An Ontology for Operator 4.0 based on Interoperability of Industrie 4.0 Reference Architectures with FIWARE

handed in
BACHELOR'S THESIS

Chieh Lin

born on the 18.12.1995
living in:
Riegerhofstr. 9
80686 Munich
Tel.: 0151 - 61259770

Human-centered Assistive Robotics
Technical University of Munich

Univ.-Prof. Dr.-Ing. Dongheui Lee

Supervisor:
Prof. Dr.-Ing. Dongheui Lee,
M. Eng. Matteo Pantano

Start:	03.11.2020
Intermediate Report:	26.01.2021
Delivery:	20.04.2021

TECHNISCHE UNIVERSITÄT MÜNCHEN
Human-centered Assistive Robotics
UNIV.-PROF. DR.-ING. DONGHEUI LEE

April 19, 2021

B A C H E L O R T H E S I S

Interoperability of Industrie 4.0 reference architectures for augmented operators

Problem description:

Robotics and human robot collaboration is largely diffused in big conglomerates and automotive companies. However, European Small Medium Enterprises (SMEs), which constitute the major part of the economy, still are lagging on their adoption [1]. Considering Human Robot Collaboration, its application is hindered by three barriers: safety, design and interfaces [2]. Therefore, methodologies for addressing these barriers and closing the gap are necessary.

In the setting up of interfaces several effort is currently spent by system integrators for ensuring data consistency and compliance. However, when data generated by operators is considered such operation is not trivial and frequently precious information is lost. On one hand, several approaches using reference architectures (e.g. RAMI4.0 [3], IDS [4]) for maintaining data integrity are present when digital tools (e.g. PLM) are used. On the other hand, their combination with human information has not yet been investigated. On this account a research effort is necessary. During this project you will have a chance to evaluate the state of the art regarding the topic and propose methodologies for human data integration in digital tools. Moreover, testing and implementation of the proposed solution in a factory alike scenario will complete the research. This will help SMEs to embrace digital tools.

Tasks:

- Literature research
- Choice of a reference architecture for integration of human components
- Proposal of a framework for human data integration
- Implementation of the proposed framework on a factory alike infrastructure

Bibliography:

[1] Eurostat, "25% of large enterprises in the EU use robots", European Commission, 2019.
[2] V. Villani, et al., "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications", Mechatronics, Volume 55, 2018, pp. 248-266, ISSN 0957-4158.
[3] *Referenzarchitekturmodell industrie 4.0 (RAMI4.0)*, DIN SPEC 91345:2016-04, Deutsches Institut für Normung, Berlin, DE, 2016.
[4] Prof. Dr.-Ing. Boris Otto, et al., "White paper Industrial Data Space", Fraunhofer-Gesellschaft zur Förderung, 2018, www.industrialdataspace.org.

Supervisor:	Prof. Dr. Dongheui Lee, M. Eng. Matteo Pantano
Start:	03/11/2020
Intermediate Report:	26/01/2021
Delivery:	20/04/2021

(D. Lee)
Univ.-Professor

Abstract

As a result of advanced technologies such as cyber–physical systems, big data, and machine learning, the evolution of smart manufacturing has become a focus point. Various improvements have already been done in the past decades, including new-developed smart manufacturing reference architectures and the enhancement of system performances and decision-making. Nevertheless, data plays an important role within these improvements, especially data that is brought into existence by these advance technologies. How can the communication between humans and machines become more efficient and seamless?

In this thesis, various existing smart manufacturing reference architectures will be reviewed and analyzed. Moreover, paradigms and methods related to smart manufacturing will be introduced. On this basis, a suitable reference architecture will be chosen and implemented into a use case with human data. The use case will be implemented on an open-source platform to make the methodology accessible for further research regarding the integration of human data in smart manufacturing.

Contents

1. Introduction

The manufacturing industry has always been driven by technology's evolution. From steam engines to electricity, microprocessors, computers, robotics, and lately, artificial intelligence, the Internet of Things, and cyber-physical systems. The increasing need for interoperability at various manufacturing systems is one of the associated obstacles of this evolution. Consequently, efficient implementation of smart manufacturing interoperability will result in efficient communication and decreased error rates throughout data exchange between devices, sensors, customers, networks, and platforms. Smart manufacturing aims to implement cloud-based services, enterprises, smart factories, smart machines, and operations that are complex and distributed. In turn, the required integration involves a smooth exchange of data between these structures. When introducing implementations, several efforts are focused on maintaining accuracy and conformity in the data. There are also several techniques available that preserve data completeness using reference architectures, such as the Industry 4.0 Reference Architectural Model (RAMI4.0) [1], International Data Spaces (IDS) [2], Industrial Internet Reference Architecture (IIRA) [3] Their integration with human data, nevertheless, has not yet been explored.

1.1. Thesis Structure

This thesis conducts a review of the notion of interoperability of Industrie 4.0 reference architectures in the context of smart manufacturing. In the first chapter, a brief introduction of the thesis and the thesis structure will be provided.

Moving on to the second chapter, which contains the state of the art. This chapter starts with an overview of the modern concepts and paradigms in smart manufacturing, along with their detailed characteristics and elements, comparative analysis, and examples. An overview of the Operator 4.0 typology will be introduced, along with examples and how they assist human operators throughout operations—furthermore, a more in-depth observation into interoperability in manufacturing and its challenges will be provided. Existing reference architectures will be analyzed and explained—moreover, an overview of an open-source platform called FIWARE, where the chosen reference architecture will be implemented, will be introduced. Smart data models will also be examined, with a few existing smart data models as examples.

Next, an overview of six reference architectures with their human-engagement will be introduced in chapter three. Model mapping from all reference architectures will be provided with an explanation of where humans data is integrated into the models. Finally, reference architectures will be compared with one another. Therefore a decision will be reached which reference architecture to implement with the human data to observe the data's accuracy and complicity throughout every layer. A vital part of this chapter is investigating existing data models. Specifically, data models appropriate to present human data. Moreover, in order to implement a suitable ontology, data modeling will be introduced.

In the fourth chapter, which contains the implementation. A use case will be introduced, along with the software and hardware that are operated for the realization of the use case. The architecture of the system will be drafted out with explanation, showing how the use case is implemented.

Lastly, a conclusion of the thesis will be overviewed in the last chapter. The future plan of this thesis will moreover be discussed.

2. State of the Art

This chapter gives an overview to the topics related to the thesis and their correlation to the research. That includes Operator 4.0 typology, reference architectures, and FIWARE, an open-source platform that would be used for the implementation. These essential techniques and proposals would provide a clearer view of the thesis topic.

2.1. Cyber-Physical Systems (CPS)

Cyber-Physical Systems (CPS) are systems in which technical information and software components are connected with mechanical or electronic components. With the help of sensors and networks built into physical objects, physical components can be linked to the Internet[4]. The Internet can be seen as a tool for transforming communication between humans and Information. Therefore, CPS can be considered as a tool for assisting communication between human and engineered systems. Data exchanged, and infrastructures are efficiently monitored and controlled, resulting in Cyber-Physical Systems accommodating humans in all kinds of fields and serving an enormous spectrum of possible functions[5]. For instance, in health care and medicine, bionic limbs and robotic surgery can allow patients to heal and regain mobility. Environmentally, CPS can help firefighters identify and prevent fires or support scientists manage underwater oil spills[6]. The areas of application often coincide with those of Industry 4.0. The systems aim to ensure greater efficiency, lower costs, and faster processing of complex operations.

2.2. Internet of Things (IoT)

Internet of Things (IoT) encompasses anything connected to the Internet, but it is increasingly being used to define objects that "communicate" with each other. IoT enables devices with Internet connections to communicate with others, and the Internet of Things brings these networks together. It gives devices the ability to interact across different networks, creating a much more connected world[7]. Almost all things can be connected to the Internet today. Ericsson predicted in 2018 that by 2023, approximately 31.4 billion devices will be connected to the Internet, including 19.8 billion IoT objects and 11.6 billion standard devices (Computers, smartphones, and landline phones)[8]. These networked devices will then generate a large amount of structured data that can be used for analysis and optimization purposes. IoT is thus one of the largest suppliers of Big Data. All fields are affected by the Internet of Things, from transportation to health, retail, and energy. In manufacturing, networked objects can improve logistics, maintenance, or production planning, as this system makes the performance and monitoring of equipment visible in real-time. An example of IoT is the Connected Factory along with Azure IoT; it is widely used by companies to manage industrial IoT devices[9]. With the help of connection from numerous resources via cloud software, many devices can be controlled, and connected factory solutions can register data from devices.

2.3. Data Analytics and Machine Learning

Massive data are generated from the integration of CPS and IoT, data coming from sensors built into physical objects, devices managed by IoT devices, and real-time feedback prompted by smart products. With the rising amount of data and diverse datasets, manufacturing companies demand smart solutions to create efficient and flexible production processes. Manufacturers may gain more insight by looking into past data set with data analytics, defining similarities and relationships between different procedures and sources, and then modify the factors that seem to have the most significant impact on development. By using forecasted outcomes, similar patterns, and quality control, production processes can be strongly enhanced. Big data has been used by big companies such as BMW™to detect error patterns and run predictive machine learning algorithms for quality control[10]. The rapid growth of machine learning algorithms and enhanced computers has generated new possibilities for more insights from their existing datasets to support manufacturing management and decision. Data analytics and machine learning have been commonly used in the manufacturing world in numerous fields, such as quality control, performance monitoring, failure detection, decision-making, and scheduling[11].

2.4. Paradigms in Manufacturing

Several new paradigms have emerged in the last decade to reflect the different demands and features of next-generation manufacturing systems. All these paradigms have a common factor. It focuses on service-orientated productions with the help of digital transformation and cloud integration of resources, along with smart reconfigurable shop-floor operations via automated tools and systems. Some common paradigms involving manufacturing are listed below.

2.4.1. Smart Manufacturing

Smart Manufacturing is a specific application of the Industrial Internet of Things (IIoT); it helps transform the IoT into reality. The implementation of smart manufacturing involves integrating sensors into manufacturing machines to collect data from their operation status and performance. The Information has usually been maintained on individual computers in local databases in the past. As for today, manufacturing engineers and data experts can more easily determine the causes of machine failures by evaluating the data stream from an entire factory's or even several facilities' equipment[12]. Smart manufacturing also predicts necessary maintenance and repairs, therefore avoiding unplanned downtime on machines. For example, manufacturing companies can use the data to evaluate patterns, thus recognizing process steps where production is inefficient because of certain materials. Furthermore, researchers can use the data to model production processes on the computers and improve efficiency[13].

2.4.2. Cyber-Physical Production Systems

Cyber-physical production systems (CPPS) are modern production systems; they can be considered as productions integrated with cyber-physical systems technology[14]. The production itself can consist of two kinds of automated components – physical (e.g., machines) and digital (e.g., data analysis); the components process data, perform decision-making, and communicate with each other across all levels of production. They assist the interaction between machines, humans, and production. CPPS assists CPS in performing autonomously basing on data, connections (e.g., humans), and responding to changes[15].

2.4.3. Industrie 4.0

After the first stages of the industry, which were essentially focused on machines, equipment, and energy, the next goals are intelligent networking, controlling, and delivering without human interventions. Until now, the Information was collected in various IT systems and evaluated by humans, who then made adjustments. This process will change fundamentally as a result of Industry 4.0. In the digitalized future, all machines and the products will be equipped with sensors. Data can then be exchanged via networked systems and react to one another; they will communicate regularly and intelligently and continuously optimize their processes. Not only among themselves but with other systems: Production, distribution, development, even customers and suppliers will be integrated into the networked world[16]. For instance, since each machine knows how many components are still in stock, if a stock runs low, the system will automatically send an order to the supplier, who then sends out a replacement.

