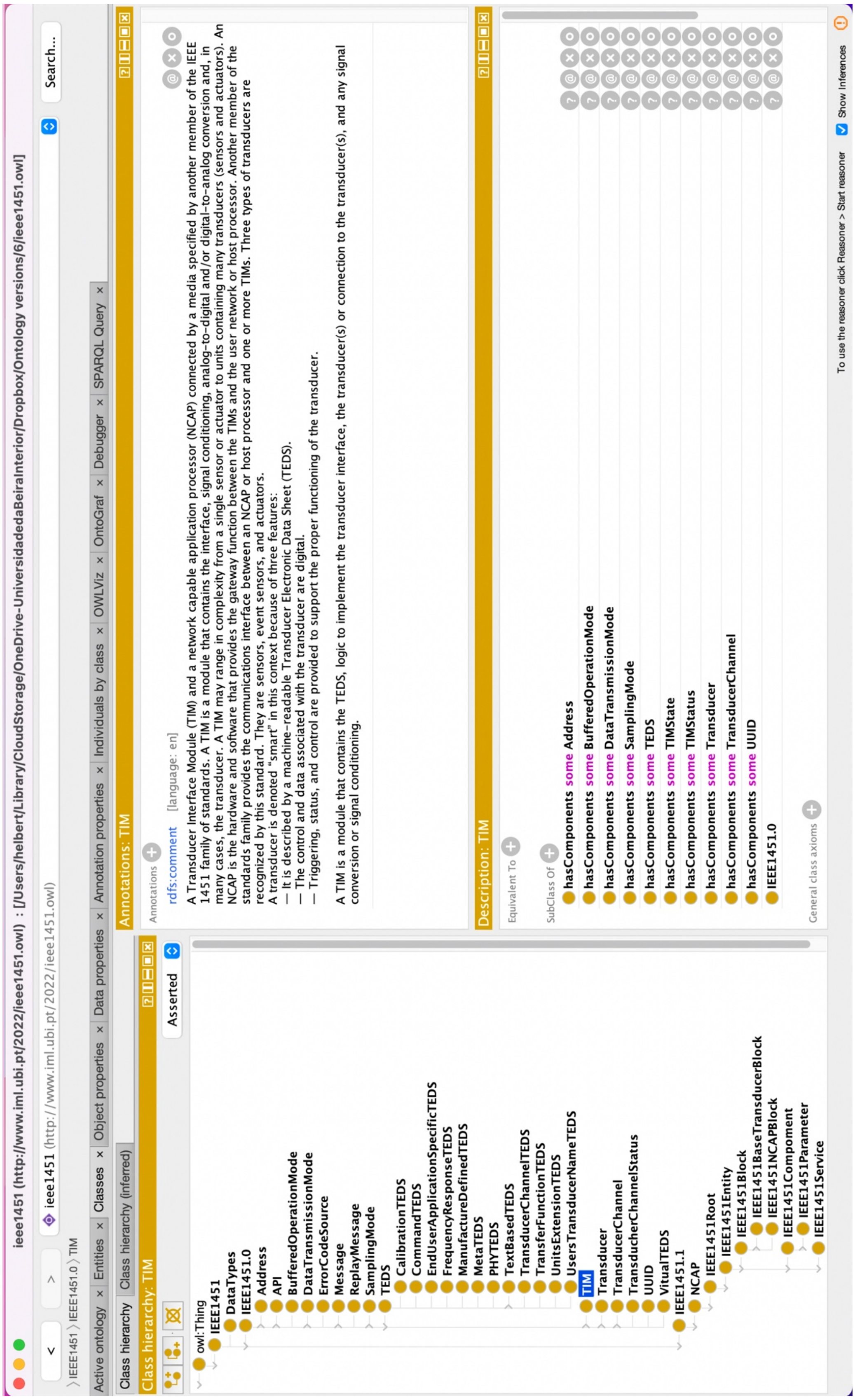


Figure 4.9 TIM restrictions.

The ontology restrictions are based on the class relationship, object properties and data properties, as shown in Figure 4.10.

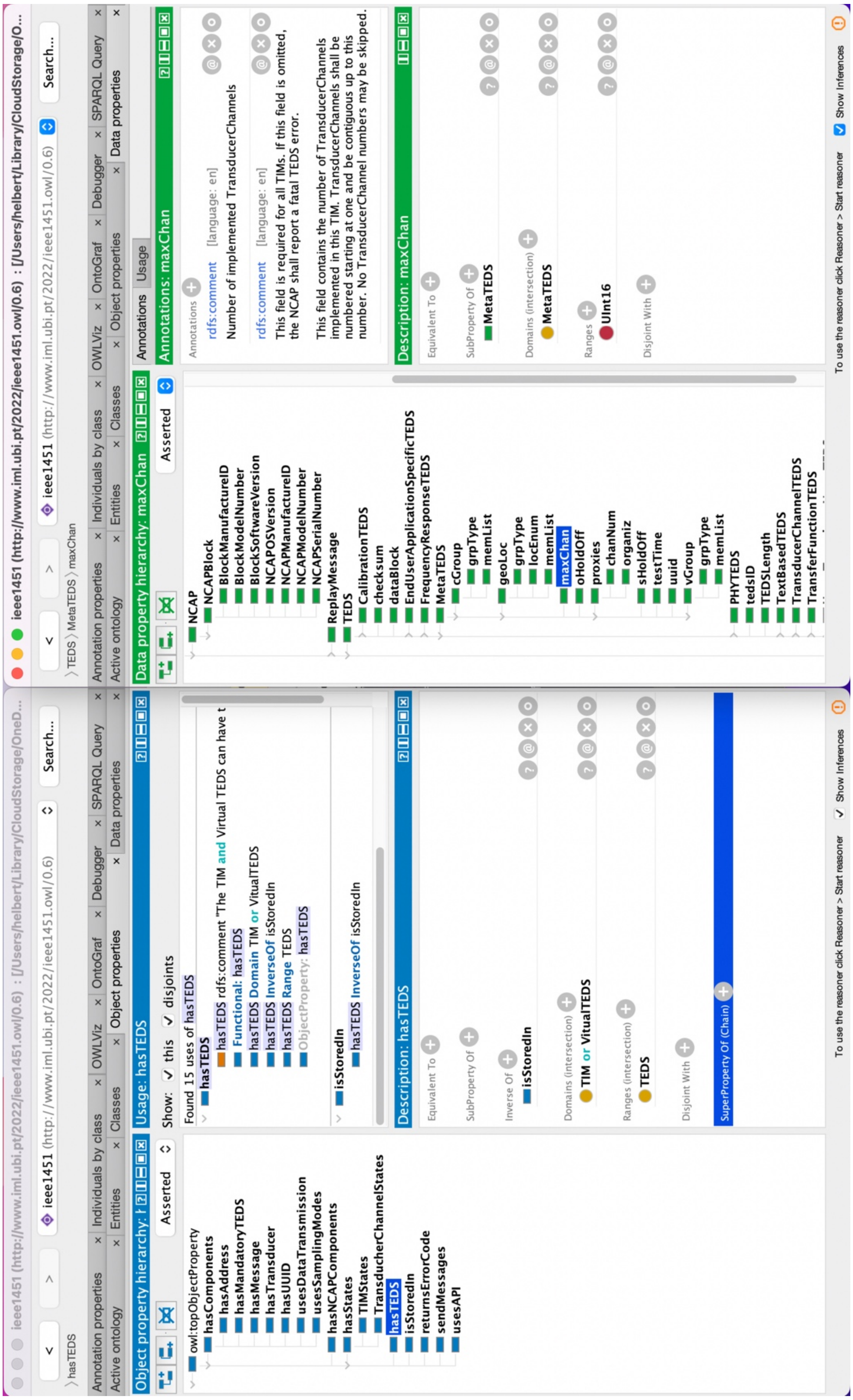


Figure 4.10 Ontology properties.

The reasoners help to verify the consistency and coherence of the ontology. It allows for keeping refactoring of the ontology in its development process. For example, it is possible to use the object properties to infer the kind of object it belongs to, as the NCAP and the NCAP block are shown in Figure 4.11.

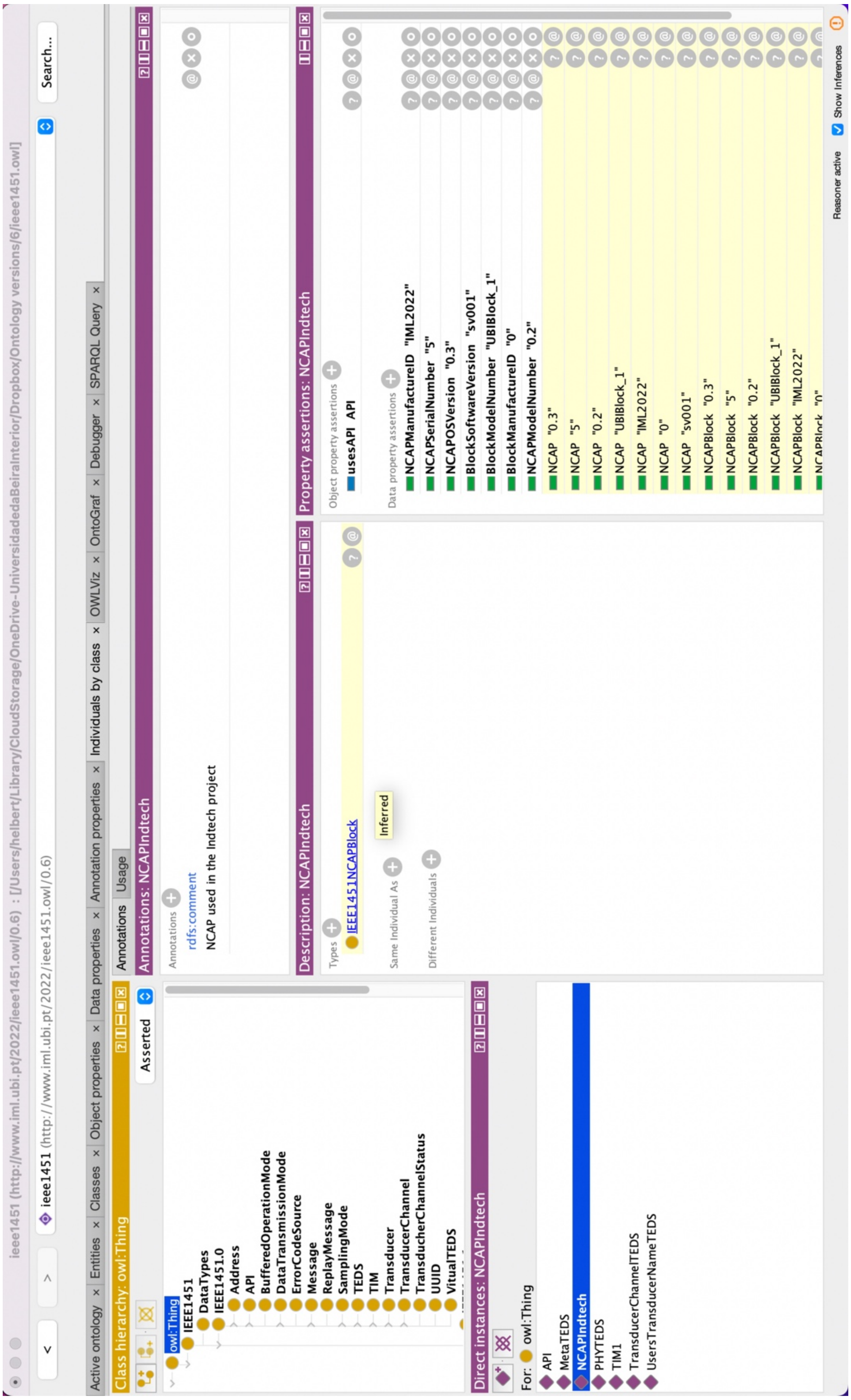


Figure 4.11 NCAP inference example.

4.3.4 Deploying

The vocabulary and ontology are available at the UBI at the address http://iml.ubi.pt/2022/ieee1451.owl. The Protégé allows saving in many formats. Based on the IEEE 1451 ontology defined on the UBI to store the ontology in the following formats: OWL, Turtle, RDF, JSON-LD, JSON-LD Context. These formats provide linked data based on ontology and achieve the semantic level of interoperability. A communication procedure is presented in Figure 4.12. The application demonstration reaches the top layers: logic, proof, and logic. The logic layer allows the writing of logic rules. The Proof executes these roles and evaluates with the Trust layer [25].

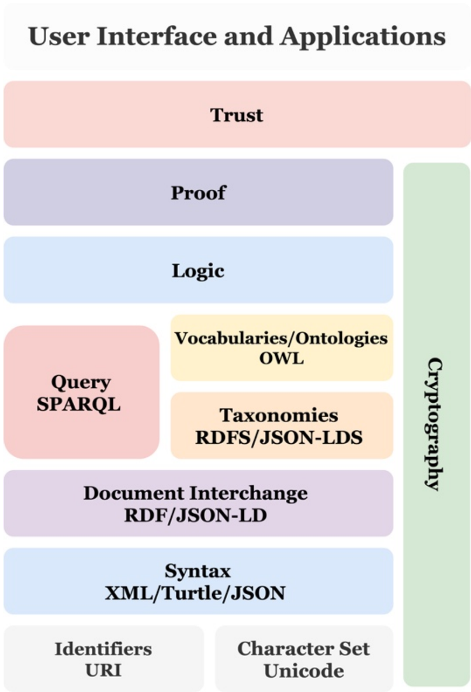


Figure 4.12 Semantic web stack adapted from [25].

The ontology documentation was developed using the Widoco software [49]. It is stored in http://iml.ubi.pt/2022/ieee1451, as shown in Figure 4.13. The documentation describes the classes, properties, data types and a visual representation of the ontology.