2.4.4. Cloud Manufacturing

Cloud manufacturing is a paradigm that transfers the idea of cloud computing, service-oriented, and IoT to manufacturing. It is described as "a new networked manufacturing paradigm that organizes manufacturing resources over networks (manufacturing clouds) according to consumers' needs and demand to provide a variety of on-demand manufacturing services via networks (e.g., the Internet) and cloud manufacturing service platforms[17]". In other words, the basic concept is to provide a virtualized database of resources with unified and on-demand access. The distribution of these resources among participants in this network is to be operated by a platform, allowing manufacturers and service suppliers to interact with each other efficiently. Different manufacturing resources can be detected and connected to a larger network. Moreover, the resources can be monitored and operated automatically using IoT technologies (e.g., RFID, sensor network, embedded system). Manufacturing resources are then virtualized and embedded into various manufacturing cloud services (MCSs) that can be retrieved, accessed, and implemented on a database through cloud computing- and service-oriented technologies[18].

2.4.5. Operator 4.0

Work environments and workforces have rapidly transformed in the past decades and will continuously be modified through 2025. Important factors that have assisted the transformation are, for example, Industrie 4.0, Smart Manufacturing, and Smart Factory. New architectures and engineering concepts for the twofold 'human-centered' and 'cyber-physical' manufacturing processes will be demanded from companies to fully adopt these paradigms successfully[19]. Therefore, Operator 4.0 are introduced, they are described as "a smart and skilled operator who performs not only - 'cooperative work' with robots - but also - 'work aided' by machines (by means of human cyber-physical systems, advanced human-machine interaction technologies and adaptive automation towards 'human-automation symbiosis work systems')"[20].

Goals of Operator 4.0:

- Build a reliable and interaction-based relationships between humans and machines.
- Provide advantage for smart factories with smart machines' strengths and capabilities.
- Empower 'smart operators' with new skills and gadgets.

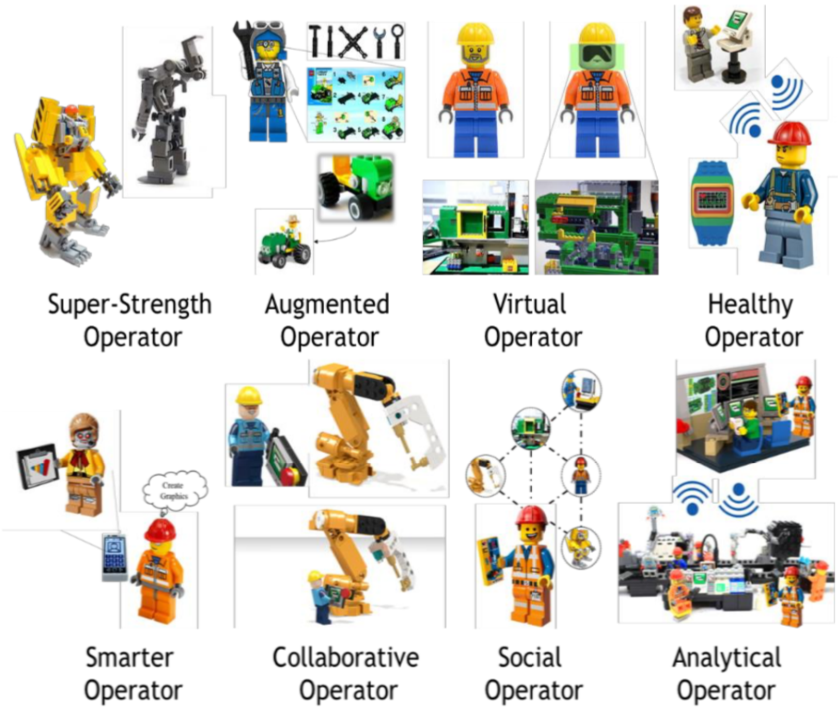

Figure 2.1.: Operator 4.0 Typology [20]

Some examples of Operator 4.0 typology which can support operators becoming 'smart operators are':

- **Super-Strength Operator = Operator + Exoskeleton:**
  Powered Exoskeletons are mobile wearables that are designed to be lightweight and flexible. They are powered by a system of electric motors, pneumatics, levers, hydraulics, or a combination of technologies that enable limb movements and increased strengths. Humans and technologies can collaborate with powered

exoskeletons, resulting in jobs being optimized and physical stress from the smart operator being minimized. Smart operators can lose their strength and mobility due to aging; powered exoskeletons can compensate by providing protection to the operators and supporting progress like lifting heavy objects or reaching items[20]. An example of this type of Operator 4.0 is the Robo-Mate systemcite; their Active Trunk module is designed to apply a supporting torque at the hip, thus reducing lower back musculoskeletal loading[21].

- **Augmented Operator = Operator + Augmented Reality:**
  Augmented Reality (AR) is described as "an interactive experience of a real-world environment where the objects that reside in the real world are enhanced by computer-generated perceptual information, sometimes across multiple sensory modalities, including visual, auditory, haptic, somatosensory and olfactory"[22]. A smart operator can benefit real-time from AR technology during operations, with advantages like faster cycle time, reduced failure rate, and error-proofing. AR technology can show real-time input to the smart operator on smart manufacturing processes and machines to optimize decision-making via integrated human-interface to manufacturing IT systems and assets. AR can provide advantages throughout different levels, from real-time sensor data about a machine at the machine level to allowing production supervisors to overview workstations in real-time monitoring at operations and enterprise levels[20]. The Satisfactory system, described as "an augmented-enabled ecosystem for increasing satisfaction and working experience in smart factory environments,"[23] is currently an early-stage example of this type of Operator 4.0.

- **Virtual Operator = Operator + Virtual Reality:**
  Virtual Reality (VR) is a digital, artificial world created with the help of special software and hardware. This digital reality goes far beyond previously existing 3D technologies, as it enables a more comprehensive experience. With lower risk and real-time feedback, VR can digitally simulate a manufacturing environment and allow the smart operator to communicate with any roles inside (e.g., a blueprint, a product, a machine tool, a robot, a production line)[20]. The computer-simulated technology can also provide the smart operator immersive augmented reality and simulation of real situations to improve decision-making. Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE) are examples of software tools that can assist the integration of VR technology[24]. Another example of a software tool that assists Operator 4.0 is the VISTRA system (“a virtual simulation and training system”), which is used for training operators and examining manual

assembly processes[25].

- **Healthy Operator = Operator + Wearable Tracker:**
  Wearable Trackers is a portable electronic device or application for tracking and reporting fitness- and health-related data such as running distance, energy consumption, and, in some cases, heart rate or sleep quality. GPS location and other biometrics data can also be monitored. The term is primarily used for body-worn electronic monitoring devices that sync (in many cases wirelessly) with a computer or smartphone for data collection over an extended period of time. Many people are now using aspects of wearable trackers with the rising of widely available devices such as the Apple Watch™, Fit-bit™, and Android wear™. Wearable trackers can facilitate positive change with increased efficiency, well-being, and proactive safety strategies for the workforce. For example, a smart operator can use 'personal analytics' to schedule and organize work-shifts, rest-breaks, and overtime based on health-related metrics, by tracking the physical workload (exercise activity) and cognitive workload (mental effort) during the work-shift, eventually setting alarms and notifications to maintain work stress and workload. Wearable devices to exploit biometrics awareness are a new aspect of this type of Operator 4.0; a smart operator can make smarter choices on health care self-management(e.g., fitness, wellness, medical[20]).

- **Collaborative Operator = Operator + Collaborative Robot:**
  Collaborative Robots (CoBots) are industrial robots that work together with humans. Differ from traditional industrial robot applications, CoBots applications are not separated from humans in the production process. The reliability of CoBots depends on lightweight construction materials, rounded edges, and inherent speed and force constraints. Moreover, as the robots have their own sensors that prevent injuries to smart operators, they can work in close proximity together with humans[26]. Safety barriers and other protective devices are then no longer necessary, as the robots switch off automatically when they come into contact with obstacles. These possibilities can provide advantages such as the recovery of shop-floor space generally lost due to safety barriers, improving the smart operator's efficiency and work satisfaction by allowing CoBots to assist a task more efficiently, and easing smart operator from stressful, non-ergonomic, and fragile operations[27].

- **Smarter Operator = Operator + Intelligent Personal Assistant:**
An *Intelligent Personal Assistant (IPA)*, also known as a voice assistant or mobile assistant, is software that allows information to be retrieved, dialogues to be held, and assistance services to be delivered by communicating in the form of a human-like voice. Speech analysis will be performed for speech recognition, furthermore, interpreting it linguistically, logically analyzing it, and, as a result, using speech synthesis to formulate a response. Some advantages from IPAs that can assist a smart operator are searching and retrieving repair or maintenance manual of a tool from a digital library based on voice request; planning and setting alerts for operations, product or asset management actions (e.g., re-certifications, check stocks, preventive maintenance); and identifying and diagnosing errors and issues; recommending methods and techniques for troubleshooting connected machines and systems. Examples of IPAs are Amazon™'s Alexa, Apple™'s Siri, and Android™'s 'Hey Google'. In particular, Amazon™'s Alexa allows users to access the API and use the existing features and resources to create new applications and services[28].

- **Social Operator = Operator + Social Operator:**
Enterprise Social Networking Networks (E-SNS) relies on the use of mobile and social interactive approaches to connect smart operators with smart factory tools at the shop-floor. With the help of the interaction between social network services ('social relations' among workforce) and social Internet of Industrial Things (social relations' between operators and smart things), decision-making can then be assisted, and production goals can be accomplished. Enabled by real-time mobile communication technologies, social networking between smart operators can enable workers to put their skills across the production line and to the shop-floor, promote ideas for product and process improvement, and encourage problem-solving by putting the right people together with the right knowledge[20]. The internal (intranet) forums and wikis can be seen as examples of this type of Operator 4.0. This still contains only parts of what is essentially expected from the Operator 4.0. In order to collaborate and interact, many firms and their staff use existing social network platforms (e.g., Facebook™or LinkedIn™) to do so.

- **Analytical Operator = Operator + Big Data Analytics**
Big Data Analytics is a process used to collect and analyze large datasets (Big Data) in order observe useful information such as hidden and unknown patterns, industry trends, and client preferences[29] . Big Data analytics provide multiple benefits for smart operators, for example, avoiding deceptive activities, improving

decision-making, and predicting corresponding events. In manufacturing, data analytics and machine learning are now fairly commonly adopted. In the near future, however, the increase in available data by cheap sensors and IIoT (connected devices) will provide much more efficient and realistic solutions[30]. Several types of Operator 4.0 mentioned above are linked to advanced data analytics; the 'collaborative operator' who requires image recognition to work closely with CoBots; the 'healthy operator' who utilizes on the analytics of collected bio-data and the 'smarter operator' applying Artificial Intelligence in a virtual assistant.

2.5. Interoperability in Manufacturing

Continuing to improve manufacturing nowadays calls for more than just finding solutions for faster operations or reducing production costs. Many manufacturing companies are hoping to find solutions for smarter operations. Therefore, interoperability and data exchange throughout connected factories are essential in enhancing manufacturing operations. Interoperability should assist smart manufacturing components and services in communicating and integrating with one another and efficiently exchanging data that are understandable by all interdependent systems. For instance, manufacturing equipment must maintain communication control systems for process updates and possible failures in shop-floor operations.

2.5.1. Definition

Few definitions regarding interoperability are listed below:

- Interoperability implies two or more parties' ability to make a perfect exchange of content, whether machine or human. Perfect means no noticeable distortions or accidental delays in origin, production, and use. [31]
- The ability of systems, units, or forces to provide services to and accept services from other systems, units, or forces and use the services so exchanged enables them to operate effectively together.[32]
- The ability of two or more systems or elements to exchange information and to use the information that has been exchanged. [33]

There are two general types of interoperability in smart manufacturing:

- The first type: vertical integration, e.g., interoperability between the development systems, the departments of the shop-floor, the operations performed by various equipment, the different shop-floor systems, etc.
- The second type: horizontal integration, e.g., interoperability between smart automation devices, cloud services, cloud platforms, and enterprises.

Successful interoperability implementation would result in the manufacturing industry running efficiently and smoothly, therefore reducing costs and increasing production and quality[34] .

2.5.2. Issues in Interoperability Implementations

Lack of unified data formats or standards, connectivity protocol in the IoT realm, and the massive variety of available devices are other interoperability challenges. The Manufacturing Interoperability Program at NIST (the National Institute of Standards and Technology) identified multiple factors impacting interoperability effectiveness[35] .

- Attempting data exchange between commercially similar or dissimilar systems.
- Attempting data exchange between same-vendor software but with different versions on each machine.
- Upward and downward compatibility between software versions.
- Misinterpreting definitions or the meaning of terms used to structure data exchange or interpret the meaning of that which is exchanged.
- Not using a recognized normative documentary standard upon which exchange data is formatted and based.
- No means of consistently testing self-declared conformant applications to ensure correct communication, one system to the other.

2.6. Reference Architecture

2.6.1. Definition

Reference architecture is a blueprint that contains documents of factors such as hardware, software, processes, specifications and configurations, as well as logical components and interrelationships. Reference architecture is considered as a framework that tracks the learning process of previous projects. Using a reference architecture, a project team can theoretically save time and prevent mistakes by learning from previous experiences. The specific structure, documents and administration should be adaptive, in order to respect the particular structure and needs of every organization. Furthermore, in order to ensure accuracy, the reference architecture should be regularly updated to provide new ideas. One example of well-known reference architectures is the Java Platform Enterprise Edition(Java EE) architecture, it assists enterprises whose systems are developed in Java with template solutions[36]. Another example is Eulynx, a reference architecture for railway signaling systems, it is intended to standardize control interfaces and railway signaling in order to reduce expense and installation time of signaling equipment[37].

2.6.2. Smart Manufacturing Reference Models(SMRMs)

Reference architectures have been categorized into types of architectures[38]:

- Physical architecture describes the components of a system (For example: automation devices, machines, software, departments);
- Functional architecture constitutes the set of functions and processes that are carried out by the systems (For example: RAMI4.0 functional layers);
- Allocated architecture provides a complete description of the system design between the functional architecture and the physical architecture;

In recent years, various reference models and architectures have been developed to deal with interoperability and implementation problems in the smart manufacturing sense. Platform Industrie 4.0[39] and the Industrial Internet Consortium (IIC)[40], two of the leading research organizations on matters related to manufacturing and the Industrial Internet, also proposed the most commonly used reference architecture. Reference Architecture Model for Industrie 4.0 (RAMI 4.0) is an architecture developed specifically for manufacturing industry applications, and IIC's Industrial Internet Reference Architecture (IIRA) was established for all industries connected to the Industrial Internet of Things (IIoT)[41]. IBM's Industrie 4.0 architecture and the NIST service-oriented architecture are other influential architecture models in this context. An analysis of the most important reference architectures will follow.

2.7. Reference Architecture Model Industrie 4.0 (RAMI4.0)

RAMI4.0, also known as Reference Architecture Model Industrie 4.0, is the most common and well-known reference architecture for smart manufacturing. RAMI4.0 was developed by the German Electrical and Electronic Manufacturers' Association. The reference architecture is a standardized model for all components, it has the purpose of guaranteeing data are exchanged efficiently by all parties in the Industrie 4.0 system. Another purpose of RAMI4.0 is making complicated processes easier to understand by breaking them down into packages. RAMI4.0 is a three-dimensional model, that consists of three axes, including "layers", "lifecycle and value stream", and "hierarchy levels"[42][43].

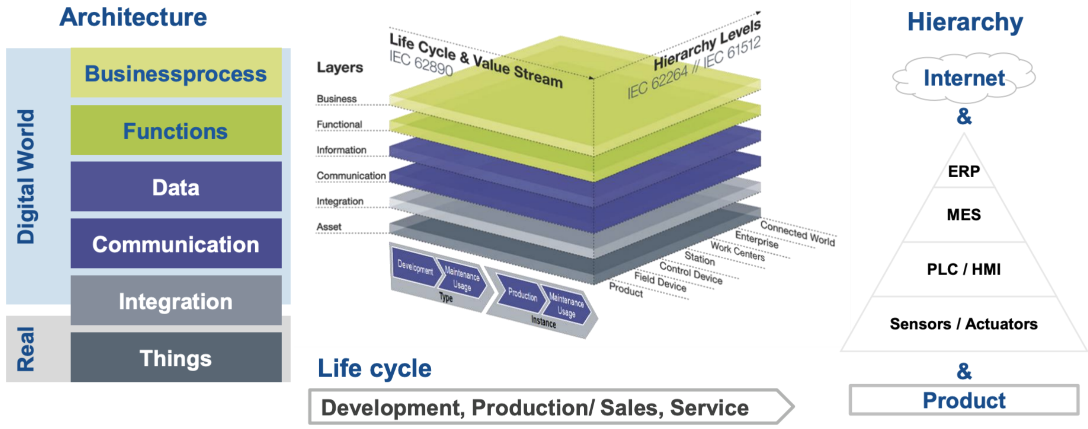

Figure 2.2.: Reference Architecture Model Industrie 4.0 [44]

2.7.1. The Layer Axis

The layer axis consists of layers of[42][43]:

- Asset
- Integration

- Communication
- Information
- Functional
- Business

Asset

The *asset* layer represents a product, a tool, an element, or a physical facility, that actually exists in the physical world. Every item in the asset layer should have a digital form of itself in the higher-level layers. Yet, every item in the digital world doesn't necessarily has an equivalent item in the asset layer.

Integration

The *integration* layer serves the purpose of transforming a physical asset into digital item, it also represents the Administration Shell. The resources and process-related features that make the asset accessible for its considered purpose are located in the integration layer. Examples of common functionalities in the integration layer are:

- A human-machine interface
- Identification of technical elements via sensor and signal converters
- Generating events from the real world in the virtual world
- Collecting resources from physical objects, hardware, software, documents etc.

Communication

The *communication* layer helps assets communicate with one another. Data that were collected from one asset, for example, the information and functions, can then be obtained by other assets via Industrie 4.0 components. Which data is obtained, the location of the data and when it is allocated are all descriptions of the communication layer.

Information

The *information* layer identifies assets' data that are adopted and exchanged between components. This layer also assists quality assurance, event pre-processing, and ensuring data integrity. Furthermore, via service interfaces it can deliver data that have been structured and organized.

Functional

The *functional* layer runs and integrates the standard functions and concrete applications of the assets. This layer also helps develop an environment for services and business processes.

Business

The *business* layer serves the purpose of mapping the business models, identifying regulations, and organizing services based on the data provided from the information layer.

2.7.2. Life cycle and Value stream Axis

The left horizontal axis represents the life cycle of facilities and products, based on IEC 62890 for life-cycle management. This axis is used to characterize an asset at a given point in time during its lifespan, from its creation right up to its disposal. The lifecycle and value stream axis consists of stages of *type* and *instance*, where the *type* stage is further categorized into *development* and its *maintenance/usage*, and the *instance* stage is categorized into *production* and its *maintenance/usage*[43][38].

Type stage

Focusing on the *type* stage, development defines construction, computer simulation, and prototype in a construction plan. Maintenance/usage consists of software updates, instruction manual, and maintenance cycles in a construction plan.

Instance stage

Furthermore, in the *instance* stage, production represents product, data, and serial number during the production. Whilst Maintenance/usage defines usage, service, and maintenance in a facility management.

2.7.3. Hierarchy Axis

Indicated on the right horizontal axis are hierarchy levels based on IEC 62264 and IEC 61512 standards. Hierarchy axis deals with location, functional hierarchy from the product to the connected world as the last stage of Industrie 4.0 development with all enterprises, customers and suppliers connected. The hierarchy axis consists of connected world, enterprise, site, area, work centers, work units or station, control device, field device, and product[42][43].

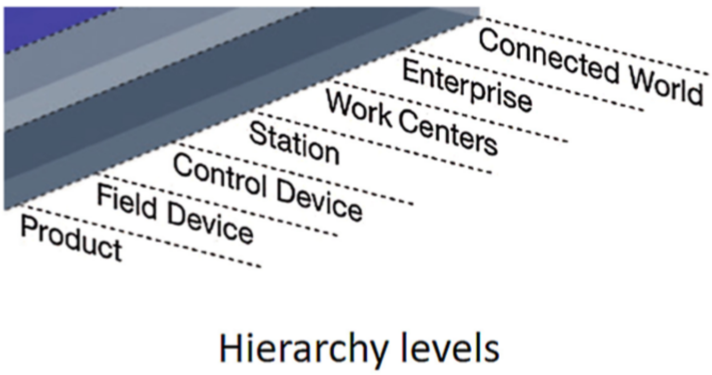

Figure 2.3.: Hierarchy Levels of RAMI4.0 [44]

2.8. Smart Manufacturing Standard Landscape (SM2)

ISO-IEC Smart Manufacturing Standard Landscape (SM2) is designed by the ISO-IEC Smart Manufacturing Standards Map Task Force. Few characteristics of SM2 (Figure ?) are that SM2 represents a cube model, and the reference architecture bears a resemblance, with some exceptions, to the RAMI4.0 model. For instance, compared to the asset layer and integration layer of RAMI4.0's layer axis, SM2 only contains the resource layer. Therefore, the resource layer should serve the purpose of sorting physical assets (similar to RAMI4.0's asset layer) and digitalizing the assets to make them accessible for their purpose. A noticeable difference between RAMI4.0 and SM2 is the production system life cycle; it is identical to PLC's "product" lifecycle since the production system is considered a "product" of the builder of the production system[43].

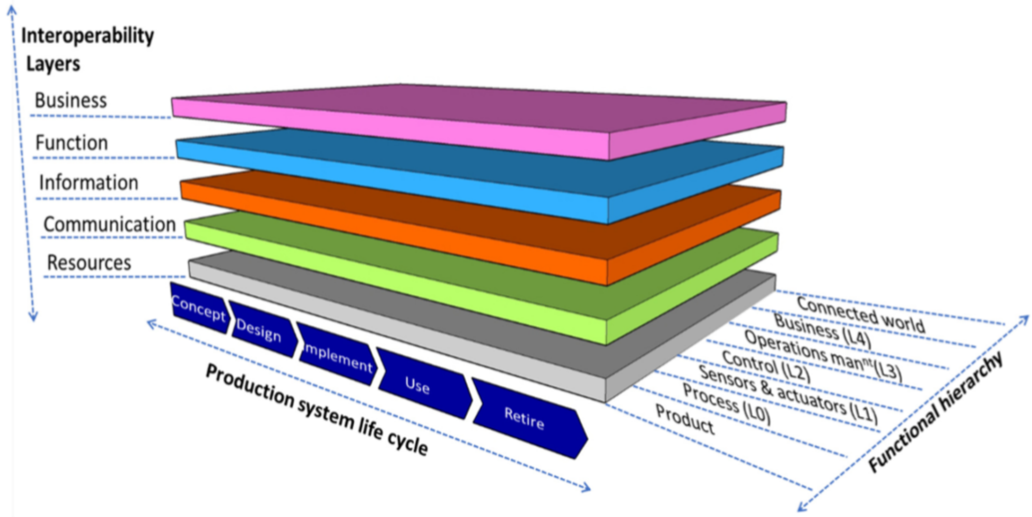

Figure 2.4.: Smart Manufacturing Standard Landscape (SM2)[43]

2.9. KSTEP

The reference architecture KSTEP, origin from Korea, is a three-dimensional cube framework. The KSTEP framework's main parameters are the same as those of RAMI4.0, except for the lifecycle axis and the telecommunication physical hierarchy axis. The lifecycle axis of KSTEP, which consists of engineering, procurement, construction, operation, and maintenance, embraced the FIATECH lifecycle[43]. The FIATECH lifecycle is a variant of ISO 10303-239 PLCS (product lifecycle support)[45]. The lifecycle axis indicates the lifecycle of a thing (e.g., a product or an asset).
On the other hand, the telecommunication physical hierarchy axis adopted the three-tier Industrial Internet of Things (IIoT) system architecture, including the Edge tier, Platform tier, and Enterprise tier. This axis can be used to represent physical hierarchy such as communication cables or wireless network[43].