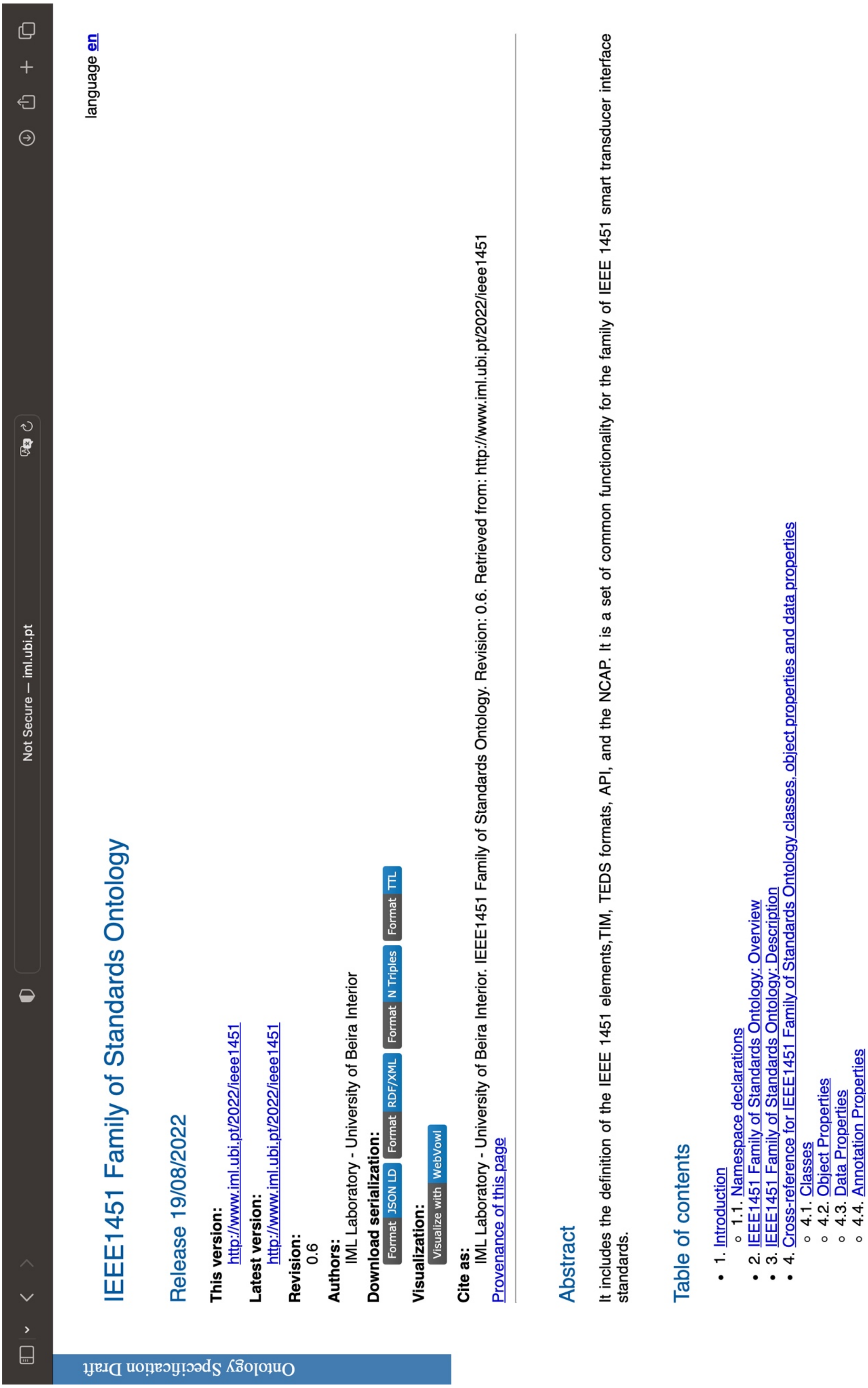


Figure 4.13 IEEE 1451 ontology documentation.

4.3.5 Evaluating

A Digital Twin (DT) was used for the INDTECH 4.0 project to test and evaluate the ontology. DT aims to be the virtual representation of the paint shop floor inside the manufacturing process. The JSON-LD format was chosen. JSON-LD allows the addition of linked data to JSON serialisation format, being it easy for humans to read and write [50]. It enables the creation of a context during communication. The IEEE 1451 context was developed for the message exchange. The defined serialisation format for the IEEE 1451 ontology is XML. The proposed semantic layer adds the JSON and JSON-LD to enable the semantic layer for communication using IEEE 1451. JSON is faster and lighter than XML for communication [51], [52].

An NCAP subscribes to the topic on the broker when it uses a publish/subscribe mechanism for message exchange. NCAP receive the message from the broker and translates it as a command to the TIM. The TIM responds to the command from the NCAP, it the data from the TIM, the NCAP analyses the response and creates a semantic response by the IEEE 1451 ontology and vocabulary stored in the UBI repository. The response at this moment is the JSON-LD based on the IEEE 1451 JSON-LD context, also stored in the UBI repository. After that, the NCAP publishes it using the corresponding topic defined by a response in the broker. The user receives the files and shows them in their application or case. It is a machine-to-machine communication that passes the data for another process. As shown in Figure 4.14.

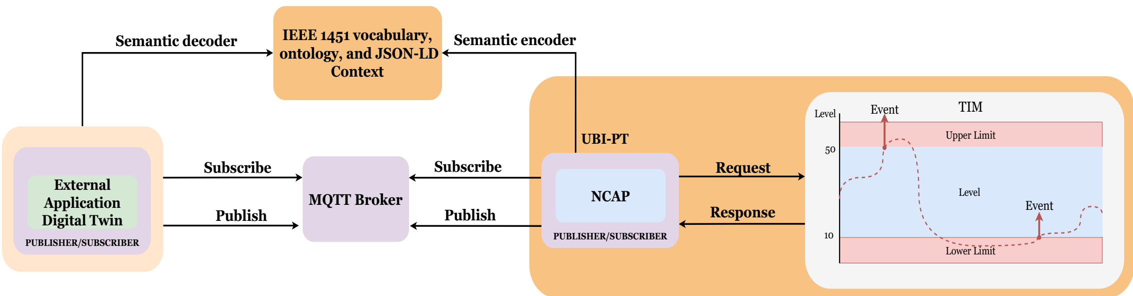


Figure 4.14 Validation scenario [53].

The actual communication for the IEEE 1451 family of standards is defined in the IEEE 1451.0 inside the API. The user can discover TIM and TransducerChannels, read, write, manage TEDS, and read TransducerChannels. For the TIM and TEDS, development utilised the IEEE 1451 planform developed inside the IML laboratory. This platform can easily create, read, update, and delete TEDS. The platform also allows for validation of the TIM and the TEDS. The TEDS development is shown in Figure 4.15 using the Project Editor. An MSP430F5529 experimental board is connected to a temperate sensor

TMP 36 and a water level sensor for the TIM to be utilised. Four transducer channels were implemented one for the temperate, one for the water level, and two for sensor limits as embedded event sensors. The NCAP was developed inside a Raspberry Pi 3 B+ programmed using Python. The Mosquitto broker was utilised as middleware for the publish/subscribe communication.

Figure 4.15 TEDS for INDTECH 4.0 project [53].

The communication algorithm is shown in Figure 4.16. The communication starts with the user's application that requests information about the system to the corresponding topic inside the broker. The NCAP subscribes to this topic and receives the message. The message is sent as a command to the TIM. The TEDS are read and stored inside the NCAP. The NCAP decode the information from the TLV value to the format specified on the standard and adds the semantic annotation using the IEEE1451 JSON-LD context. The message is serialised in JSON and sent over communication protocol. The user receives it, decodes it using a JSON-LD format tool, and uses the information received.

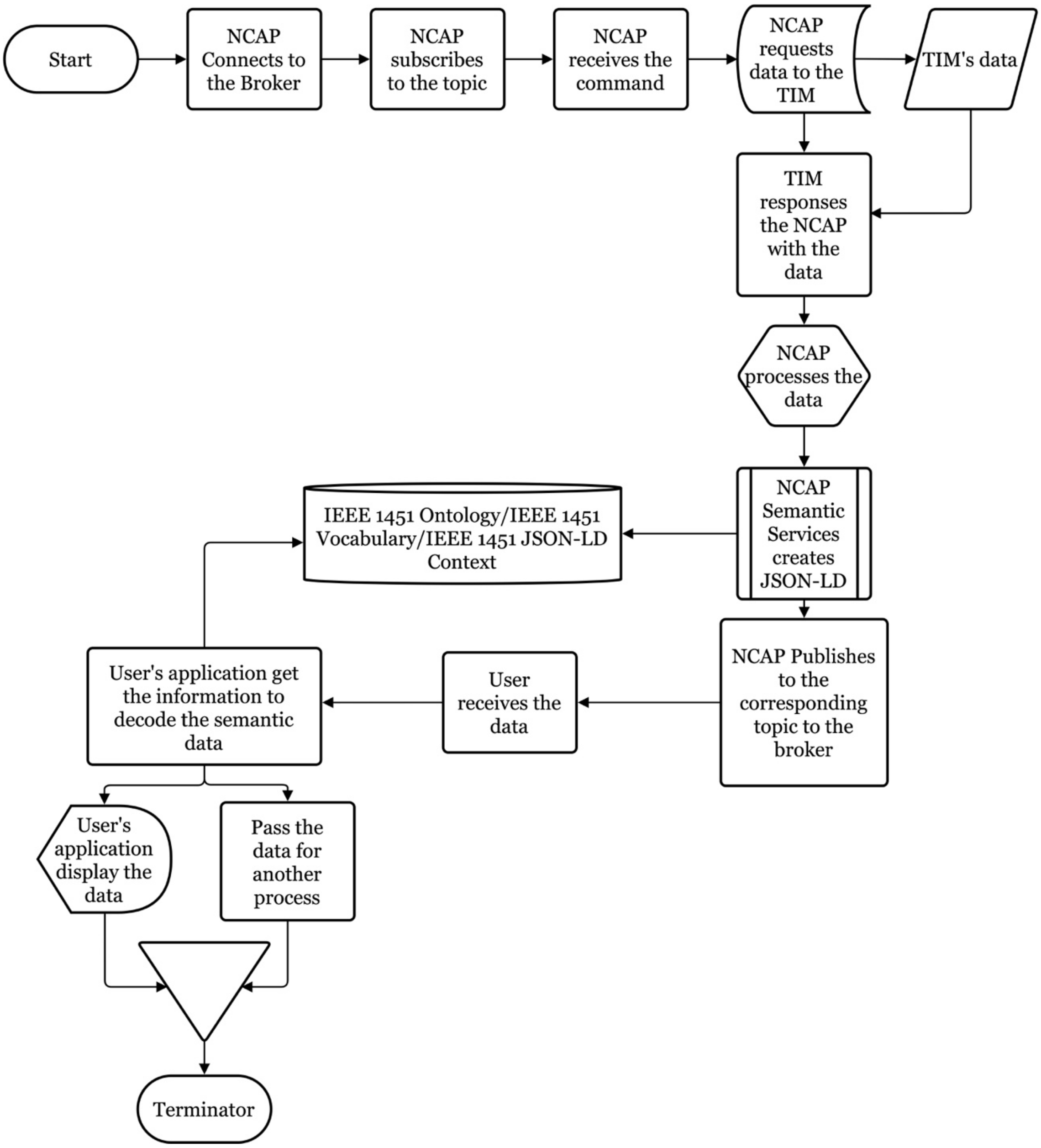


Figure 4.16 Semantic methodology [27].

At this moment, for the "ReadTEDS" command, the user using the publisher/subscribe paradigm the user publishes to the topic "IEEE1451/NCAPIndtech/TEDSManager/ReadTEDS/MetaTEDS", the NCAP receives it, interprets it as a command publishes the cached TEDS from the TIM to the related topic to the user. The users need to know the IEEE 1451.0 and Type/Length/Value (TLV) format to decode the message from the NCAP, as shown in Figure 4.17. For this application utilised the publish/subscribe paradigm. However, this methodology supports the client-server paradigm since the encoded and decoded independently of the communication protocol encodes and decodes independently.

[{
  "error_code" : "0",
  "timId" : "0x00 0x01",
  "channelId" : "0x00 0x01",
  "tedsType" : "0x01",
  "teds" : "0x00 0x00 0x00 0x2C 0x03 0x04 0x00 0x00 0x00 0x01 0x04 0x0A 0x91 0xB3 0x50 0x1A 0x69 0x05 0xF9 0x82 0xDA 0x79 0x0A 0x04 0x3F 0x80 0x00 0x00 0x0B 0x04 0x40 0x40 0x00 0x00 0x0C 0x04 0x3F 0x80 0x00 0x00 0x0D 0x02 0x00 0x04 0x0E 0x09 0x15 0x01 0x01 0x16 0x04 0x01 0x02 0x03 0x04 0xF8 0x0"
 }
]

Figure 4.17 Read Meta TEDS response.

The communication with the new proposed approach introduces semantics in the communication by the JSON-LD format. The proposed vocabulary and ontology service inside the NCAP parses the data to an understandable format for the users and machine-to-machine communication by the linked data that goes with the message. A user requests information about the NCAP and the TIMs, pushing a message to the related topic. The NCAP publishes the answer, for example, with the NCAP and TEDS information, as a “ReadTEDS” command before the information about the NCAP to the topic “IEEE1451/Discovery”, as shown in Figure 4.18. The complete example is in Appendix B. The semantic layer model uses JSON-LD to serialise with the IEEE1451 context. After this process, the NCAP sends the message to the corresponding topic in the broker. The user reives the message and uses the JSON-LD to decode the data. The data contain the linked data to the IEEE 1451 vocabulary store at the UBI.

```
{
  "@context": "http://iml.ubi.pt/2022/ieee1451/ieee1451.jsonld",
  "NCAP": {
    "BlockManufactureID": "0",
    "BlockModelNumber": "UBIBlock_1",
    "BlockSoftwareVersion": "sv001",
    "NCAPManufactureID": "IML2022",
    "NCAPModelNumber": "0.2",
    "NCAPSerialNumber": "5",
    "NCAPOSVersion": "0.3",
    "TIM":[{
      "timId": "1",
      "MetaTEDS": {
        "TEDSLength": "59",
        "tedsID ": "1",
        "UUID": "91B3501A6905F982DA79",
        "oHoldOff": "1",
        "sHoldOff": "1",
        "testTime": "1",
        "maxChan": "4",
        "cGroup": {
          "grpType": "1",
          "memList": "1"
        }
      }
    }]
  }
}
```


Figure 4.18 Meta TEDS semantic encoded.