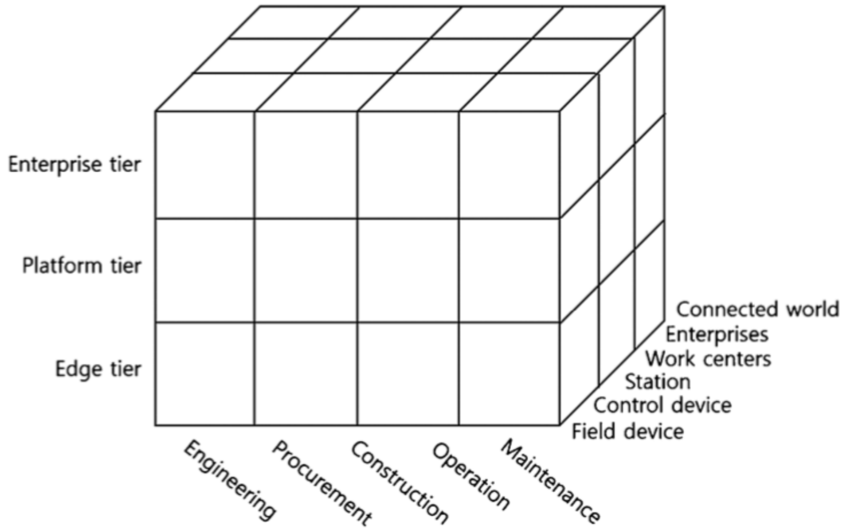

Figure 2.5.: KSTEP [43]

2.10. Intelligent Manufacturing System Architecture (IMSA)

China's Intelligent Manufacturing System Architecture (IMSA) is a three-dimensional reference architecture, it is conceptually similar to RAMI4.0. IMSA consists of three axes: the lifecycle axis, the intelligent characteristics axis, and the system hierarchy axis. The lifecycle axis describes more of a product than a production system and the phases of value creation. The intelligent characteristics axis represents the new information and communications technology for Industrie 4.0. The system hierarchy axis describes the organizational levels of manufacturing processes. When compared to RAMI4.0, IMSA's system hierarchy axis does not include the product as one level[43][46].

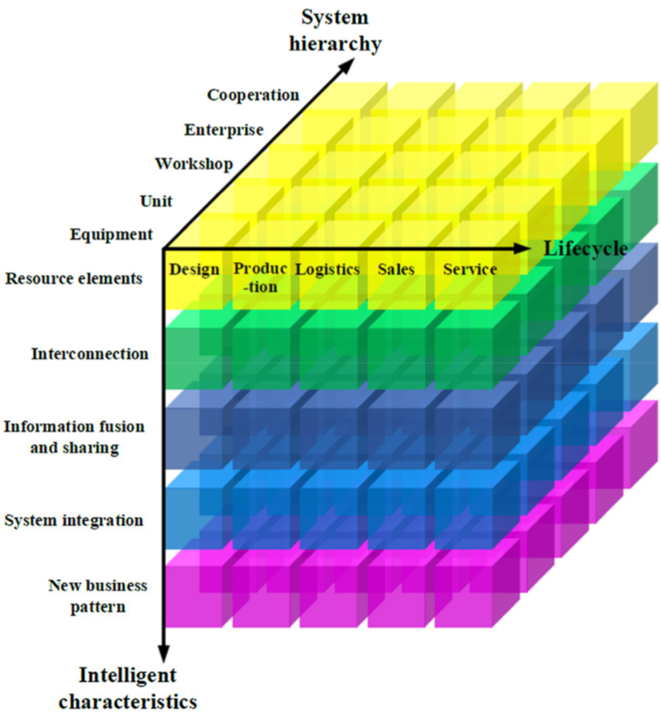

Figure 2.6.: Intelligent Manufacturing System Architecture (IMSA) [43]

2.11. NIST Smart Manufacturing Architecture

The National Institute of Standards and Technology (NIST) provided a four-dimensional model with different specifications for Smart Manufacturing systems, with the help of Advanced Manufacturing Series' (AMS) standards. The NIST model consists of three lifecycle axes: the product lifecycle, the production life cycle, and the business life cycle. Moreover, a manufacturing pyramid exists at the crossing point of three lifecycle axes (the center)[43][38].

- Production lifecycle contains *design, build, commission, operation and maintenance,* and *decommission and recycling*. It defines the lifecycle of production equipment. Moreover, from the manufacturing equipment providers' point of view, the production lifecycle bears a resemblance to RAMI4.0's Life Cycle and Value Stream.

- Product lifecycle contains *design, process planning, production engineering, manufacturing, use and service,* and *recycling*. It defines the six phases of product development lifecycle. The production lifecycle's perspective is also similar to RAMI4.0's Life Cycle and Value Stream.

- Business lifecycle contains *source, plan, and delivery* and *return*. This viewpoint addresses the Supply-Chain Operations Reference model's plan-source-make-deliver-return phases.

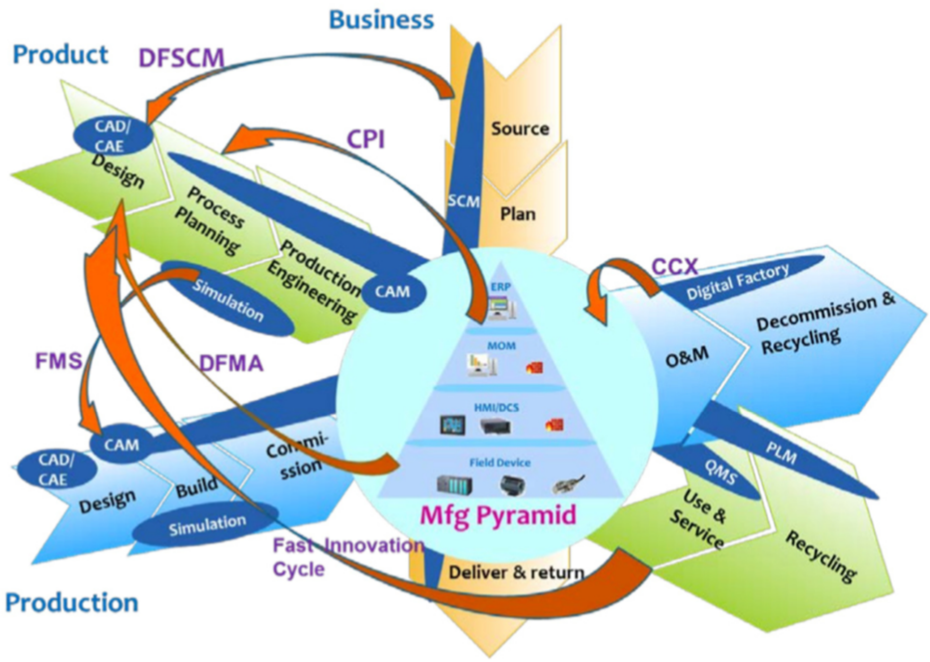

Figure 2.7.: NIST Smart Manufacturing Architecture [43]

2.12. ISO/ TC 184

ISO/TC 184 Big Picture shows similarities to NIST, it can be considered as a simplified version of NIST model. The reference architecture consists of the life cycle lifecycle (lifecycle of product), the value chain lifecycle (the outsourcing of product components) and the enterprise level[43].

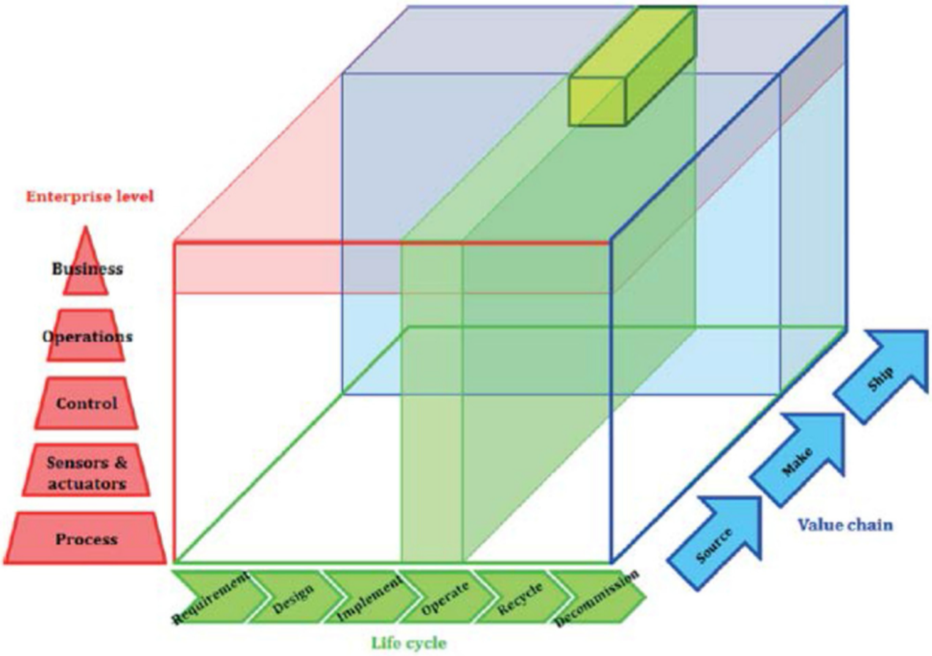

Figure 2.8.: ISO/ TC 184 Big Picture [43]

2.13. Industrial Value Chain Reference Architecture (IVRA)

Industrial Value Chain Reference Architecture is a reference model that originated from Japan; it consists of three different views: asset view, activity view, and management view[42][47].

- *The Asset view* shows valuable assets to the manufacturing process; there are four classes of assets in this view: personnel assets (e.g., plant workers), process assets (e.g., knowledge of the operation), product assets (e.g., the outcome of manufacturing) and plant assets (e.g., equipment, machines, and devices).
- *The Activity view* covers activities that are performed by human and equipment at manufacturing sites; there are four basic classes of activities: Plan (e.g., make a list of actions to be executed by a deadline), Do (e.g., executing activities), Check (e.g., examine whether a goal has been achieved), and Action (e.g., defining the ideal situation and tasks for fixing any problem).
- *The Management view* represents management-relevant purposes and indices; there

are four management classes: quality management (e.g., quality related to humans and methods), cost management (e.g., costs from consumed materials, invested service, and consumption of energy), delivery accuracy management (e.g., delivery date, time, location, and method), and environment management (e.g., managing emission of toxic substances).

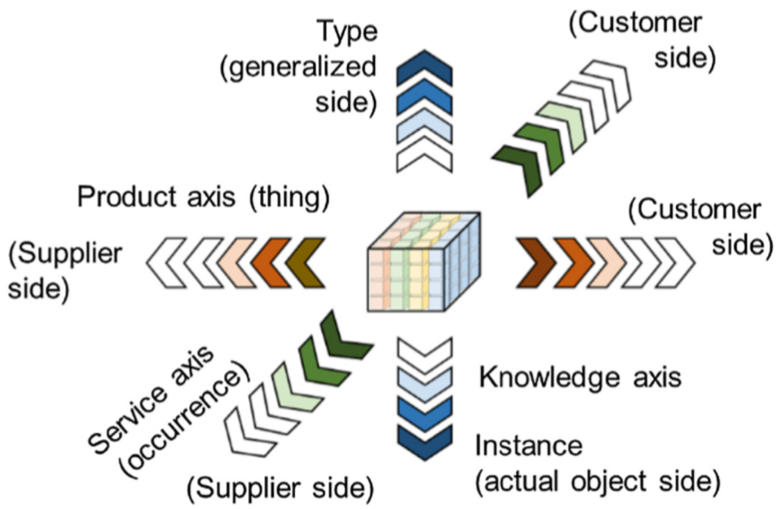

Figure 2.9.: Industrial Value Chain Reference Architecture (IVRA) [43]

2.14. Scandinavian Smart Industry Framework (SSIF)

Scandinavian Smart Industry Framework (SSIF) consists of three space axes and three lifecycle time axes. The three space axes are product, production, and business. The three lifecycle time axes that correspond to each of the three space axes are product lifecycle, production lifecycle, and business lifecycle. When compared with RAMI4.0, the product dimension bears a resemblance to the lifecycle and value stream axis, the production dimension is similar to the hierarchy axis, and the business dimension is equivalent to the layer axis[43].