The IEEE 1451 context was developed based on the IEEE 1451 ontology, where every field defined in the JSON-LD document corresponds to the field corresponding in the file that defines the IEEE 1451 ontology, creating the linked data for the formats and data information working as a vocabulary for the IEEE 1451 family of standards. Within this approach, the user or machine does not need to know the IEEE 1451 before the communication once the information is linked and only used when needed. The NCAP and Meta TEDS are presented as linked data from the JSON-LD, as shown in Figure 4.19.

```
{
 "@context": "http://iml.ubi.pt/2022/ieee1451/ieee1451.jsonld",
 "http://www.iml.ubi.pt/2022/ieee1451#NCAP": {
  "http://www.iml.ubi.pt/2022/ieee1451#BlockManufactureID": {
   "type": "http://www.iml.ubi.pt/2022/ieee1451#String",
   "@value": "0"
  },
  "http://www.iml.ubi.pt/2022/ieee1451#BlockModelNumber": {
   "type": "http://www.iml.ubi.pt/2022/ieee1451#String",
   "@value": "UBIBlock_1"
  },
  "http://www.iml.ubi.pt/2022/ieee1451#BlockSoftwareVersion": {
   "type": "http://www.iml.ubi.pt/2022/ieee1451#String",
   "@value": "sv001"
  },
  "http://www.iml.ubi.pt/2022/ieee1451#NCAPManufactureID": {
   "type": "http://www.iml.ubi.pt/2022/ieee1451#String",
   "@value": "IML2022"
  },
  "http://www.iml.ubi.pt/2022/ieee1451#NCAPModelNumber": {
   "type": "http://www.iml.ubi.pt/2022/ieee1451#String",
   "@value": "0.2"
  },
  "http://www.iml.ubi.pt/2022/ieee1451#NCAPOSVersion": {
   "type": "http://www.iml.ubi.pt/2022/ieee1451#String",
   "@value": "0.3"
  },
  "http://www.iml.ubi.pt/2022/ieee1451#NCAPSerialNumber": {
   "type": "http://www.iml.ubi.pt/2022/ieee1451#String",
   "@value": "5"
  },
  "http://www.iml.ubi.pt/2022/ieee1451#TIM": {
   "http://www.iml.ubi.pt/2022/ieee1451#MetaTEDS": {
    "http://www.iml.ubi.pt/2022/ieee1451#TEDSLength": {
     "type": "http://www.iml.ubi.pt/2022/ieee1451#UInt32",
     "@value": "59"
    },
    "http://www.iml.ubi.pt/2022/ieee1451#UUID": "91B3501A6905F982DA79",
    "http://www.iml.ubi.pt/2022/ieee1451#oHoldOff": {
     "type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
     "@value": "1"
    },
    "http://www.iml.ubi.pt/2022/ieee1451#sHoldOff": {
     "type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
     "@value": "1"
    },
    "http://www.iml.ubi.pt/2022/ieee1451#testTime": {
     "type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
```


Figure 4.19 Meta TEDS semantic decoded.

Two digital twin applications were developed to monitor and visualise the data collected from the sensors using the IEEE 1451 vocabulary. The first application was a DT for the water monitoring setup installed at the IML laboratory-UBI. The setup is shown in Figure 4.20. Figure 4.21 represents the DT developed based on the UBI setup. The Water Level Monitoring DT was the digital representation receiving the data from the physical setup. The 3D digital representation was developed using the three.js framework for the digital representation [54]. The DT application subscribes to the topics that the NCAP published the encoded using IEEE 1451 JSON-LD context to add the semantic information about the UBI setup. The DT application decoded the information using JSON-LD and keeps updating this information in real-time.

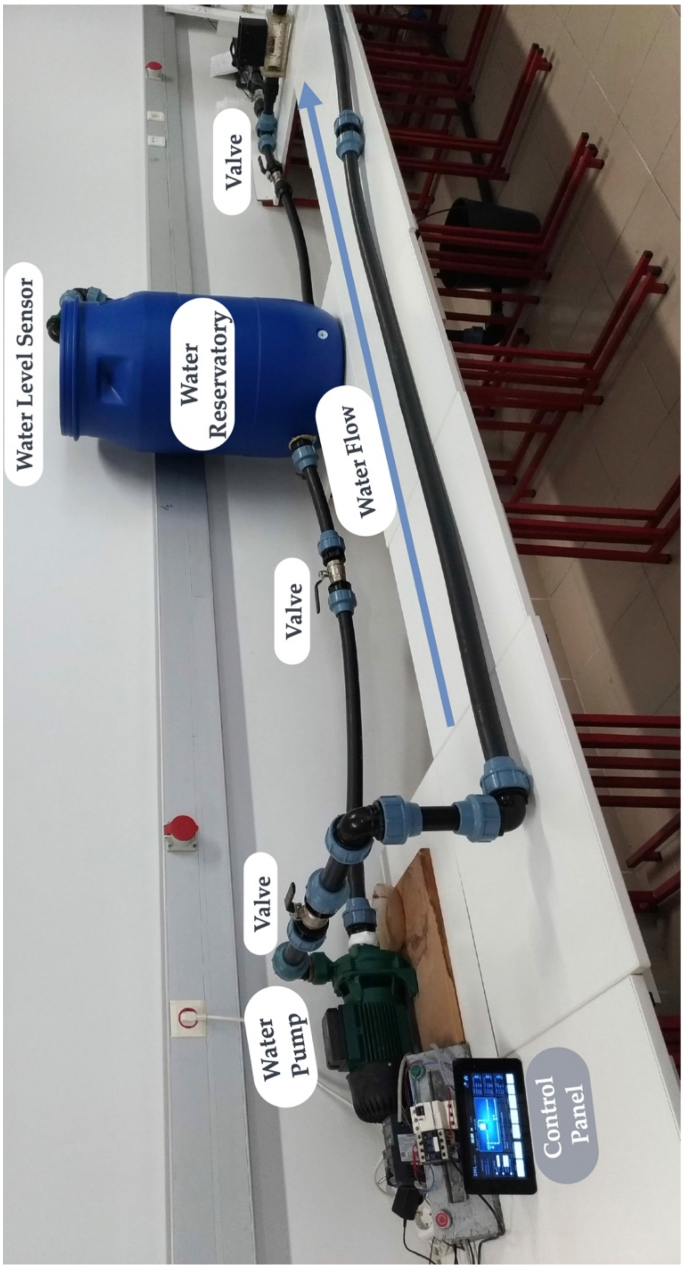


Figure 4.20 UBI water monitoring setup [53].

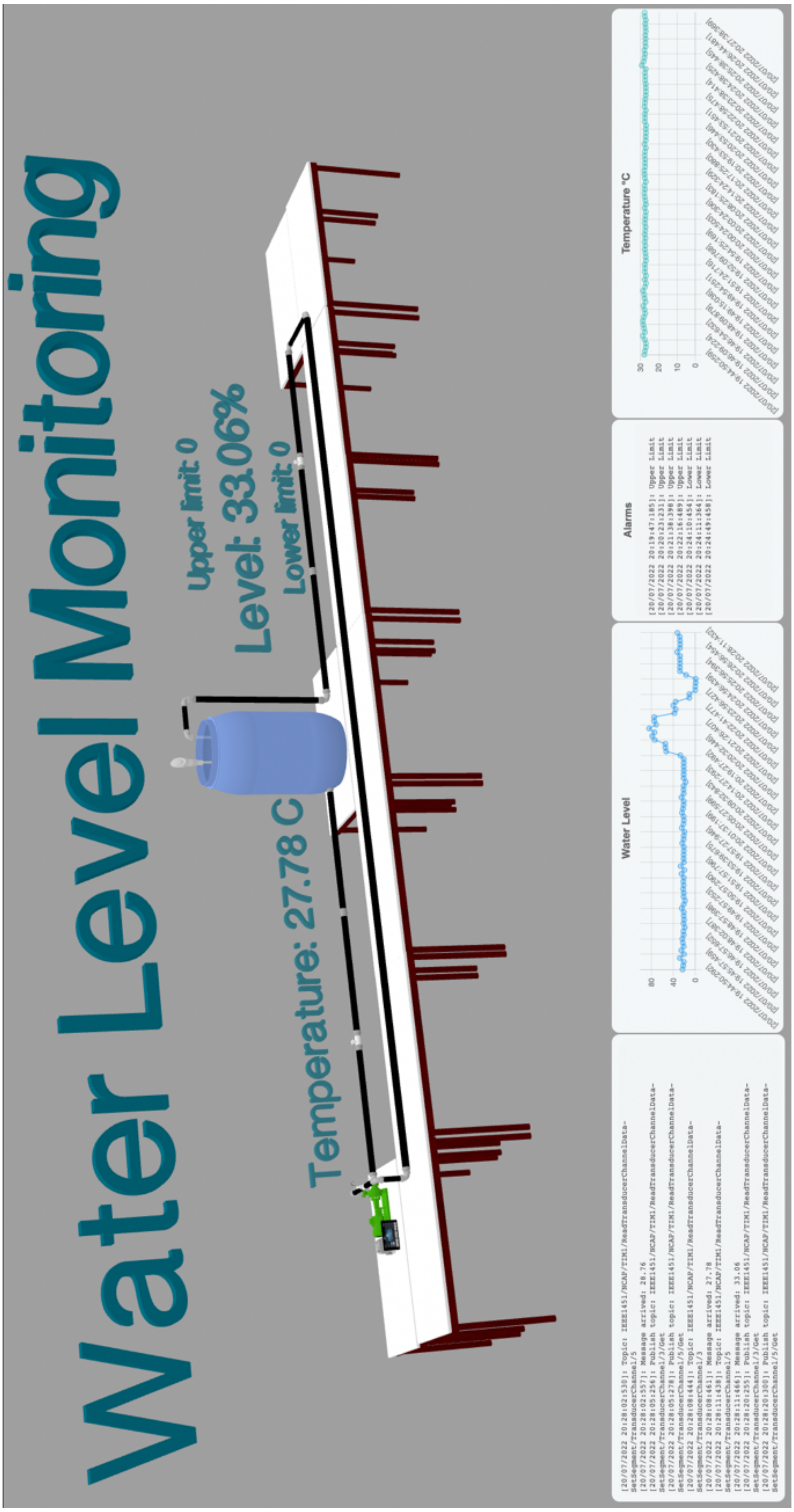


Figure 4.21 UBI Digital Twin water level monitoring [53].

The second DT application was developed for part of the INDTECH 4.0 project. A digital twin was built to simulate the painting process. A smaller version of the UBI setup was developed for the demonstration on the shop floor. The setup utilised for this demonstration is shown in Figure 4.22. The DT application for this setup also uses the IEEE 1451 JSON-LD context. It receives the data encoded using the JSON-LD format, decodes the information and shows it in real-time in the 3D model, as shown in Figure 4.23.

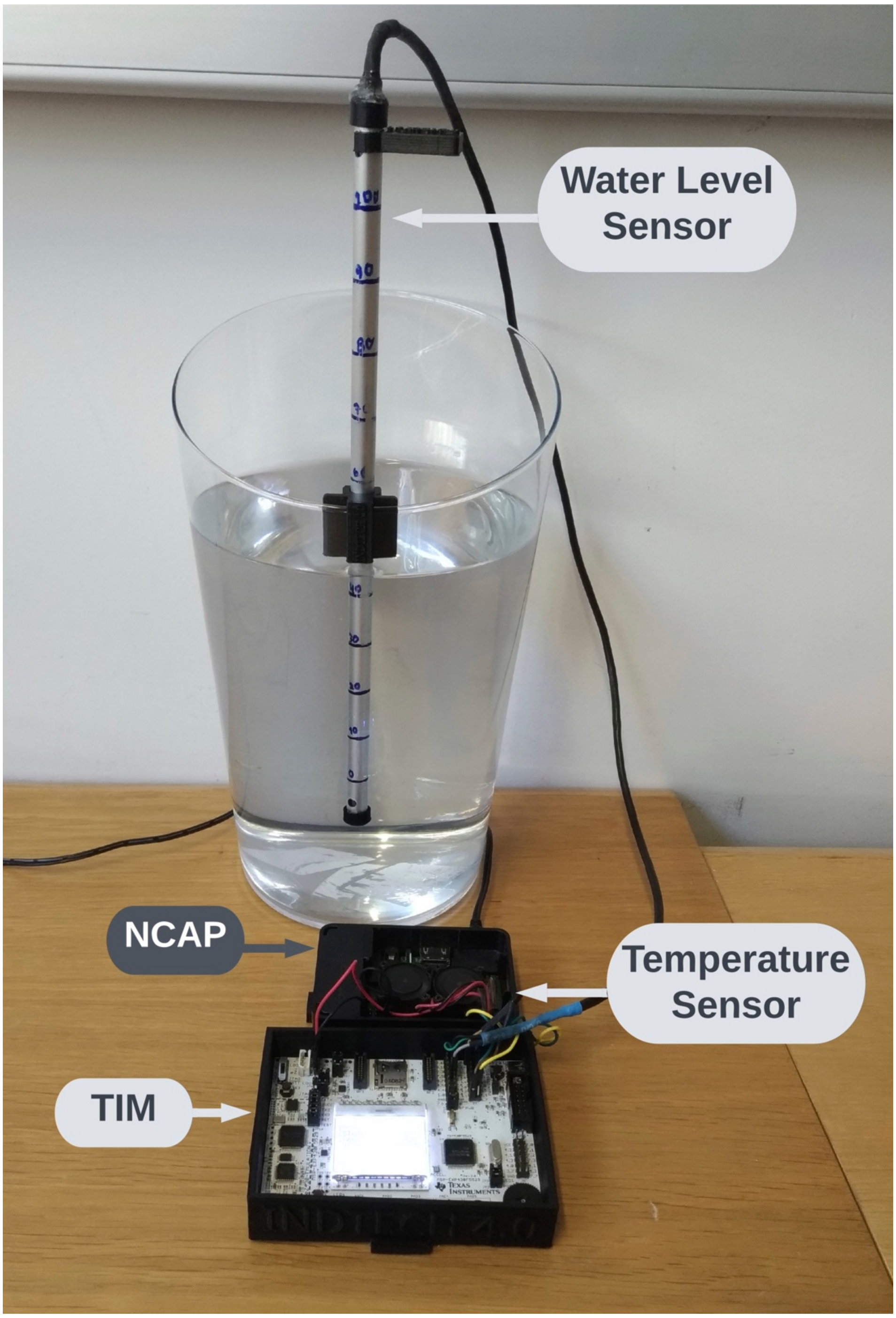


Figure 4.22 INDTECH 4.0 demonstration.

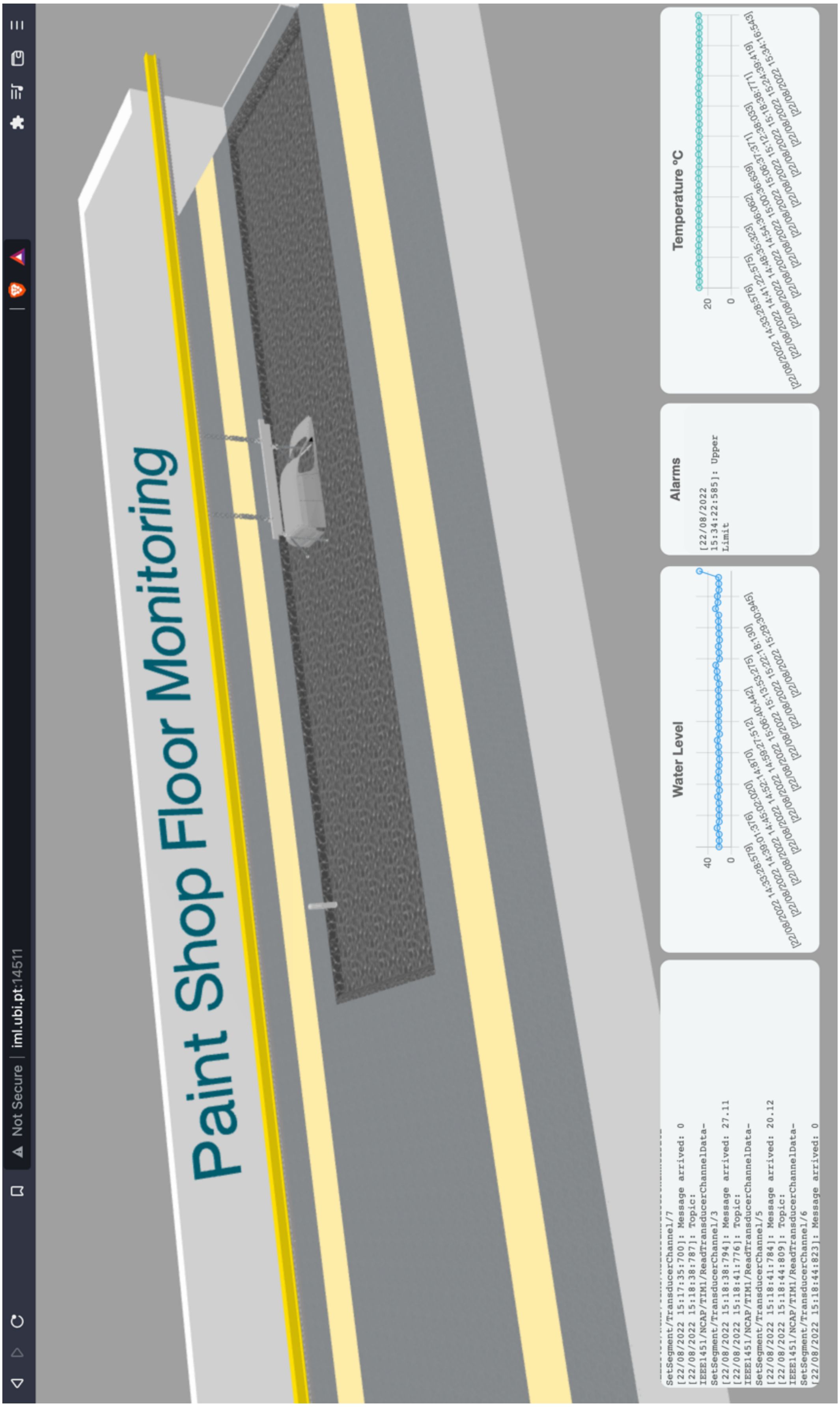


Figure 4.23 INDTECH 4.0 Digital Twin.

4.4 Conclusion

The proposed semantic layer for IEEE 1451 allows reaching the semantic level during communication inside an Industry 4.0 environment. Semantic allows messaging exchange unambiguously between the industrial elements. This allows the IEEE 1451 reaches a higher level of communication, being a complete framework for the industrial communication [55].

The Digital Twin developed for the INDTECH 4.0 project uses the vocabulary developed in this thesis to enable semantic communication. The data from the TIM are encoded by the NCAP using the JSON-LD serialisation format. The application decodes the information using the same format linking the data to the IEEE 1451 ontology file stored inside the UBI.

This approach suits Industry 4. 0 and the smart factory context. The heavyweight IEEE 1451 ontology proposed in this thesis can be used in different contexts, even in the IoT or IIoT. Helping to focus on disseminating and adopting the IEEE 1451 family of standards in these contexts. Lightweight ontologies can be developed using the same ontology and vocabulary based on this ontology.

References

[1] H. Van Der Veer and A. Wiles, "Achieving technical interoperability," *Eur. Telecommun. Stand. Inst.*, 2008.

[2] J. Nilsson and F. Sandin, "Semantic interoperability in Industry 4.0: survey of recent developments and outlook," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, 2018, pp. 127–132.

[3] R. M. Pereira, A. L. Szejka, and O. C. Junior, "Towards an information semantic interoperability in smart manufacturing systems: contributions, limitations and applications," *Int. J. Comput. Integr. Manuf.*, vol. 34, no. 4, pp. 422–439, 2021, doi: 10.1080/0951192X.2021.1891571.

[4] Y. Lu, X. Xu, and L. Wang, "Smart manufacturing process and system automation – A critical review of the standards and envisioned scenarios," *J. Manuf. Syst.*, vol. 56, pp. 312–325, 2020, doi: https://doi.org/10.1016/j.jmsy.2020.06.010.

[5] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial

communication: Automation networks in the era of the internet of things and industry 4.0,” *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 17–27, 2017.

[6] D. G. S. Pivoto, L. F. F. de Almeida, R. da Rosa Righi, J. J. P. C. Rodrigues, A. B. Lugli, and A. M. Alberti, “Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: A literature review,” *J. Manuf. Syst.*, vol. 58, pp. 176–192, 2021.

[7] S.-W. Lin, S. Mellor, H. Munz, and E. Barnstedt, “Architecture Alignment and Interoperability,” *Ind. Internet Consort.*, p. 17, 2017, Accessed: Dec. 07, 2019. [Online]. Available: https://www.iiconsortium.org/pdf/JTG2_Whitepaper_final_20171205.pdf.

[8] T. R. Gruber, “A translation approach to portable ontology specifications,” *Knowl. Acquis.*, vol. 5, no. 2, pp. 199–220, 1993.

[9] N. F. Noy, D. L. McGuinness, and others, “Ontology development 101: A guide to creating your first ontology.” Stanford knowledge systems laboratory technical report KSL-01-05 and~..., 2001.

[10] I. S. O. Iso, “IEC/IEEE International Standard-Systems and Software Engineering–Vocabulary,” ISO/IEC/IEEE 24765: 2017 (E), 2017.

[11] S. Beden, Q. Cao, and A. Beckmann, “Semantic Asset Administration Shells in Industry 4.0: A Survey,” in *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, 2021, pp. 31–38, doi: 10.1109/ICPS49255.2021.9468266.

[12] I. Grangel-González *et al.*, “The industry 4.0 standards landscape from a semantic integration perspective,” in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2017, pp. 1–8.

[13] S. R. Bader, I. Grangel-Gonzalez, P. Nanjappa, M.-E. Vidal, and M. Maleshkova, “A Knowledge Graph for Industry 4.0,” in *The Semantic Web*, 2020, pp. 465–480.

[14] S. Gil and G. D. Zapata-Madrigal, “Semantic Automation Systems, a Suitable Approach for Automation Networks in the Industry 4.0,” in *2019 IEEE 4th Colombian Conference on Automatic Control (CCAC)*, 2019, pp. 1–6.

[15] G. Kim, S. Kang, J. Park, and K. Chung, “An MQTT-based context-aware

autonomous system in oneM2M architecture," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8519–8528, 2019.

[16] J. Higuera and J. Polo, "Autonomous and Interoperable Smart Sensors for Environmental Monitoring Applications," in *Lecture Notes in Electrical Engineering*, vol. 146, 2012, pp. 323–359.

[17] M. Eid, R. Liscano, and A. El Saddik, "A Novel Ontology for Sensor Networks Data," in *2006 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, 2006, pp. 75–79, doi: 10.1109/CIMSA.2006.250753.

[18] F. Pu, Z. Wang, C. Du, W. Zhang, and N. Chen, "Semantic integration of wireless sensor networks into open geospatial consortium sensor observation service to access and share environmental monitoring systems," *IET Softw.*, vol. 10, no. 2, pp. 45–53, 2016, doi: https://doi.org/10.1049/iet-sen.2014.0141.

[19] A. Haller, K. Janowicz, S. D. Cox, D. Le Phuoc, K. Taylor, and M. Lefrançois, *Semantic Sensor Network Ontology*. 2017.

[20] K. Janowicz, A. Haller, S. J. D. Cox, D. Le Phuoc, and M. Lefrançois, "SOSA: A lightweight ontology for sensors, observations, samples, and actuators," *J. Web Semant.*, vol. 56, pp. 1–10, 2019, doi: https://doi.org/10.1016/j.websem.2018.06.003.

[21] M. Botts and A. Robin, "OpenGIS®Sensor Model Language (SensorML) Implementation Specification. Version 1.0. 0.," 2007, doi: DOI: http://dx.doi.org/10.25607/OBP-646.

[22] S. Cox, "Observations and Measurements-XML Implementation. Version 2.0.," 2011.

[23] W. Yun, X. Zhang, Z. Li, H. Liu, and M. Han, "Knowledge modeling: A survey of processes and techniques," *Int. J. Intell. Syst.*, vol. 36, no. 4, pp. 1686–1720, 2021, doi: https://doi.org/10.1002/int.22357.

[24] M. A. Musen, "The protégé project: a look back and a look forward," *{AI} Matters*, vol. 1, no. 4, pp. 4–12, 2015, doi: 10.1145/2757001.2757003.

[25] D. L. McGuinness, F. Van Harmelen, and others, "OWL web ontology language

overview," *W3C Recomm.*, vol. 10, no. 10, p. 2004, 2004.

[26] World Wide Web Consortium, "OWL 2 Web Ontology Language Document Overview (Second Edition)." https://www.w3.org/TR/owl2-overview/ (accessed Aug. 16, 2022).

[27] H. Da Rocha, A. Espírito-Santo, and R. Abrishambaf, "Semantic Level of Interoperability by Proposing an IEEE 1451 Family of Standards Ontology," in *IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society*, 2022, pp. 1–6, doi: 10.1109/IECON49645.2022.9968883.

[28] A. Polenghi, I. Roda, M. Macchi, A. Pozzetti, and H. Panetto, "Knowledge reuse for ontology modelling in Maintenance and Industrial Asset Management," *J. Ind. Inf. Integr.*, vol. 27, p. 100298, 2022, doi: https://doi.org/10.1016/j.jii.2021.100298.

[29] P. Bierly, E. Kessler, and E. Christensen, "Organizational Learning, Knowledge and Wisdom," *J. Organ. Chang. Manag.*, vol. 13, pp. 595–618, 2000, doi: 10.1108/09534810010378605.

[30] D. Li, Å. Fast-Berglund, and D. Paulin, "Current and future Industry 4.0 capabilities for information and knowledge sharing," *Int. J. Adv. Manuf. Technol.*, vol. 105, no. 9, pp. 3951–3963, 2019, doi: 10.1007/s00170-019-03942-5.

[31] D. Li, A. Landström, Å. Fast-Berglund, and P. Almström, "Human-Centred Dissemination of Data, Information and Knowledge in Industry 4.0," *Procedia CIRP*, vol. 84, pp. 380–386, 2019, doi: https://doi.org/10.1016/j.procir.2019.04.261.

[32] C. Zhang, X. Yang, and S. Du, "A Distributed Knowledge Model for Knowledge Management System," in *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, 2008, pp. 1–4, doi: 10.1109/WiCom.2008.2699.

[33] O. Sebubi, I. Zlotnikova, and H. Hlomani, "A Lightweight Version of National Performance Indicator Ontology (NPIonto)," in *2019 Conference on Next Generation Computing Applications (NextComp)*, 2019, pp. 1–6, doi: 10.1109/NEXTCOMP.2019.8883619.

[34] B. Ramis Ferrer *et al.*, “Comparing ontologies and databases: a critical review of lifecycle engineering models in manufacturing,” *Knowl. Inf. Syst.*, vol. 63, no. 6, pp. 1271–1304, 2021, doi: 10.1007/s10115-021-01558-4.

[35] R. Li, T. Mo, J. Yang, S. Jiang, T. Li, and Y. Liu, “Ontologies-Based Domain Knowledge Modeling and Heterogeneous Sensor Data Integration for Bridge Health Monitoring Systems,” *IEEE Trans. Ind. Informatics*, vol. 17, no. 1, pp. 321–332, 2021, doi: 10.1109/TII.2020.2967561.

[36] T. Chungoora, “Practical Knowledge Modelling: Ontology Development 101.” https://www.udemy.com/course/practical-knowledge-modelling/ (accessed Mar. 12, 2021).

[37] O. Lassila, R. R. Swick, W. Wide, and W. Consortium, “Resource Description Framework (RDF) Model and Syntax Specification.” 1998.

[38] B. McBride, “The Resource Description Framework (RDF) and its Vocabulary Description Language RDFS,” in *Handbook on Ontologies*, S. Staab and R. Studer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 51–65.

[39] I. S. IEEE 1451.0, “IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats,” *IEEE Std 1451.0-2007*, pp. 1–335, 2007, doi: 10.1109/IEEESTD.2007.4338161.

[40] “IEEE Standard for a Smart Transducer Interface for Sensors and Actuators Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats,” *IEEE Std 1451.5-2007*, pp. C1-236, 2007, doi: 10.1109/IEEESTD.2007.4346346.

[41] “IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats,” *IEEE Std 1451.2-1997*, pp. 1–120, 1998, doi: 10.1109/IEEESTD.1998.88285.