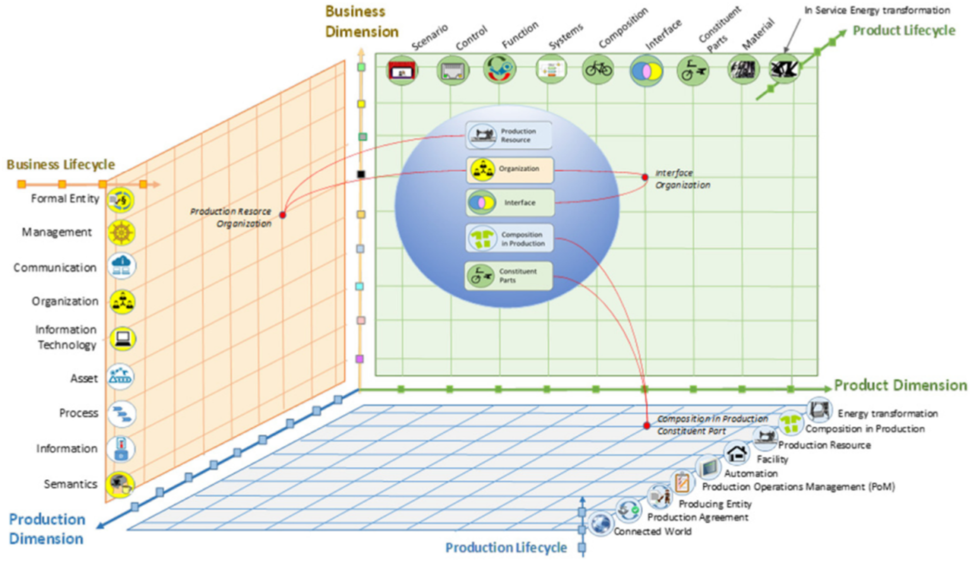

Figure 2.10.: Scandinavian Smart Industry Framework (SSIF) [43]
[h!]

2.15. International Data Spaces Reference Architecture Model (IDS-RAM)

IDS is at a higher level of abstraction than traditional architecture models of concrete software solutions, focusing on the generalization of concepts, features, and overall processes involved in developing a stable "network of trusted data." IDS is built with five layers[2]:

- The Business layer categorizes the various roles that can be exercised by the International Data Spaces participants and specifies the essential tasks and interactions relevant to each of these roles;
- The Functional layer specifies the International Data Spaces' functional specifications and the concrete characteristics generated from these;
- The Process layer defines the interaction between the various components, offering a detailed view of the Reference Architecture using the BPMN notation;
- The Information layer enables the (semi-)automated exchanging of digital resources within the distributed parties' trusted ecosystem while ensuring the data

sovereignty of the Data Owners;

- The System layer analyzes the logical software components, taking into account factors such as the integration, installation, implementation, and extensibility of such components;

Furthermore, IDS consists of three perspectives that need to be applied across all five layers: *Security, Certification,* and *Governance.*

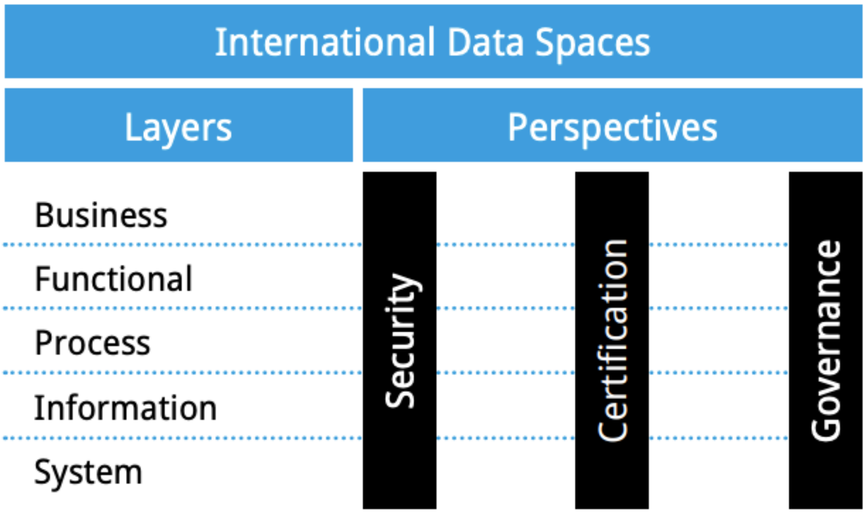

Figure 2.11.: International Data Spaces Reference Architecture Model (IDS) [2]

2.16. FIWARE

FIWARE is a framework that is designed for open-source platform components[48]. The open-source platform components can be incorporated along with other third-party platform components to facilitate Smart Solutions' production. FIWARE offers an entirely open and public architecture and a collection of standards to allow developers, service providers, companies, and other organizations to develop products that suit their needs. FIWARE can be categorized into five different specialties[48]:

- **FIWARE NGSI (Next Generation Service Interface) API**: FIWARE-NGSI manages the entire lifecycle of context data (including updates, queries, registrations, and subscriptions);
- **Core Context Management (Context Broker)**: Including NGSI context broker implementations (e.g., Orion, Scorpio, and Stellio), developing context-aware applications (e.g., STH-Comet, Draco, and Cygnus), and big data analysis (e.g., Cosmos);
- **Interface to IoT, Robotics and Third-Party Systems**: including connecting to external systems (e.g., Oliot), connection to IoT (e.g., IoT Agents and OpenMTC), connecting to robots (e.g., FIROS, Fast DDS, and Micro XRCE-DDS), and exchanging documents (e.g., Domibus);
- **Context Processing, Analysis, and Visualization**: including real-time processing of context events (e.g., Perseo), creating applications dashboards (e.g., Wirecloud), real-time processing of media streams (e.g., Kurento, OpenVidu), and cloud edge (e.g., FogFlow);
- **Data/API Management, Publication, and Visualization**: including publication and monetization of context information (e.g., CKAN extensions, Data/API Biz Framework, and IDRA) and handling authorization and accessing control to APIs (e.g., Keyrock, Wilma, and APInf);

2.16.1. Context Broker

The context broker plays a vital role in the communication between context providers and context consumers. Context data can be collected from IoT devices or robots, then passed into the context broker. The context broker, which retains the system's current state, continues passing the context data for further processing, such as business intelligence or big data analysis[49].

An example of the context data's transition between an IoT device and a user via IoT agent and context broker is demonstrated(Figure 2.12). IoT agent is a component that administrates IoT devices' data; the data is sent via device protocol. IoT agent will translate device protocol into NSGI to forward IoT devices' data to the context broker. The data can then be retrieved from the user, therefore assist the user with further actions. The user can either send data to activate the device or retrieve the information from the device that will be processed. In conclusion, the communication between the user and IoT devices can be performed in both directions[50].

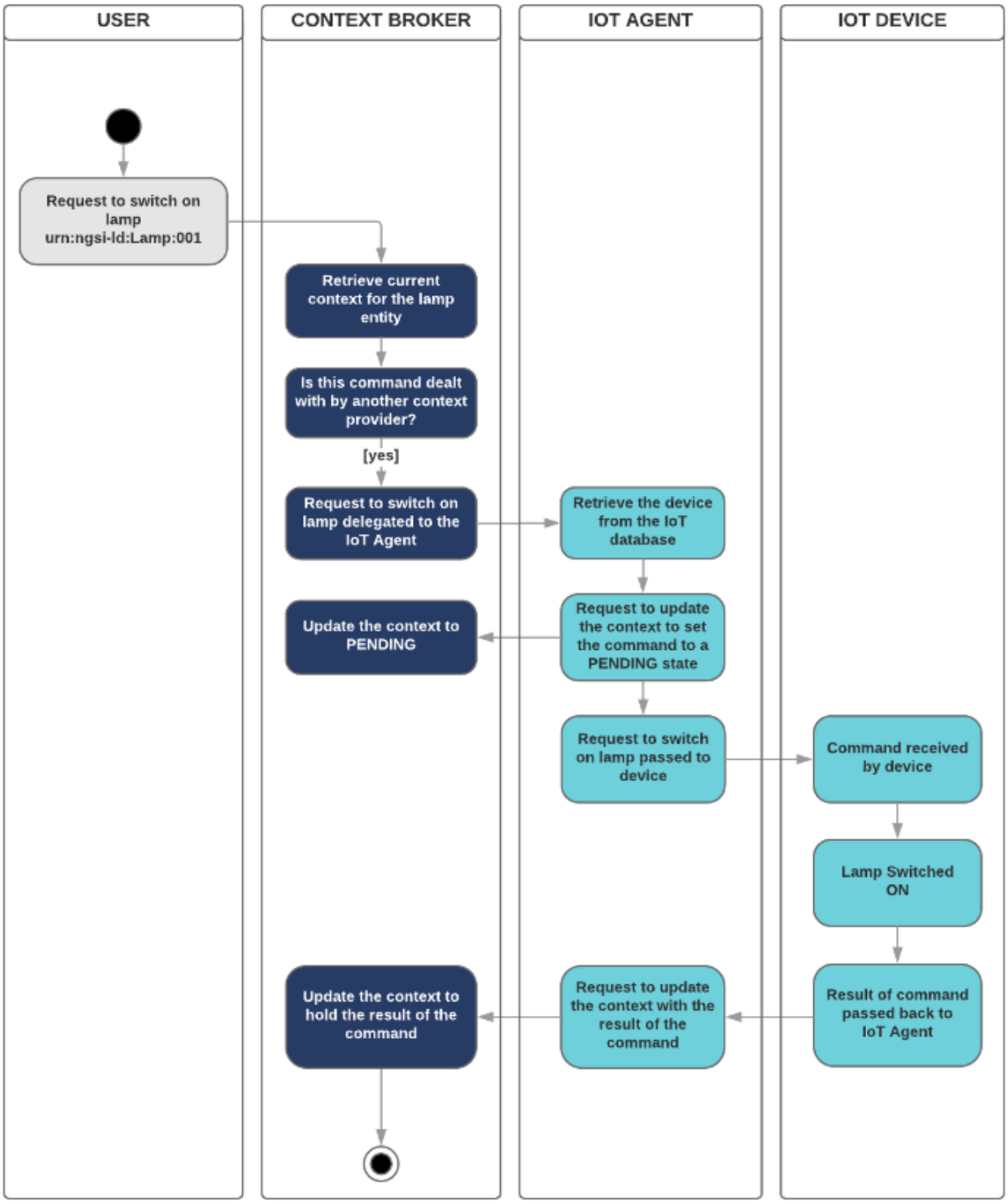

Figure 2.12.: An Example of Context Data's transition between IoT Device and User via Context Broker and IoT Agent[50]

2.16.2. Data Model

"A Data Model (or datamodel) is an abstract model that organizes elements of data and standardizes how they relate to one another and the properties of real-world entities"[51]. A vital purpose of a system is to have the ability to manage a vast amount of data, regardless of the data is structured or not. Data models serve as a foundation for the implementation of information systems by defining and formatting data. Furthermore, related applications may exchange data if the same data systems are used to store and view data. Smart data models are data models specifically constructed for application domains, such as Smart Cities, Smart Environment, and Smart Water. The data models are designed to be compatible with FIWARE NGSI V2 and NGSI-LD[52].

An example of an existing smart data model: Alert data model. The model's goal is to assist users in generating updates or take further actions in response to these alerts. Warnings may be used to submit notifications about traffic jams, weather conditions, and various triggered situations. The specific requirements and restrictions of existing smart data models are listed on the corresponding website, this helps the user to follow the JSON Schema. For example, when "severity" is established, only specific values (informational, low, medium, high, critical) are allowed. Another example is the attribute "alertSource," which is mandatory when creating an alert data model[53].


```json
{
    "id": "Alert:1",
    "type": "Alert",
    "category": {
        "value": "traffic"
    },
    "subCategory": {
        "value": "trafficJam"
    },
    "validTo": {
        "value": "2017-01-02T10:25:55.00Z"
    },
    "description": {
        "value": "The road is completely blocked for 3kms"
    },
    "location": {
        "type": "geo:json",
        "value": {
            "type": "Point",
            "coordinates": [-3.712247222222222, 40.423852777777775]
        }
    },
    "dateIssued": {
        "type": "DateTime",
        "value": "2017-01-02T09:25:55.00Z"
    },
    "alertSource": {
        "value": "https://account.lab.fiware.org/users/8"
    },
    "validFrom": {
        "type": "DateTime",
        "value": "2017-01-02T09:25:55.00Z"
    },
    "severity": {
        "value": "high"
    }
}
```


Figure 2.13.: Alert Data Model from FIWARE's Smart Data Models[53]

3. Concept of Human Data in Reference Architectures

Successful application of the most advanced complex systems relies on human components as part of their device architectures. Although these human factors further complicate these systems, most of these complex device architectures can also be well-designed using traditional reference architectures' techniques and methodologies. There are various reference architectures for smart manufacturing nowadays, most of them bear a resemblance to RAMI4.0. In this chapter, the reference architectures will be further investigated and compared with one another. The main goal is to determine where the human components and their data are involved in every model.

In this chapter, six reference architectures will be analyzed based on the engagement of humans. Three-axes model are commonly embraced by many reference architectures, therefore reference models can be effectively presented as a box or a cube. By mapping every reference architecture into a cube mode, human components and human data can be easily identified throughout every model's axes. Furthermore, this method can assist the process of choosing a suitable reference architecture to implement. Even though most of the reference architecture mentioned bears a resemblance to RAMI4.0, there are still plenty of variations that can be spotted in every model.

3.1. RAMI4.0

RAMI4.0 is the most common and well-known reference architecture; it is often compared with newly arisen reference architectures. Human data that are possibly collected from smart devices based on operator 4.0 typology can be viewed as the 'Asset' in the layer axis. Human components can be located in all life-cycle management processes; therefore, human components are scattered throughout the life cycle and value stream axis. Lastly, human operators can be discovered in 'Work Centers' and 'Station' regarding the hierarchy axis. A work center contains one or several work stations and is responsible for high-level manufacturing elements (e.g., production line, storage zone, process cell).

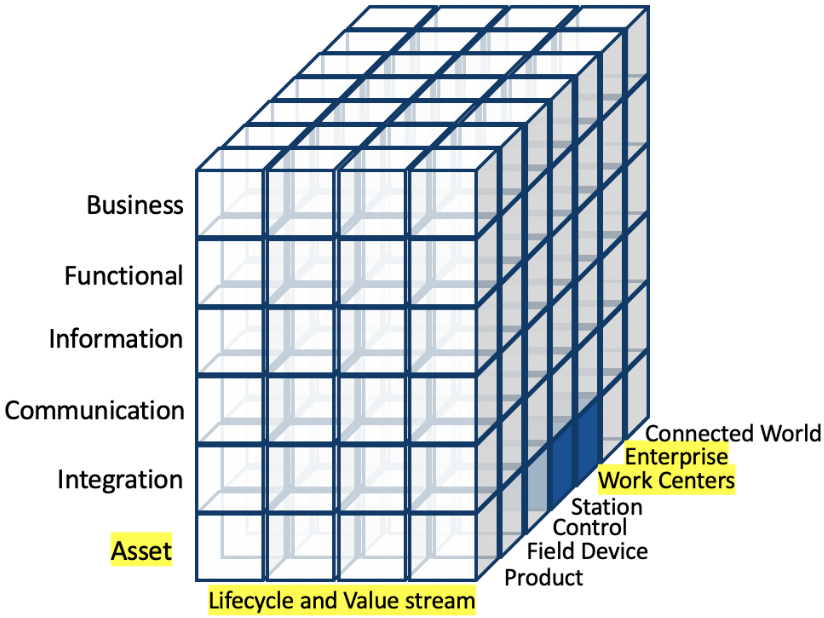

Figure 3.1.: RAMI4.0 Cube Model (Human components marked as yellow)

3.2. SM2

Moving on to SM2, this model is similar to RAMI4.0; nevertheless, there are still differences that can be observed. For example, as opposed to identifying human data as 'Asset,' SM2 refers to human data as 'Resources.' The idea of naming human data as 'Resources' is much more accurate than 'Asset.' Furthermore, instead of existing on different levels of the functional hierarchy axis, human operators can be easily located in 'Operations man.' Lastly, human components can be found in all sections of a product's life cycle regarding the production system life cycle.

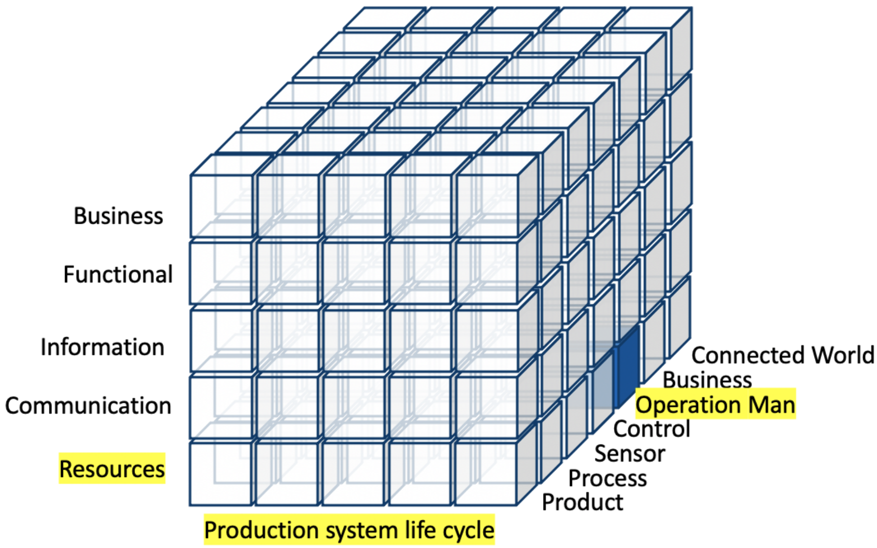

Figure 3.2.: SM2 Cube Model (Human components marked as yellow)

3.3. KSTEP

The Korean KSTEP reference architecture has a much different approach concerning 'Telecommunication hierarchy,' with three-tier IIoT system architecture. Human data exists in the 'Edge Tier,' human components are located throughout the whole lifecycle axis and human operators functioning in the 'Work centers' and 'Station' level, similar to RAMI4.0.

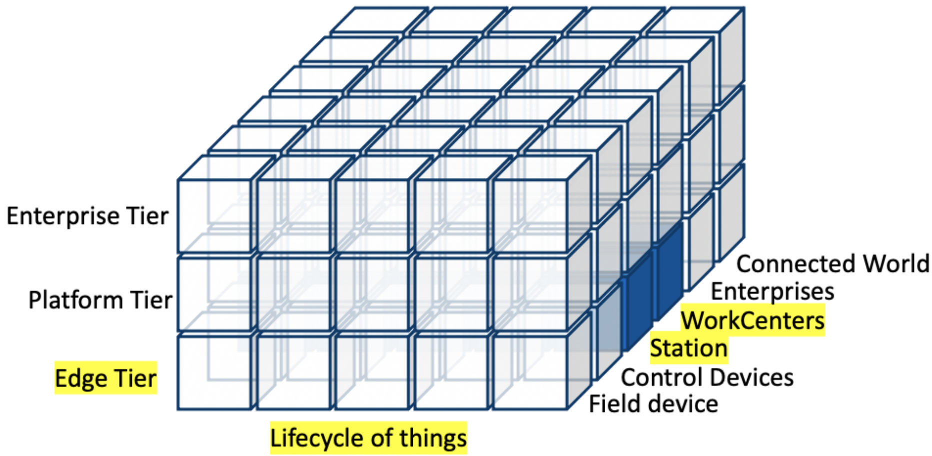

Figure 3.3.: KSTEP Cube Model (Human components marked as yellow)

3.4. IMSA

IMSA is likewise identical to RAMI4.0, even though the IMSA's cube model looks slightly different (with 'Resource Elements' starting from the top layer). Human data exists in the 'Resource Elements' layer, human components are found on the IMSA' lifecycle axis, and 'Workshop' (Similar to RAMI4.0's 'Work Centers') and 'Unit' (Similar to RAMI4.0's 'Station') consist of human operators.

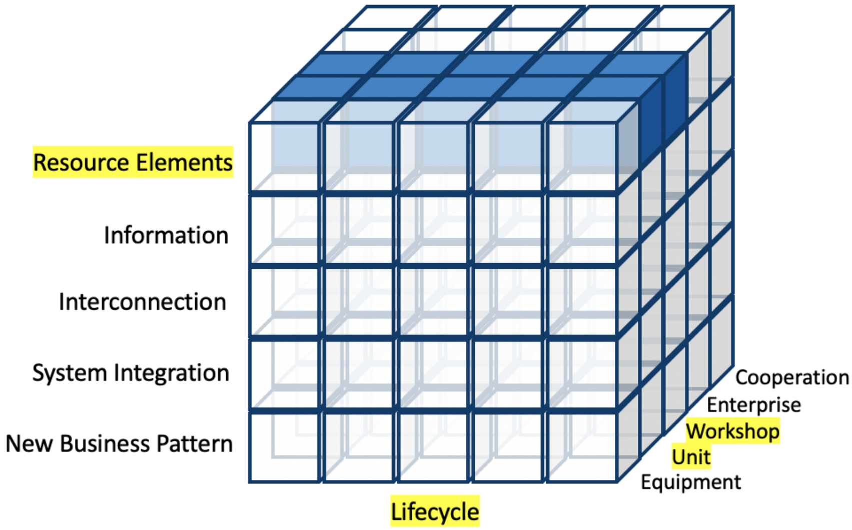

Figure 3.4.: IMSA Cube Model (Human components marked as yellow)

3.5. ISO/TC 184

Compared to SM2, KSTEP, and ISMA, ISO/TC 184 Big Picture shows no noticeable similarity to RAMI4.0; the model is similar to the NIST reference architecture. After observation, human data can be located in 'Operations' and 'Control' on the enterprise level, human components are found throughout the whole lifecycle of a product, and human operators are responsible for the 'Make' process of product components' outsourcing.

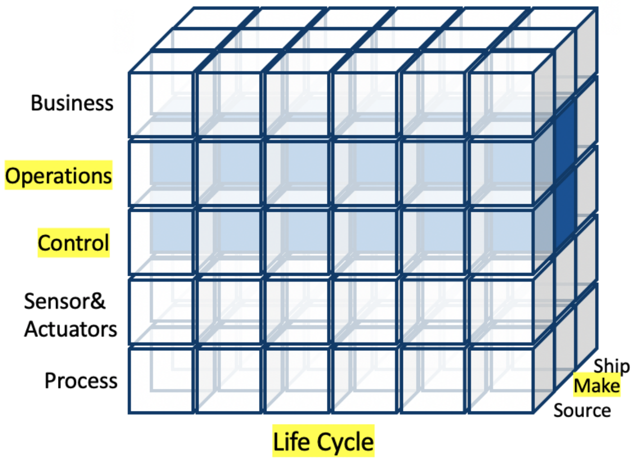

Figure 3.5.: ISO/TC 184 Cube Model (Human components marked as yellow)

3.6. IVRA

Also bearing no resemblance to RAMI4.0, IVRA consists of three different views. Regarding the Asset View, human data can easily be identified in the 'Personnel' assets (e.g., plant workers). Human operators are then located in the 'Quality' management (e.g., quality related to humans and methods) concerning the Management view. Lastly, within the Activity View, human components can be found in 'Do,' 'Check,' and 'Action.'