[42] I. 2010 IEEE 21451-1:, “ISO/IEC/IEEE Information technology – Smart transducer interface for sensors and actuators – Part 1: Network Capable Application Processor (NCAP) information model,” *ISO/IEC/IEEE 21451-1:2010(E)*, pp. 1–356, 2010, doi: 10.1109/IEEESTD.2010.5668469.

[43] S. Cranefield and M. Purvis, “UML as an ontology modelling language,” 1999.

[44] G. Guizzardi, G. Wagner, and H. Herre, “On the Foundations of UML as an Ontology Representation Language BT - Engineering Knowledge in the Age of the Semantic Web,” 2004, pp. 47–62.

[45] Y. Ye, D. Yang, Z. Jiang, and L. Tong, “Ontology-based semantic models for supply chain management,” *Int. J. Adv. Manuf. Technol.*, vol. 37, no. 11, pp. 1250–1260, 2008, doi: 10.1007/s00170-007-1052-6.

[46] A. Stepanenko, K. Lee, R. Kochan, V. Kochan, and A. Sachenko, “Development of a minimal IEEE 1451.1 model for microcontroller implementation,” in *Proceedings of the 2006 IEEE Sensors Applications Symposium, 2006.*, 2006, pp. 88–93.

[47] S. Abburu, “A survey on ontology reasoners and comparison,” *Int. J. Comput. Appl.*, vol. 57, no. 17, 2012.

[48] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, “Pellet: A practical OWL-DL reasoner,” *J. Web Semant.*, vol. 5, no. 2, pp. 51–53, 2007, doi: https://doi.org/10.1016/j.websem.2007.03.004.

[49] D. Garijo, “WIDOCO: a wizard for documenting ontologies,” in *International Semantic Web Conference*, 2017, pp. 94–102, doi: 10.1007/978-3-319-68204-4_9.

[50] M. Sporny, D. Longley, G. Kellogg, M. Lanthaler, P.-A. Champin, and N. Lindström, “JSON-LD 1.1--a JSON-Nased Serialization for Linked Data,” W3C, 2019.

[51] G. Goyal, K. Singh, and K. R. Ramkumar, “A detailed analysis of data consistency concepts in data exchange formats (JSON & XML),” in *2017 International Conference on Computing, Communication and Automation (ICCCA)*, 2017, pp. 72–77, doi: 10.1109/CCAA.2017.8229774.

[52] A.-R. Breje, R. Gyorödi, C. Gyorödi, D. Zmaranda, and G. Pecherle, “Comparative study of data sending methods for xml and json models,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 12, 2018.

[53] H. da Rocha, J. Pereira, R. Abrishambaf, and A. Espirito Santo, “An Interoperable

Digital Twin with the IEEE 1451 Standards," *Sensors*, vol. 22, no. 19, 2022, doi: 10.3390/s22197590.

[54] "Three.js – JavaScript 3D Library." https://threejs.org/ (accessed Aug. 30, 2022).

[55] R. Joshi, P. Didier, J. Jimenez, and T. Carey, "The Industrial Internet of Things Volume G5: Connectivity Framework," *Ind. Internet Consort.* , 2017, Accessed: Aug. 22, 2022. [Online]. Available: https://www.iiconsortium.org/pdf/IIC_PUB_G5_V1.0_PB_20170228.pdf.

Chapter 5

Conclusion and Future Work

5.1 Final Remarks

This thesis proposes a semantic layer inside the IEEE 1451 family of standards to promote the semantic level of interoperability in an Industry 4.0 context. The main contribution was building a vocabulary and ontology based on the IEEE 1451 standards that allow users and machines to communicate and interoperate inside the industry. This semantic layer permits the IEEE 1451 to be a framework to manage transducers from the data acquisition to the communication to other standards and applications.

Chapter 2 introduces the conceptualisation of the Internet of Things, Industrial Internet of Things, Industry 4.0, Cyber-Physical Systems, Cyber-Physical Production Systems, and Digital Twin. Those concepts are important to place the work developed inside a correct context. The levels of Interoperability as shown in this Chapter. The syntactic and semantic levels are the required interoperability for an Industry 4.0 context.

Chapter 3 presents the syntactical level of interoperability reached at the current state of the IEEE 1451 family of standards. The tests proved that the IEEE 1451 standards could interoperate with other standards, such as the IEC 61499. The implementation of the syntactical level of interoperability was validated beyond the IEC 61499 standard, with the other implementation at the INTEROP event promoted by the IEEE Industrial and Electronics Society. Based on the validation promoted by the syntactical levels of interoperability, it was realised that the level of interoperability that the IEEE 1451 needs to reach the semantic level of interoperability. However, to reach the semantic level of interoperability employing the IEEE 1451.0-2007 standard, it needs to use a framework, such as OPC UA or oneM2M. It turns development into a complex task. The developer needs to learn the IEEE 1451 and the other standard to interoperate between them.

The vocabulary and ontology developed based on the IEEE 1451 applied inside the semantic layer proposed for the NCAP inside the IEEE 1451 standards solve this problem. The semantic layer was proposed in Chapter 4. With the employment of linked data is possible to interoperate with other users and standards without knowing the IEEE 1451 standards. The user and the application receive the data using the context developed for IEEE 1451 that links to the IEEE 1451 vocabulary. In this vocabulary are stored the component and its meaning inside the IEEE 1451 standards with the data type used by each property inside the standard. Using ontology is possible to see the relationships between the elements that compose the IEEE 1451 family of standards. It reduces learning time and uses the standard in different scenarios and contexts. The ontology developed focused on transducer management and communication, allowing interoperability between smart devices. This ontology utilises elements and properties from other ontologies in its development, and the concepts, elements, properties, and relations defined in the IEEE 1451 proposed ontology can be utilised inside other ontologies, such as the Industry 4.0 Knowledge Graph Ontology, a complete ontology for the IIoT and Industry 4.0.

The heavyweight ontology developed based on the IEEE 1451 contains the semantic annotation and relationships that can be used for a user to develop a lightweight ontology based on their application scenario. It also allows a reasoner to infer the data utilised in the context. For example, machines can utilise the reasoner to infer context inside an industrial context. The semantic level reaches the framework level required for a reference architecture model, as shown in Figure 5.1.

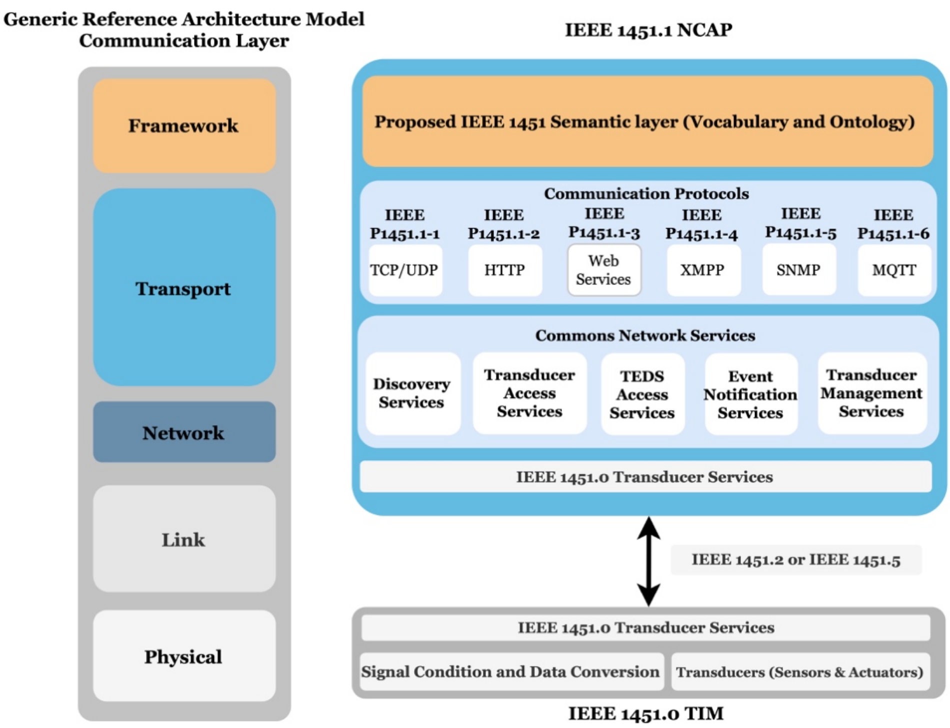


Figure 5.1 Proposed IEEE 1451 Semantic layer.

For IEEE 1451 Semantic Layer validation, a Digital Twin was developed. The Digital Twin is the digital representation of the Cyber-Physical Production Systems. The water management system simulates a painting shop floor in the manufacturing process in the INDTECH 4.0 project. The Digital Twin permits visualising and monitoring of the shop floor in real-time. It benefits the maintenance and production efficiency of the enterprise once an error can be prevented and corrected as faster as possible.

5.2 Future Works

For future work, some directions can be addressed:

- Ontology development is a continuous process. The improvements in the vocabulary developed during the thesis will be continued.
- The IEEE 1451 standards are in revision. When this revision is published, update the new version of the IEEE 1451.0-2007 standard to the IEEE P1451.0 standard, along with studying and adapting to the new standards being developed for the IEEE 1451 family of standards.
- Another essential feature of Industry 4.0 is security. Study and implement more security protocols and mechanisms besides those provided by the standard, such as blockchain engine, TLS v1.3 supported protocols and handshakes.
- Artificial intelligence is becoming a tool for everyday use, employing artificial intelligence to improve the proposed ontology and specific use cases.
- Develop a low-code or no-code graphics platform to help develop new ontologies based on the IEEE 1451 family of standards, such as the Node-RED, that enable users to drag and drop elements to generate the code.

5.3 Publications

The doctoral research work behind this thesis has originated from the following publications:

- **An Interoperable Digital Twin with IEEE 1451 Standards**

  H. da Rocha, J. Pereira, R. Abrishambaf, and A. Espirito Santo, *Sensors*, vol. 22, no. 19, p. 7590, Oct. 2022, doi: 10.3390/s22197590.

- **Smart Transducers Promoting Smart Cities Interoperability**

  M. Moreira, H. Da Rocha, J. D. Pereira, J. Salvado and A. E. Santo, *IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society*, Brussels, Belgium, 2022, pp. 1-6, doi: 10.1109/IECON49645.2022.9968431.

- **A Semantic Level of Interoperability by a Proposed IEEE 1451 Family of Standards Ontology**

  H. Da Rocha, A. Espírito-Santo and R. Abrishambaf, *IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society*, Brussels, Belgium, 2022, pp. 1-6, doi: 10.1109/IECON49645.2022.9968883.

- **An Energy-Efficient Process for Optimal Communication Synchronization in Low Power Wireless Smart Sensors**

  H. da Rocha, J. L. D. Pereira, T. A. G. N. Rodrigues, J. A. Salvado and A. Espirito-Santo, *2022 IEEE International Symposium on Measurements & Networking (M&N)*, Padua, Italy, 2022, pp. 1-6, doi: 10.1109/MN55117.2022.9887722.

- **Integrating the IEEE 1451 and IEC 61499 Standards with the Industrial Internet Reference Architecture**

  H. da Rocha, R. Abrishambaf, J. Pereira, and A. Espirito Santo, Sensors, vol. 22, no. 4, p. 1495, Feb. 2022, doi: 10.3390/s22041495.

- **IEC 61499 and IEEE 1451 for Distributed Control and Measurement Systems**

  R. Abrishambaf, H. Da Rocha and A. Espírito-Santo, *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, Toronto, ON, Canada, 2021, pp. 1-6, doi: 10.1109/IECON48115.2021.9589997.

- **Semantic Interoperability in the Industry 4.0 Using the IEEE 1451 Standard**

  H. da Rocha, A. Espirito-Santo and R. Abrishambaf, *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, Singapore, 2020, pp. 5243-5248, doi: 10.1109/IECON43393.2020.9254274.

- **A Platform for IEEE 1451 Standard's Education, Development and Validation for Industry 4.0**

  J. L. D. Pereira, H. da Rocha and A. E. Santo, *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Dubrovnik, Croatia, 2020, pp. 1-6, doi: 10.1109/I2MTC43012.2020.9129114.

- **A Discussion about the Implementation of a WSN to Industry 4.0 based on the IEEE 1451 Standard**

  R. Pinto, J. Pereira, H. da Rocha, R. I. Martin and A. Espírito-Santo, *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, Helsinki, Finland, 2019, pp. 1573-1578, doi: 10.1109/INDIN41052.2019.8972222.

Appendix A

IEEE 1451 Knowledge Model

The IEEE 1451 knowledge model developed using UML was divided into six parts for better visualisation, as shown in Figure A.1 to Figure A.6.