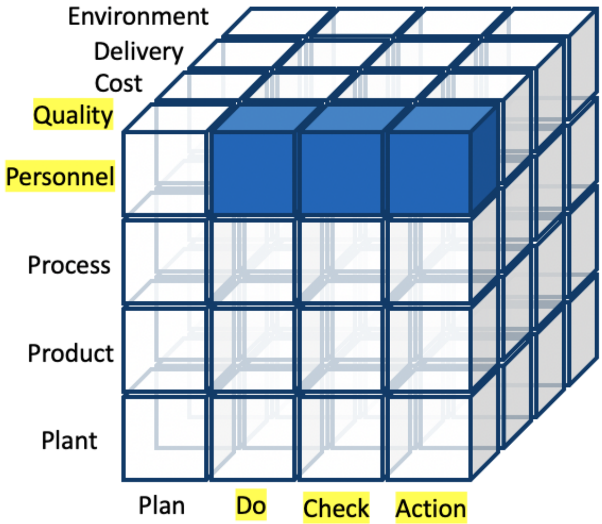

Figure 3.6.: IVRA Cube Model (Human components marked as yellow)

3.7. Decision on Implementation

While observing the reference architectures mentioned in this chapter, IVRA and ISO/TC 184 were quick to be eliminated. Both models have a different approach compare to RAMI4.0. Furthermore, it seemed harder to identify these two models' human factors. For example, regarding ISO/TC 184, the human data can be both identified within 'Operations' and 'Control.' To spot human factors in the reference architecture is essential for the implementation process and this thesis's concept.

The remaining reference architectures are RAMI4.0, SM2, KSTEP, and IMSA. When comparing RAMI4.0 and SM2, a noticeable flaw on RAMI4.0 was soon discovered, the human data are regarded as ’Asset’ in RAMI4.0, ’Asset’ is a general form of physical objects. SM2’s ’Resource’ layer is much more accurate considering a human-centered approach. Moving on to KSTEP, even though human data are located in the ’Edge tier’ on the telecommunication physical hierarchy axis, the ’Edge tier’ layer includes not only human data but also other objects and processes. ’Edge tier’ resulting in more challenging to particularly identify human data and access them. Subsequently, comparing IMSA to SM2, it is noticeable that human operators exist both on the ’Workshop’ layer and ’Unit’ layer of the IMSA model. While on the SM2 model, human operators are easily discovered at the ’Operation man’ level.

3.8. Data Modeling

The method of constructing a data model for an information system using structured techniques is known as data modeling. Data modeling is a method for defining and analyzing the data specifications for supporting processes[51]. After determining which reference model to implement, the next step is to do thorough research on existing data models. The goal is to figure out what kind of data models can assist the communication between different layers of SM2, specifically regarding human data generated by operator 4.0. Various existing smart data models can be found on the Smart Data Models[54](A collaborative initiative impulsed by FIWARE foundation, TMForum and IUDX) homepage. Suitable data models can already be determined right off the bat, such as the Alert data model and the User data model. Regardless, these are only the fundamental data models that describe general data. There are no specific data models that can determine human data, such as risk factor or trace pen pose. Therefore, creating data models is an essential step to assist the transition of human data throughout the reference architecture.

Smart Data Models has provided a valuable platform to help users generate basic versions of data model. Users can fill in the appropriate subject, name of data model, title of data model, global description, properties’ names, NGSI type, and data type into an easy understanding spreadsheet(Figure 3.7) provided by Smart Data Models[55]. First, the healthy operator is examined, and two explicit data models can already be determined – the risk level and the health level. The risk level contains essential properties such as riskFactor, signOrSymptom, and rulaRiskLevel. The health level consists of properties such as heartrate, bloodpressure_h (The number refers to the amount of pressure in your arteries during the contraction of your heart muscle.), bloodpressure_l (The bottom number refers to your blood pressure when your heart muscle is between beats.), and bloodsugar. Next, the augmented operator is examined, it is concluded that a data model regarding the trace pen pose can be implemented. The data model of trace pen pose contains properties such as pose position (x, y, z) and pose orientation (x, y, z, w), both presented with an array. The properties can vary depending on what kind of human data is provided. Nevertheless, this basic version of a data model presents that data models can be generated to implement human data adequately.

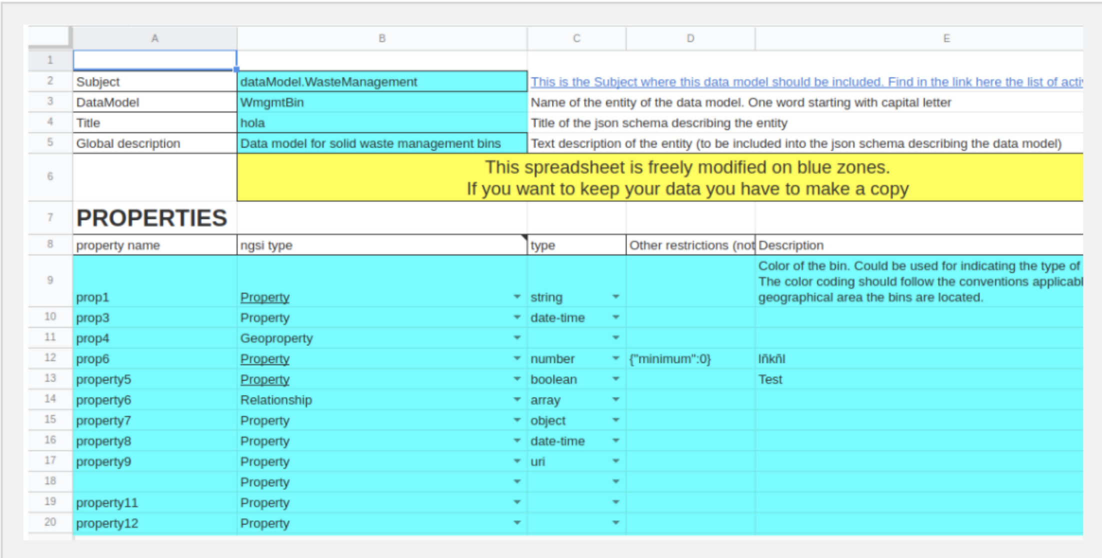

Figure 3.7.: Sample Spreadsheet from Smart Data Models[55]

4. Implementation

Subsequent to the last chapter, the reference architecture SM2 is determined for the implementation with human data. The further action is to create a use case and attempt to implement it with Python and, most notably, with FIWARE. The goal is to analyze what benefits and restrictions will be expected while communicating human data throughout different layers from SM2. Furthermore, to examine the data models created for operator 4.0 and determine whether improvements can be made to assist future data models.

4.1. Use Case

The concept of the use case is to collect data from an operator/human, particularly data that was produced by operator 4.0. The chosen operator 4.0 for the use case are the healthy operator and the augmented operator. The human data provided by the healthy operator is the risk level, which is based on the rula risk level (1-2: negligible risk, 3-4: low risk, 5-6: medium risk, 6+: very high risk). On the other hand, the augmented operator provides the trace pen pose, which contains the pose position (x, y, z) and the pose orientation (x, y, z, w). After retrieving the human data, the human data should be implemented into the data models, and the transition between layers via FIWARE context broker will be examined.

4.2. Tools

4.2.1. Docker

Docker is a software that allows applications to be virtualized as containers. Applications, along with their dependencies, can be packed into an image. The packed application can then be run in a Docker container with the help of a particular engine. Compared to virtual machines, docker is light weight; it only carries OS processes and dependencies necessary to execute the code[56]. To maintain the context data's stability, the Orion-LD Context Broker currently uses open-source MongoDB[57]. In order to link the Orion-LD context broker to MongoDB, the docker-compose must be installed.

Visual Studio Code offers docker as a container tools extension[58], therefore assisting the connection between the context broker and the database. A benefit that docker offers is to allow all users to access FIWARE, regardless of which OS is currently in use.

Figure 4.1.: Relation between Orion Context Broker and MongoDB[59]

4.2.2. Conversion of Human Data

After receiving the human data, the data is most likely to arrive in the form of a ROS bag. The fundamental step to implementing human data into a data model is to convert the bag file into a JSON file. This step is divided into two parts:

- **Convert ROS bag file into a CSV file**[1]
- **Convert CSV file to a JSON file**[2]

Both of these parts are performed with the help of two different GitHub repositories. Once the bag file is converted to a CSV file, users can already have a clear overview of what topics are included in the data, therefore determine what data can be retrieved and what data to set aside. After the second part of the conversion, the human data as JSON file can be easily accessed by the python code and therefore implemented into the designed data models.

[1] `https://gist.github.com/garaemon/831611`
[2] `https://gist.github.com/bradwbonn/10145604`

4.2.3. NGSIv2 REST API

The elements in the NGSI data model are context entities, attributes, and metadata (Figure 4.2). NGSIv2 REST API assists the transition of context data. Therefore, it is required when issuing commands such as creating an entity, updating context data, and retrieving context data from the context broker. Some primary commands are POST(submitting entities), PUT(update or create an entity), PATCH(updating entities), and GET(retrieving data from entities)[60].

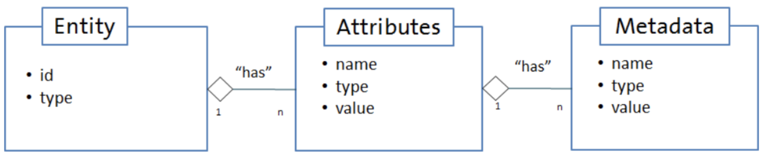

Figure 4.2.: Main Elements of the NGSI Data Model[60]

4.3. Architecture

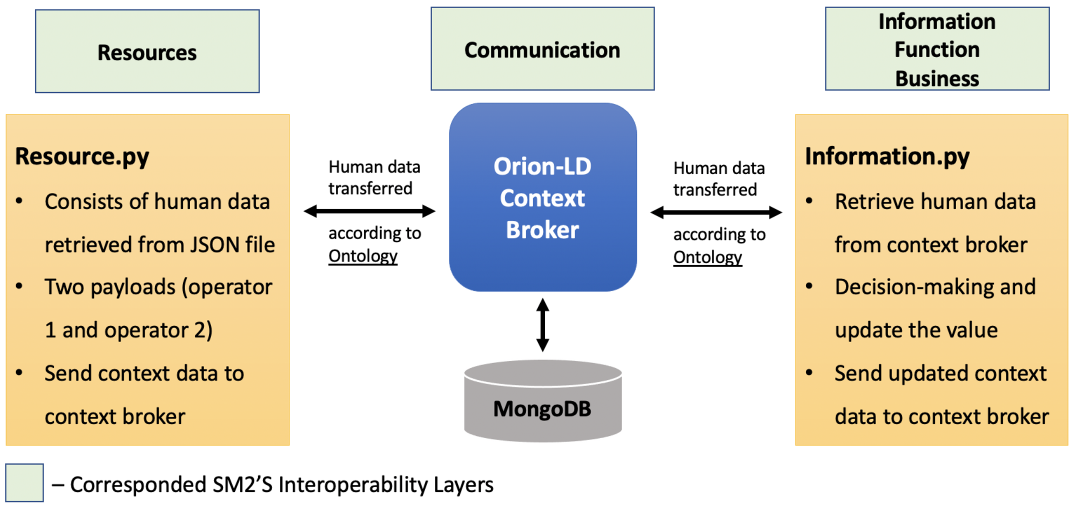

Figure 4.3.: An Overview of the Implementation

As seen in Figure 4.3, the implementation was done with two separate python codes – Resource.py and Information.py. Resource.py corresponds to SM2's resource layer, and Information.py contains actions corresponding to SM2's information, function, and business layer. At the same time, the context broker represents SM2's communication layer. It should be noted that the transition of data (specifically human data) between Resource.py, context broker, and Information.py is according to an ontology that is designed for the healthy operator and the augmented operator. Moreover, the data models that were created will also be integrated into the ontology. The ontology provides a fundamental structure for human data, therefore assisting a seamless data exchange.