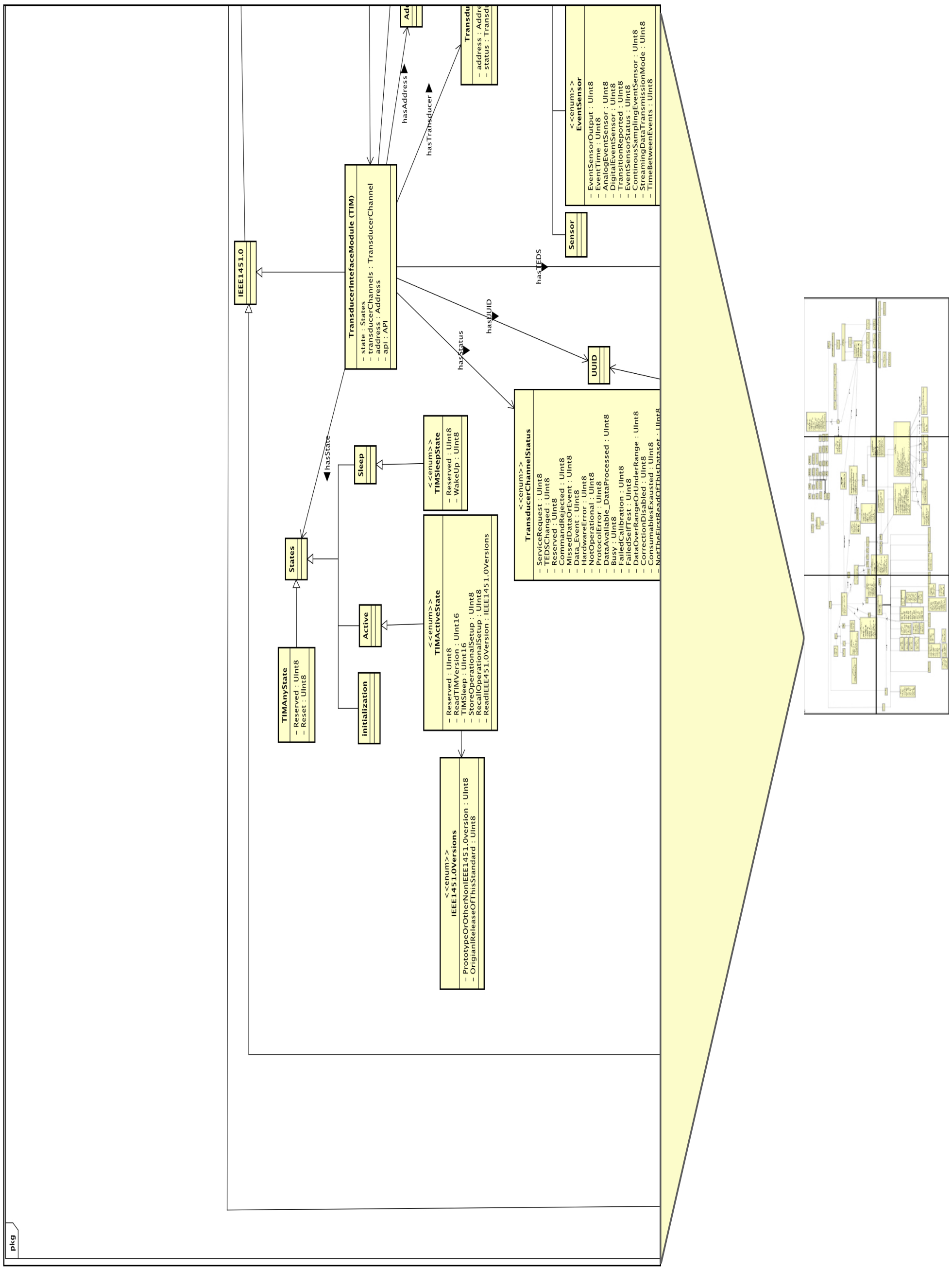


Figure A.1 Knowledge model TIM internal.

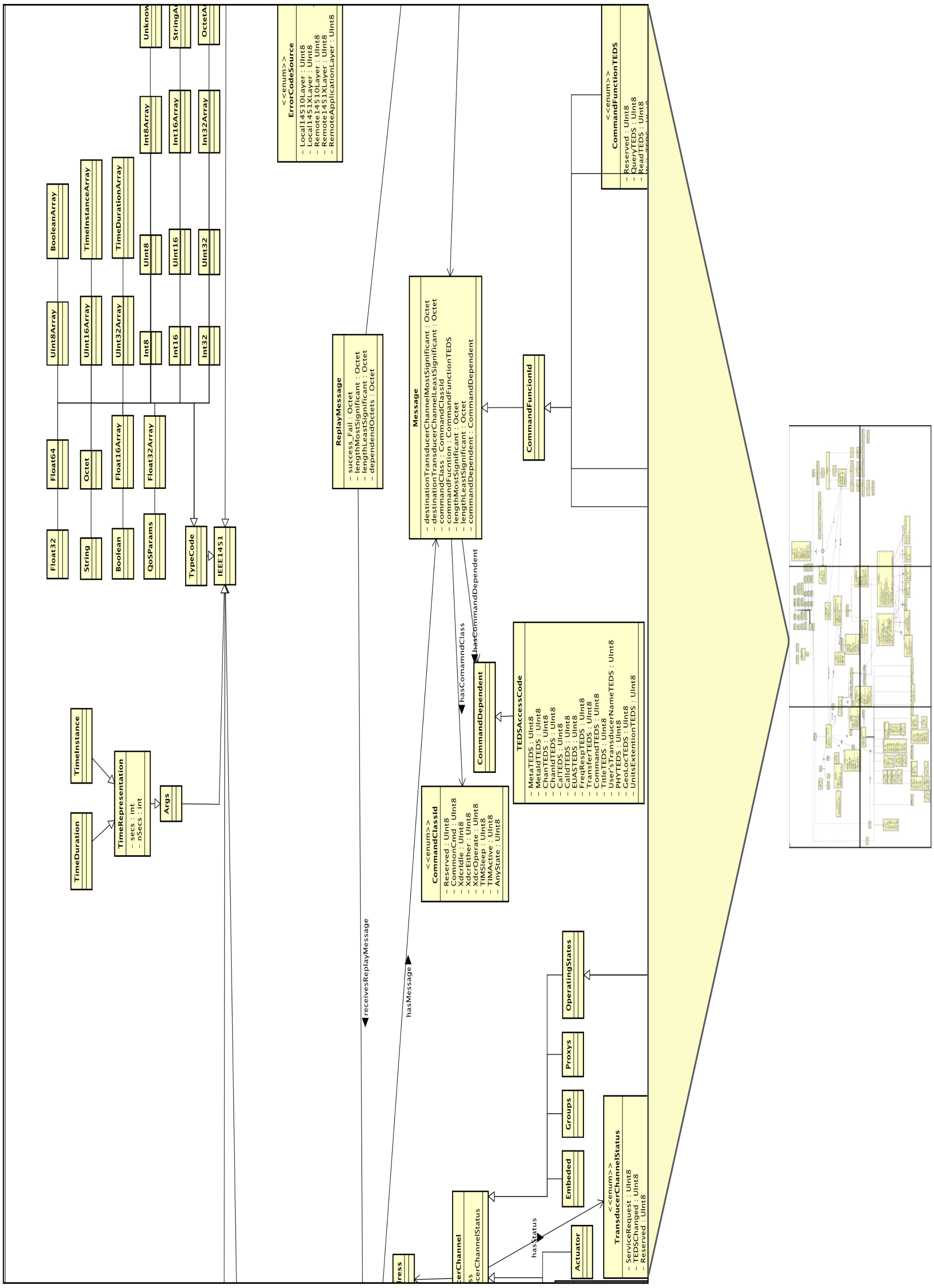


Figure A.2 Knowledge model data type and commands.

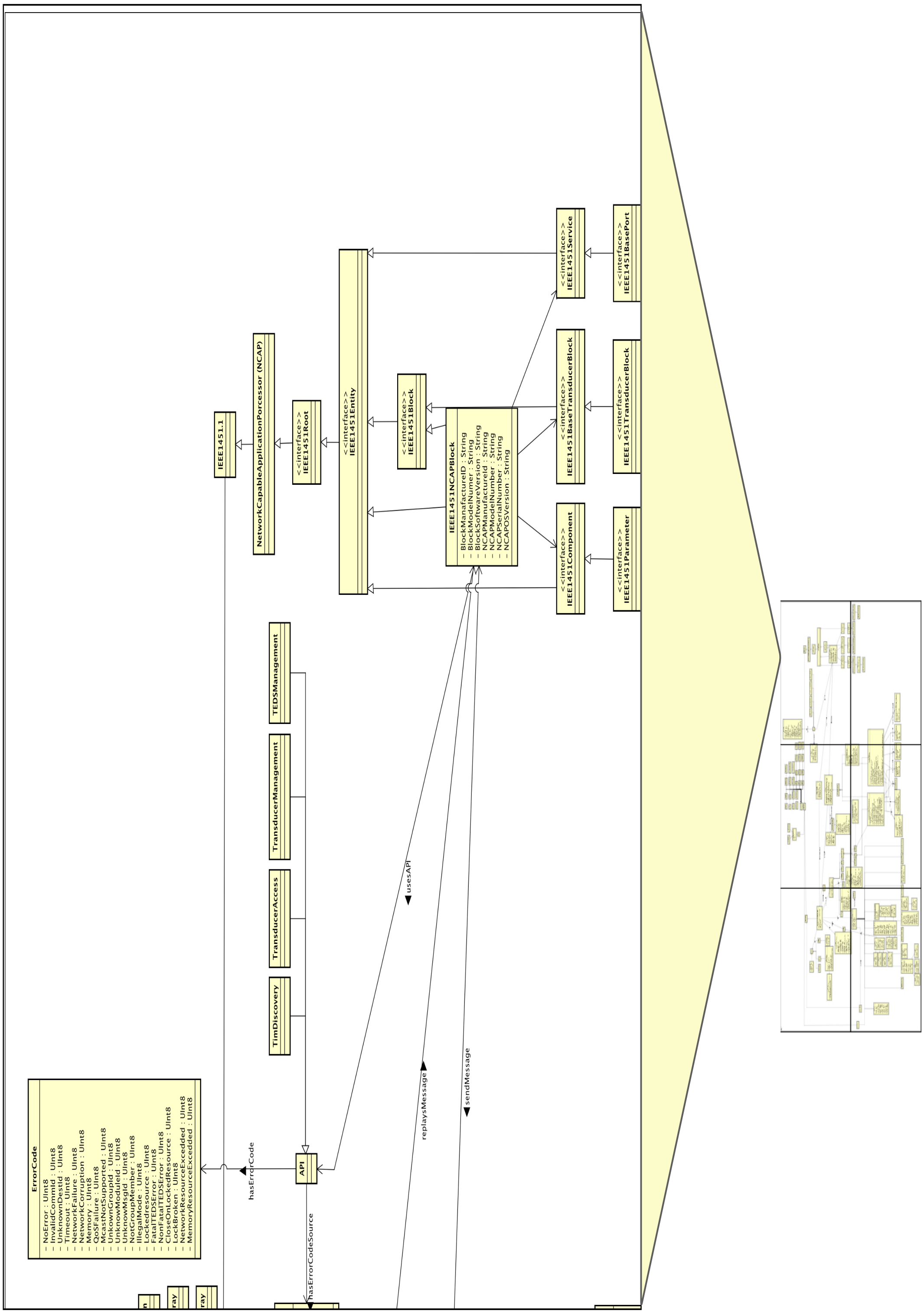


Figure A.3 Knowledge model API and NCAP.

Figure A.4 Knowledge model TEDS.

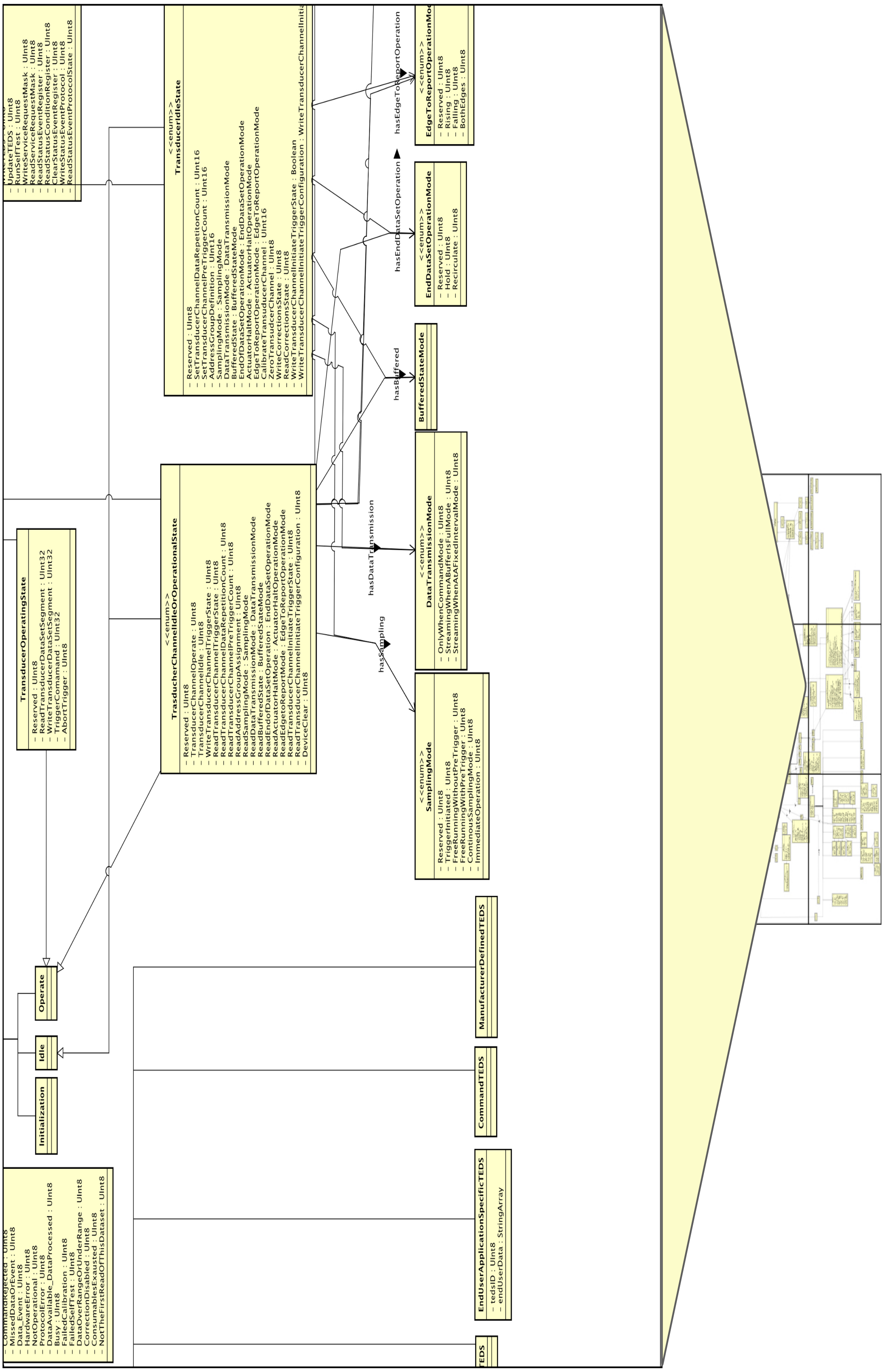


Figure A.5 Knowledge model TransducerChannel commands.

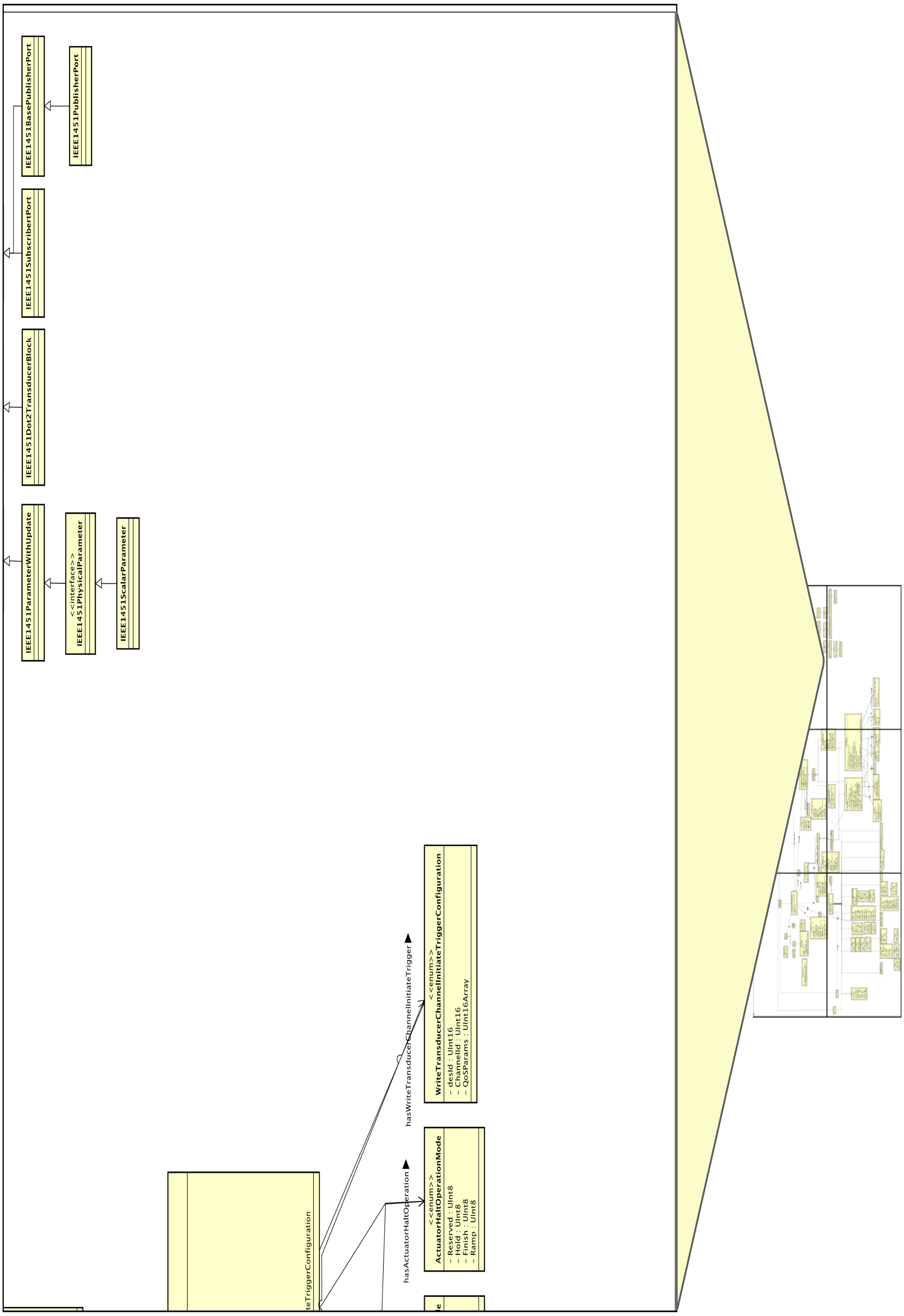


Figure A.6 Knowledge model NCAP classes.

Appendix B

Semantic IEEE 1451 Read TEDS Example

The current version of the standard for communication by octet array on the water level monitoring scenario.

[{
"error_code" : "0",
"timId" : "0x00 0x01",
"channelId" : "0x00 0x01",
"tedsType" : "0x01",
"teds" : "0x00 0x00 0x00 0x2C 0x03 0x04 0x00 0x00 0x00 0x01 0x04 0x0A 0x91 0xB3 0x50 0x1A 0x69 0x05 0xF9 0x82 0xDA 0x79 0x0A 0x04 0x3F 0x80 0x00 0x00 0x0B 0x04 0x40 0x40 0x00 0x00 0x0C 0x04 0x3F 0x80 0x00 0x00 0x0D 0x02 0x00 0x04 0x0E 0x09 0x15 0x01 0x01 0x16 0x04 0x01 0x02 0x03 0x04 0xF8 0x0"
},
{
"error_code" : "0",
"timId" : "0x00 0x01",
"channelId" : "0x00 0x01",
"tedsType" : "0x03",
"teds" : [
"0x00 0x00 0x00 0x62 0x03 0x04 0x00 0x00 0x00 0x01 0x0A 0x01 0x0 0x0B 0x01 0x0 0x0C 0x21 0x0 0x32 0x01 0x1 0x33 0x01 0x80 0x34 0x01 0x80 0x35 0x01 0x80 0x36 0x01 0x80 0x37 0x01 0x80 0x38 0x01 0x80 0x39 0x01 0x80 0x3A 0x01 0x80 0x3B 0x01 0x80 0x3C 0x01 0x80 0x0D 0x04 0x43 0x69 0x26 0x66 0x0E 0x04 0x43 0xC7 0x13 0x33 0x0F 0x04 00 0x00 0x00 0x00 0x10 0x01 00 0x11 0x01 00 0x12 0x28 0x01 02 0x29 0x01 00 0x2A 0x01 00 0x13 0x2B 0x01 0x00 0x2C 0x01 0x00 0x2D 0x01 0x00 0x2E 0x01 0x00 0x2F 0x01 0x00 0x14 0x04 0x00 0x00 0x00 0x00 0x15 0x04 0x00 0x00 0x00 0x00 0x16 0x04 0x00 0x00 0x00 0x00 0x17 0x04 0x00 0x00 0x00 0x00 0x18 0x04 0x00 0x00 0x00 0x00 0x19 0x04 0x00 0x00 0x00 0x00 0x1A 0x04 0x00 0x00 0x00 0x00 0x1B 0x01 0x00 0x1C 0x04 0x00 0x00 0x00 0x00 0x1D 0x04 0x00 0x00 0x00 0x00 0x1E 0x04 0x00 0x00 0x00 0x00 0x1F 0x30 0x01 0x00 0x31 0x01 0x01 0x20 0x01 0x00 0x21 0x01 0x00 0x22 0x01 0x00 0x23 0x01 0x00 0x24 0x01 0x00 0x25 0x04 0x00 0x00 0x00 0x00 0x26 0x08 0x00 0x00 0x00 0x00 0x00 0x00 0x00 00 0x27 0x01 0x01 0xFF 0x7E",
"0x00 0x00 0x00 0x62 0x03 0x04 0x00 0x00 0x00 0x01 0x0A 0x01 0x0 0x0B 0x01 0x0 0x0C 0x21 0x0 0x32 0x01 0x0 0x33 0x01 0x80 0x34 0x01 0x80 0x35 0x01 0x80 0x36 0x01 0x80 0x37 0x01 0x80 0x38 0x01 0x80 0x39 0x01 0x80 0x3A 0x01 0x80 0x3B 0x01 0x80 0x3C 0x01 0x80 0x0D 0x04 0x00 0x00 0x00 0x00 0x0E 0x04 0x42 0xC8 0x00 0x00 0x0F 0x04 00 0x00 0x00 0x00 0x10 0x01 00 0x11 0x01 00 0x12 0x28 0x01 02 0x29 0x01 00 0x2A 0x01 00 0x13 0x2B 0x01 0x00 0x2C 0x01 0x00 0x2D 0x01 0x00 0x2E 0x01 0x00 0x2F 0x01 0x00 0x14 0x04 0x00 0x00 0x00 0x00 0x15 0x04 0x00 0x00 0x00 0x00 0x16 0x04 0x00 0x00 0x00 0x00 0x17 0x04 0x00 0x00 0x00 0x00 0x18 0x04 0x00 0x00 0x00 0x00 0x19 0x04 0x00 0x00 0x00 0x00 0x1A 0x04 0x00 0x00 0x00 0x00 0x1B 0x01 0x00 0x1C 0x04 0x00 0x00 0x00 0x00 0x1D 0x04 0x00 0x00 0x00 0x00 0x1E 0x04 0x00 0x00 0x00 0x00 0x1F 0x30 0x01 0x00 0x31 0x01 0x01 0x20 0x01 0x00 0x21 0x01 0x00 0x22 0x01 0x00 0x23 0x01 0x00 0x24 0x01 0x00 0x25 0x04 0x00 0x00 0x00 0x00 0x26 0x08 0x00 0x00 0x00 0x00 0x00 0x00 0x00 00 0x27 0x01 0x01 0xFF 0x7E",
"0x00 0x00 0x00 0x62 0x03 0x04 0x00 0x00 0x00 0x01 0x0A 0x01 0x0 0x0B 0x01 0x2 0x0C 0x21 0x0 0x32 0x01 0x1 0x33 0x01 0x80 0x34 0x01 0x80 0x35 0x01 0x80 0x36 0x01 0x80 0x37 0x01 0x80 0x38 0x01 0x80 0x39 0x01 0x80 0x3A 0x01 0x80 0x3B 0x01 0x80 0x3C 0x01 0x80 0x0D 0x04 0x00 0x00 0x00 0x00 0x0E 0x04 0x00 0x00 0x00 0x00 0x0F 0x04 00 0x00 0x00 0x00 0x10 0x01 00 0x11 0x01 00 0x12 0x28 0x01 00 0x29 0x01 00 0x2A 0x01 00 0x13 0x2B 0x01 0x00 0x2C 0x01 0x00 0x2D 0x01 0x00 0x2E 0x01 0x00 0x2F 0x01 0x00 0x14 0x04 0x00 0x00 0x00 0x00 0x15 0x04 0x00 0x00 0x00 0x00 0x16 0x04 0x00 0x00 0x00 0x00 0x17 0x04 0x00 0x00 0x00 0x00 0x18 0x04 0x00 0x00 0x00 0x00 0x19 0x04 0x00 0x00 0x00 0x00 0x1A 0x04 0x00 0x00 0x00 0x00 0x1B 0x01 0x00 0x1C 0x04 0x00 0x00 0x00 0x00 0x1D 0x04 0x00 0x00 0x00 0x00 0x1E 0x04 0x00 0x00 0x00 0x00 0x1F 0x30 0x01 0x00 0x31 0x01 0x01 0x20 0x01 0x00 0x21 0x01 0x00 0x22 0x01 0x00 0x23 0x01 0x00 0x24 0x01 0x00 0x25 0x04 0x00 0x00 0x00 0x00 0x26 0x08 0x00 0x00 0x00 0x00 0x00 0x00 0x00 00 0x27 0x01 0x01 0xFF 0x7E",
"0x00 0x00 0x00 0x62 0x03 0x04 0x00 0x00 0x00 0x01 0x0A 0x01 0x0 0x0B 0x01 0x2 0x0C 0x21 0x0 0x32 0x01 0x1 0x33 0x01 0x80 0x34 0x01 0x80 0x35 0x01 0x80 0x36 0x01 0x80 0x37 0x01 0x80 0x38 0x01 0x80 0x39 0x01 0x80 0x3A 0x01 0x80 0x3B 0x01 0x80 0x3C 0x01 0x80 0x0D 0x04 0x00 0x00 0x00 0x00 0x0E 0x04 0x42 0xC8 0x00 0x00 0x0F 0x04 00 0x00 0x00 0x00 0x10 0x01 00 0x11 0x01 00 0x12 0x28 0x01 00 0x29 0x01 00 0x2A 0x01 00 0x13 0x2B 0x01 0x00 0x2C 0x01 0x00 0x2D 0x01 0x00 0x2E 0x01 0x00 0x2F 0x01 0x00 0x14 0x04 0x00 0x00 0x00 0x00 0x15 0x04 0x00 0x00 0x00 0x00 0x16 0x04 0x00 0x00 0x00 0x00 0x17 0x04 0x00 0x00 0x00 0x00 0x18 0x04 0x00 0x00 0x00 0x00 0x19 0x04 0x00 0x00 0x00 0x00 0x1A 0x04 0x00 0x00 0x00 0x00 0x1B 0x01 0x00 0x1C 0x04 0x00 0x00 0x00 0x00 0x1D 0x04 0x00 0x00 0x00 0x00 0x1E 0x04 0x00 0x00 0x00 0x00 0x1F 0x30 0x01 0x00 0x31 0x01 0x01 0x20 0x01 0x00 0x21 0x01 0x00 0x22 0x01 0x00 0x23 0x01 0x00 0x24 0x01 0x00 0x25 0x04 0x00 0x00 0x00 0x00 0x26 0x08 0x00 0x00 0x00 0x00 0x00 0x00 0x00 00 0x27 0x01 0x01 0xFF 0x7E"
]
}, {
"error_code" : "0",
"timId" : "0x00 0x01",
"channelId" : "0x00 0x01",
"tedsType" : "0x0c",

"teds" : "0x00 0x00 0x00 0x1C 0x03 0x04 0x00 0x0C 0x01 0x01 0x04 0x01 0x00 0x05 0x11 0x57 0x61 0x74 0x65 0x72 0x20 0x4C 0x65 0x76 0x65 0x6C 0x 0xFB 0x9E"
}, {
"error_code" : "0",
"timId" : "0x00 0x01",
"channelId" : "0x00 0x01",
"tedsType" : "0x0d",
"teds" : "0x00 0x00 0x00 0x46 0x03 0x04 0x00 0x00 0x00 0x0C 0x0A 0x01 0x04 0x0B 0x04 0x00 0x00 0x00 0x01 0x0C 0x02 0x00 0x01 0x0D 0x02 0x00 0x01 0x0E 0x02 0x00 0x00 0x0F 0x01 0x00 0x10 0x02 0x00 0x00 0x11 0x02 0x00 0x01 0x12 0x02 0x00 0x0A 0x13 0x04 0x00 0x00 0x00 0x01 0x14 0x04 0x00 0x00 0x00 0x01 0x15 0x01 0x01 0x16 0x01 0x00 0x17 0x02 0x00 0x00 0x18 0x02 0x00 0x02 0xFE 0x78"
}]

The request for data for communication using the semantic context from the IEEE 1451 semantic layer encoder.

{
"@context": "http://iml.ubi.pt/2022/ieee1451/ieee1451.jsonld",
"NCAP": {
"BlockManufactureID": "0",
"BlockModelNumber": "UBIBlock_1",
"BlockSoftwareVersion": "sv001",
"NCAPManufactureID": "IML2022",
"NCAPModelNumber": "0.2",
"NCAPSerialNumber": "5",
"NCAOOSVersion": "0.5",
"TIM":[{
"timId": "1",
"MetaTEDS": {
"TEDSLength": "59",
"tedsID ": "1",
"UUID": "91B3501A6905F982DA79",
"oHoldOff": "1",
"sHoldOff": "1",
"testTime": "1",
"maxChan": "4",
"cGroup": {
"grpType": "1",
"memList": "1"
}
},
"TransducerChannelTEDS": [{
"TEDSLength": "99",
"tedsID": "3",
"calKey": "0",
"chanType": "0",
"phyUnits": {
"unitType": "1",
"radians": "128",
"sterRad": "128",
"meters": "128",
"kilogram": "128",
"seconds": "128",
"amperes": "128",
"kelvins": "129",
"moles": "128",
"candelas": "128",
"unitsExt": "3"
},
"lowLimit": "233.15",
"hiLimit": "398.15",
"oError": "1",
"selfTest": "0",
"mRange": "0",
"sample": {
"datModel": "0",
"modLenth": "0",
"sigBits": "0"
},
"updateT": "2",
"rSetupT": "1",
"sPeriod": "10",
"warmUpT": "1",
"rDelayT": "1",
"testTime": "2",

"sampling": {
"sampMode": "0"
}
},{
"TEDSLength": "99",
"tedsID": "3",
"calKey": "0",
"chanType": "0",
"phyUnits": {
"unitType": "1",
"radians": "128",
"sterRad": "128",
"meters": "128",
"kilogram": "128",
"seconds": "128",
"amperes": "128",
"kelvins": "128",
"moles": "128",
"candelas": "128",
"unitsExt": "3"
},
"lowLimit": "0",
"hiLimit": "100",
"oError": "0",
"selfTest": "0",
"mRange": "0",
"sample": {
"datModel": "0",
"modLenth": "0",
"sigBits": "0"
},
"updateT": "2",
"rSetupT": "1",
"sPeriod": "10",
"warmUpT": "1",
"rDelayT": "1",
"testTime": "2",
"sampling": {
"sampMode": "0"
}
},{
"TEDSLength": "99",
"tedsID": "3",
"calKey": "0",
"chanType": "2",
"phyUnits": {
"unitType": "1",
"radians": "128",
"sterRad": "128",
"meters": "128",
"kilogram": "128",
"seconds": "128",
"amperes": "128",
"kelvins": "128",
"moles": "128",
"candelas": "128",
"unitsExt": "3"
},
"lowLimit": "0",
"hiLimit": "100",
"oError": "1",
"selfTest": "0",
"mRange": "0",
"sample": {
"datModel": "0",
"modLenth": "0",
"sigBits": "0"
},
"updateT": "2",
"rSetupT": "1",
"sPeriod": "10",
"warmUpT": "1",
"rDelayT": "1",
"testTime": "2",
"sampling": {
"sampMode": "0"
}

},{
"TEDSLength": "99",
"tedsID": "3",
"calKey": "0",
"chanType": "2",
"phyUnits": {
"unitType": "1",
"radians": "128",
"sterRad": "128",
"meters": "128",
"kilogram": "128",
"seconds": "128",
"amperes": "128",
"kelvins": "128",
"moles": "128",
"candelas": "128",
"unitsExt": "3"
},
"lowLimit": "0",
"hiLimit": "100",
"oError": "1",
"selfTest": "0",
"mRange": "0",
"sample": {
"datModel": "0",
"modLenth": "0",
"sigBits": "0"
},
"updateT": "2",
"rSetupT": "1",
"sPeriod": "2",
"warmUpT": "1",
"rDelayT": "1",
"testTime": "2",
"sampling": {
"sampMode": "0"
}
}],
"UsersTransducerNameTEDS": {
"TEDSLength": "29",
"tedsID": "12",
"format": "0",
"tCName": "Water Level"
},
"TEDSLength": "20",
"tedsID": "13",
"radioType": "4",
"maxBPS": "1",
"maxCDev" : "10",
"maxRDev" : "1",
"encryption": "0",
"authent" : "0",
"minKeyL" : "0",
"maxKeyL" : "1",
"maxSDU" : "10",
"minALat" : "1",
"maxTLat" : "1",
"maxXAct" : "1",
"battery" : "0",
"radio" : "0",
"maxRetry" : "2"
}]
}
}

The request for data for communication using the semantic context from the IEEE 1451 semantic layer decoder.

{
"http://www.iml.ubi.pt/2022/ieee1451#NCAP": {
"http://www.iml.ubi.pt/2022/ieee1451#BlockManufactureID": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#String",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#BlockModelNumber": {

"@type": "http://www.iml.ubi.pt/2022/ieee1451#String",
"@value": "UBIBlock_1"
},
"http://www.iml.ubi.pt/2022/ieee1451#BlockSoftwareVersion": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#String",
"@value": "sv001"
},
"http://www.iml.ubi.pt/2022/ieee1451#NCAPManufactureID": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#String",
"@value": "IML2022"
},
"http://www.iml.ubi.pt/2022/ieee1451#NCAPModelNumber": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#String",
"@value": "0.2"
},
"http://www.iml.ubi.pt/2022/ieee1451#NCAPSerialNumber": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#String",
"@value": "5"
},
"http://www.iml.ubi.pt/2022/ieee1451#TIM": {
"http://www.iml.ubi.pt/2022/ieee1451#MetaTEDS": {
"http://www.iml.ubi.pt/2022/ieee1451#TEDSLength": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt32",
"@value": "59"
},
"http://www.iml.ubi.pt/2022/ieee1451#UUID": "91B3501A6905F982DA79",
"http://www.iml.ubi.pt/2022/ieee1451#cGroup": {
"http://www.iml.ubi.pt/2022/ieee1451#grpType": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#memList": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt16Array",
"@value": "1"
}
},
"http://www.iml.ubi.pt/2022/ieee1451#maxChan": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt16",
"@value": "4"
},
"http://www.iml.ubi.pt/2022/ieee1451#oHoldOff": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#sHoldOff": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#testTime": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "1"
}
},
"http://www.iml.ubi.pt/2022/ieee1451#TEDSLength": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt32",
"@value": "20"
},
"http://www.iml.ubi.pt/2022/ieee1451#TransducerChannelTEDS": [
{
"http://www.iml.ubi.pt/2022/ieee1451#TEDSLength": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt32",
"@value": "99"
},
"http://www.iml.ubi.pt/2022/ieee1451#calKey": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#chanType": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#hiLimit": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "398.15"
},
"http://www.iml.ubi.pt/2022/ieee1451#lowLimit": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",

"@value": "233.15"
},
"http://www.iml.ubi.pt/2022/ieee1451#mRange": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#oError": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#phyUnits": {
"http://www.iml.ubi.pt/2022/ieee1451#amperes": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#candelas": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#kelvins": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "129"
},
"http://www.iml.ubi.pt/2022/ieee1451#kilogram": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#meters": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#moles": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#radians": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#seconds": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#sterRad": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#unitType": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#unitsExt": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "3"
}
},
"http://www.iml.ubi.pt/2022/ieee1451#rDelayT": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#rSetupT": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#sPeriod": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "10"
},
"http://www.iml.ubi.pt/2022/ieee1451#sample": {
"http://www.iml.ubi.pt/2022/ieee1451#datModel": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#modLenth": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},

"http://www.iml.ubi.pt/2022/ieee1451#sigBits": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt16",
"@value": "0"
}
},
"http://www.iml.ubi.pt/2022/ieee1451#sampling": {
"http://www.iml.ubi.pt/2022/ieee1451#sampMode": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
}
},
"http://www.iml.ubi.pt/2022/ieee1451#selfTest": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#tedsID": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "3"
},
"http://www.iml.ubi.pt/2022/ieee1451#testTime": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "2"
},
"http://www.iml.ubi.pt/2022/ieee1451#updateT": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "2"
},
"http://www.iml.ubi.pt/2022/ieee1451#warmUpT": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "1"
}
},
{
"http://www.iml.ubi.pt/2022/ieee1451#TEDSLength": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt32",
"@value": "99"
},
"http://www.iml.ubi.pt/2022/ieee1451#calKey": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#chanType": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#hiLimit": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "100"
},
"http://www.iml.ubi.pt/2022/ieee1451#lowLimit": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#mRange": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#oError": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#phyUnits": {
"http://www.iml.ubi.pt/2022/ieee1451#amperes": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#candelas": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#kelvins": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#kilogram": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"

},
"http://www.iml.ubi.pt/2022/ieee1451#meters": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#moles": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#radians": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#seconds": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#sterRad": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#unitType": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#unitsExt": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "3"
}
},
"http://www.iml.ubi.pt/2022/ieee1451#rDelayT": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#rSetupT": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#sPeriod": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "10"
},
"http://www.iml.ubi.pt/2022/ieee1451#sample": {
"http://www.iml.ubi.pt/2022/ieee1451#datModel": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#modLenth": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#sigBits": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt16",
"@value": "0"
}
},
"http://www.iml.ubi.pt/2022/ieee1451#sampling": {
"http://www.iml.ubi.pt/2022/ieee1451#sampMode": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
}
},
"http://www.iml.ubi.pt/2022/ieee1451#selfTest": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#tedsID": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "3"
},
"http://www.iml.ubi.pt/2022/ieee1451#testTime": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "2"
},
"http://www.iml.ubi.pt/2022/ieee1451#updateT": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "2"

},
"http://www.iml.ubi.pt/2022/ieee1451#warmUpT": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "1"
}
},
{
"http://www.iml.ubi.pt/2022/ieee1451#TEDSLength": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt32",
"@value": "99"
},
"http://www.iml.ubi.pt/2022/ieee1451#calKey": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#chanType": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "2"
},
"http://www.iml.ubi.pt/2022/ieee1451#hiLimit": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "100"
},
"http://www.iml.ubi.pt/2022/ieee1451#lowLimit": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#mRange": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#oError": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#phyUnits": {
"http://www.iml.ubi.pt/2022/ieee1451#amperes": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#candelas": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#kelvins": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#kilogram": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#meters": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#moles": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#radians": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#seconds": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#sterRad": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#unitType": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#unitsExt": {

"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "3"
}
},
"http://www.iml.ubi.pt/2022/ieee1451#rDelayT": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#rSetupT": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#sPeriod": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "10"
},
"http://www.iml.ubi.pt/2022/ieee1451#sample": {
"http://www.iml.ubi.pt/2022/ieee1451#datModel": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#modLenth": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#sigBits": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt16",
"@value": "0"
}
},
"http://www.iml.ubi.pt/2022/ieee1451#sampling": {
"http://www.iml.ubi.pt/2022/ieee1451#sampMode": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
}
},
"http://www.iml.ubi.pt/2022/ieee1451#selfTest": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#tedsID": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "3"
},
"http://www.iml.ubi.pt/2022/ieee1451#testTime": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "2"
},
"http://www.iml.ubi.pt/2022/ieee1451#updateT": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "2"
},
"http://www.iml.ubi.pt/2022/ieee1451#warmUpT": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "1"
}
},
{
"http://www.iml.ubi.pt/2022/ieee1451#TEDSLength": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt32",
"@value": "99"
},
"http://www.iml.ubi.pt/2022/ieee1451#calKey": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#chanType": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "2"
},
"http://www.iml.ubi.pt/2022/ieee1451#hiLimit": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "100"
},
"http://www.iml.ubi.pt/2022/ieee1451#lowLimit": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "0"

},
"http://www.iml.ubi.pt/2022/ieee1451#mRange": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#oError": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#phyUnits": {
"http://www.iml.ubi.pt/2022/ieee1451#amperes": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#candelas": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#kelvins": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#kilogram": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#meters": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#moles": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#radians": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#seconds": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#sterRad": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "128"
},
"http://www.iml.ubi.pt/2022/ieee1451#unitType": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#unitsExt": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "3"
}
},
"http://www.iml.ubi.pt/2022/ieee1451#rDelayT": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#rSetupT": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#sPeriod": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "2"
},
"http://www.iml.ubi.pt/2022/ieee1451#sample": {
"http://www.iml.ubi.pt/2022/ieee1451#datModel": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#modLenth": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#sigBits": {

"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt16",
"@value": "0"
}
},
"http://www.iml.ubi.pt/2022/ieee1451#sampling": {
"http://www.iml.ubi.pt/2022/ieee1451#sampMode": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
}
},
"http://www.iml.ubi.pt/2022/ieee1451#selfTest": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#tedsID": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "3"
},
"http://www.iml.ubi.pt/2022/ieee1451#testTime": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "2"
},
"http://www.iml.ubi.pt/2022/ieee1451#updateT": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "2"
},
"http://www.iml.ubi.pt/2022/ieee1451#warmUpT": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Float32",
"@value": "1"
}
}
],
"http://www.iml.ubi.pt/2022/ieee1451#UsersTransducerNameTEDS": {
"http://www.iml.ubi.pt/2022/ieee1451#TEDSLength": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt32",
"@value": "29"
},
"http://www.iml.ubi.pt/2022/ieee1451#format": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#tCName": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#String",
"@value": "Water Level"
},
"http://www.iml.ubi.pt/2022/ieee1451#tedsID": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "12"
}
},
"http://www.iml.ubi.pt/2022/ieee1451#authent": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#Boolean",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#battery": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#maxBPS": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt32",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#maxCDev": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt16",
"@value": "10"
},
"http://www.iml.ubi.pt/2022/ieee1451#maxKeyL": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt16",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#maxRDev": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt16",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#maxRetry": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt16",
"@value": "2"

},
"http://www.iml.ubi.pt/2022/ieee1451#maxSDU": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt16",
"@value": "10"
},
"http://www.iml.ubi.pt/2022/ieee1451#minALat": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt32",
"@value": "1"
},
"http://www.iml.ubi.pt/2022/ieee1451#minKeyL": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt16",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#radio": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt32",
"@value": "0"
},
"http://www.iml.ubi.pt/2022/ieee1451#tedsID": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt8",
"@value": "13"
},
"http://www.iml.ubi.pt/2022/ieee1451#timId": {
"@type": "http://www.iml.ubi.pt/2022/ieee1451#UInt16",
"@value": "1"
}
}
}
}