Resource.py fulfills the functions of (Appendix A.1):

- Retrieving human data from JSON files that were converted from ROS bag files
- Creating an ontology that contains two payload data of the Healthy operator (operator 1) and the Augmented operator (operator 2)
- Assigning the human data from JSON files to its matching attribute
- Posting entities that contain the ontology via POST request to context broker

Information.py fulfills the functions of (Appendix A.2):

- Retrieving the entities from context broker via GET request;
- Executing decision-making. For example, determining if the operator is in danger, based on the rula risk level provided by the healthy operator;
- Updating attribute's value. For example, if the rula risk level is between 5 and 6, a "medium" alert would be updated to the attribute "severity" under the attribute "alert";
- Posting the updated entity back to context broker via POST request;
- After the updated entity is transferred back to the context broker, Resource.py can retrieve the updated entity from the context broker;

Few restrictions were encountered while creating the entity[61], such as:

- "id" field of the entity must be a Universal Resource Identifier (URI);
- "type" field of the entity is mandatory;
- Attributes of the entity must be JSON Objects;
- Every attribute must have a "type" field; the value should only consist of either Property, Relationship, or GeoProperty;
- Attribute with type Property must have a "value" field; the "value" field should not be empty or contain empty double brackets (" ");
- Attribute with type Relationship must have an "object" field, the value of "object" field must be a URI;

While these restrictions can be complicated, they also help maintain the entities' structure and avoid inconsistency of entities. Only with a response code of 201, entities can then be successfully posted on to context broker.

Examined from the perspective of SM2, the resource layer can be determined as where the human data contains and where the human data is converted into the ontology. The ontology is naturally based on the data models that are designed or suitable for every operator 4.0. Next, with the help of the communication layer, the human data according to the ontology can be retrieved from the information layer. Moreover, the function layer of SM2 will identify what the human data represents (For example, operator's action and health). Lastly, the business layer can map business models and organize services based on the identified functions.

5. Conclusion

This thesis aims to analyze smart manufacturing reference architectures precisely to determine how an operator and the operator's data are associated with each reference architecture. Moreover, to identify a suitable reference architecture to implement with human data via FIWARE. It is also essential to create an ontology; an ontology provides a basic structure for the human data generated from operator 4.0, it also assists consistent transitions between different reference architecture's layers. Methods on how ontologies and data models specifically for human data can be generated were presented and examined. Furthermore, how these human data can efficiently and safely be used on FIWARE, FIWARE is an open-source platform that was designed to assist application domains such as Smart Cities.

The integration of human data is an essential focus for smart manufacturing processes, regardless of its relation to security, efficiency, or completeness. The future plan for this thesis is to examine more operator 4.0 and human data that were generated by them, moreover, to create more data models suited to this area. These data models can hopefully support seamless and efficient integration of human data in the future.

A. Appendix

A.1. Resource.py

```
import json
import csv
import requests

risk_count = 0
risk_avr = 0
fppx = []
fppy = []
fppz = []
fpox = []
fpoy = []
fpoz = []
fpow = []

head = {"Content-Type": "application/json"}

with open("risk.json", "r+", encoding="utf-8") as file:
    risk = json.load(file)

    for key in risk:
        risk_sum = sum(key["field.data"] for key in risk)
        risk_count = risk_count + 1

risk_avr = risk_sum/risk_count

print("The␣sum␣of␣the␣risk:␣", risk_sum)
print("The␣sum␣of␣the␣risk␣count:␣", risk_count)
print("The␣average␣of␣the␣risk:␣", risk_avr)
```

```
with open("tracepen_pose.json", "r+", encoding="utf-8") as file2:
    tracepen1 = json.load(file2)

for a in range(3):
    for i in tracepen1:
      if i['field.header.seq'] == a:
        fppx.append(i['field.pose.position.x'])
        fppy.append(i['field.pose.position.y'])
        fppz.append(i['field.pose.position.z'])
        fpox.append(i['field.pose.orientation.x'])
        fpoy.append(i['field.pose.orientation.y'])
        fpoz.append(i['field.pose.orientation.z'])
        fpow.append(i['field.pose.orientation.w'])

        break

print("The␣first␣three␣field.pose.position.x:␣", fppx)
print("The␣first␣three␣field.pose.position.y:␣", fppy)
print("The␣first␣three␣field.pose.position.z:␣", fppz)
print("The␣first␣three␣field.pose.orientation.x:␣", fpox)
print("The␣first␣three␣field.pose.orientation.y:␣", fpoy)
print("The␣first␣three␣field.pose.orientation.z:␣", fpoz)
print("The␣first␣three␣field.pose.orientation.w:␣", fpow)

payload={ #This is a sample payload
  "@context": {
    "status": "http://a.b.c/attrs/status",
    "state": "http://a.b.c/attrs/state"
  },
  "id": "urn:entities:E2",
  "type": "T",
  "status": {
    "type": "Property",
    "value": "From␣Core␣Context"
  },
  "state": {
    "type": "Property",
    "value": "From␣User␣Context"
```

```
  },
  "state2": {
    "type": "Property",
    "value": "From␣Default␣URL"
  }
}


operator1={ #Healthy Operator
   "@context":{
      "operator":"https://github.com/j5j1j2j3/thesis.code-implementation/
          blob/ea1bffffe701b32f0657a7f1eec9f0763fcd3a73/operatorinfo.json",

      "device":"https://uri.fiware.org/ns/data-models#Device",
      "risklevel": "https://github.com/j5j1j2j3/thesis.code-implementation
          /blob/dd50ed4fa73e1d4830a91756e0d02abacee8bffc/risklevel.json",
      "healthlevel":"https://github.com/j5j1j2j3/thesis.code-
          implementation/blob/dd50ed4fa73e1d4830a91756e0d02abacee8bffc/
          healthlevel.json",
      "alert" : "https://uri.fiware.org/ns/data-models#Alert"
   },
   "id":"urn:entities:E1",
   "type":"healthyoperator",
   "operator":{
      "type":"Property",
      "value":"vcard",
      "firstname":{
         "type":"Property",
         "value":"Tom"
      },
      "lastname":{
         "type":"Property",
         "value":"Gomez"
      },
      "hasemail":{
         "type":"Property",
         "value":"t.gomez@gmail.com"
      },
      "gender":{
```

```
      "type":"Property",
      "value":"m"
    },
    "bday":{
      "type":"Property",
      "value":"18.05.1959"
    },
    "address":{
      "type":"Property",
      "value":"Gotthardstr.␣135"
    }
  },

  "device":{
    "type":"Property",
    "value":"sensor",
    "batteryLevel": {
      "type": "Property",
      "value": 0.75
   },
   "dateFirstUsed": {
      "type": "Property",
      "value": "20.02.2021"
   },
   "serialNumber": {
      "type": "Property",
      "value": "9845A"
   }
  },
   "risklevel":{
    "type":"Property",
    "value":"test",
    "riskFactor":{
      "type":"Property",
      "value":"undetermined"
    },
    "signOrSymptom":{
      "type":"Property",
      "value":"undetermined"
```

```
    },
    "rulaRiskLevel":{
       "type":"Property",
       "value":risk_avr
    }
 },
 "healthlevel":{
    "type":"Property",
    "value":"test",
    "heartrate":{
       "type":"Property",
       "value":"undetermined"
    },
    "bloodpressure_h":{
       "type":"Property",
       "value":"undetermined"
    },
    "bloodpressure_l":{
       "type":"Property",
       "value":"undetermined"
    },
    "bloodsugar":{
       "type":"Property",
       "value":"undetermined"
    }
 },

 "alert":{
    "type": "Property",
    "value":"alert",
    "category": {
      "type": "Property",
      "value": "operator’s␣risk"
    },
    "subCategory": {
      "type": "Property",
      "value": "operator’s␣risk␣based␣on␣rula"
    },
    "description": {
```

```
        "type": "Property",
        "value": "Risk␣level␣of␣ergonomic␣assessment,␣only␣performing␣
            ergonomic␣evaluation␣when␣two␣views␣are␣available␣and␣main␣
            keypoints␣are␣well␣detected.␣Including␣specific␣descriptions:␣
            risk␣factor,␣signs␣or␣symptoms␣and␣rula␣risk␣level.."
      },
      "location": {
        "type": "Property",
        "value": "undetermined"
      },
      "alertSource": {
        "type": "Property",
        "value": "undetermined"
      },
    "severity": {
        "type": "Property",
        "value": "undetermined"
    }
   }
}


operator2={ #Augmented Operator
  "@context": {
    "tracepen": "https://github.com/j5j1j2j3/thesis.code-implementation/
        blob/9bf76dde52cfdbe585c6cc2f980499fd11e213e0/tracepenpose.json"
  },
  "id": "urn:entities:E2",
  "type": "augmentedoperator",
  "tracepen": {
    "type": "Property",
    "value": "tracepen",
    "posePosition": {
      "type": "Property",
      "value": [fppx[1],fppy[1],fppz[1]]
    },
    "poseOrientation" : {
      "type": "Property",
      "value": [fpox[1], fpoy[1], fpoz[1], fpow[1]]
```

```
    }
  }
}

'''
operator2={ #This is the test version of operator2
  "@context": {
    "craneoperator": "http://a.b.c/attrs/status",
    "device": "http://a.b.c/attrs/status",
    "measurement": "http://a.b.c/attrs/status",
    "position": "http://a.b.c/attrs/status"
   },
  "id": "urn:entities:E7",
  "type": "augmentedoperator",
  "craneoperator": {
    "type": "Property",
    "value": "vcard",
    "firstname": {
      "type": "Property",
      "value": "Tom"},
    "lastname": {
      "type": "Property",
      "value": "Gomez"},
    "hasemail": {
      "type": "Property",
      "value": "t.gomez@gmail.com"},
    "gender": {
      "type": "Property",
      "value": "m"},
    "bday": {
      "type": "Property",
      "value": "18.05.1959"},
    "address": {
      "type": "Property",
      "value": "Gotthardstr. 135"}
   },
   "device": {
    "type": "Property",
    "value": "",
```

```
    "deviceID": {
      "type": "Property",
      "value": ""},
    "producedyear": {
      "type": "Property",
      "value": ""}
   },
  "measurement": {
    "type": "Property",
    "value": "",
    "distance": {
      "type": "Property",
      "value": "",
      "target": {
       "type": "Property",
       "value": "" },
      "indepth": {
       "type": "Property",
       "value": "" }
   }
  },
   "position": {
    "type": "Property",
    "value": "",
    "tracepen_pose": {
      "type": "Property",
      "value": "",
      "position.x": {
       "type": "Property",
       "value": "" },
      "position.y": {
       "type": "Property",
       "value": "" },
      "position.z": {
       "type": "Property",
       "value": "" },
      "orientation.x": {
       "type": "Property",
       "value": "" },
```

```
      "orientation.y": {
       "type": "Property",
       "value": "" },
      "orientation.z": {
       "type": "Property",
       "value": "" },
      "orientation.w": {
       "type": "Property",
       "value": "" },
      "timestamp": {
       "type": "Property",
       "value": "" }
   }
  }
}
'''

testdata=json.dumps(operator1)
testdata2=json.dumps(operator2)
testdata3=json.dumps(payload)

response = requests.post(url='http://localhost:1026/ngsi-ld/v1/entities',
     headers={
     "content-type": "application/ld+json"}, data=testdata)

response2 = requests.post(url='http://localhost:1026/ngsi-ld/v1/entities',
     headers={
    "content-type": "application/ld+json"}, data=testdata2)

print(response.status_code)

print(response2.status_code)

#url='http://localhost:1026/ngsi-ld/v1/entities?id=urn:entities:E6'
#response3 = requests.get(url)
#print(response3.json())

#print(testdata)
```

A.2. Information.py

```
import requests, json

url='http://localhost:1026/ngsi-ld/v1/entities?id=urn:entities:E1'
response = requests.get(url)
#response_json = response.json()

json_data = json.loads(response.text)
for r in json_data:
      risk_value = r["https://github.com/j5j1j2j3/thesis.code-
          implementation/blob/dd50ed4fa73e1d4830a91756e0d02abacee8bffc/
          risklevel.json"]["rulaRiskLevel"]["value"]
      if risk_value >= 1.0 and risk_value <= 2.0:
            r["https://uri.fiware.org/ns/data-models#Alert"]["severity"][
                "value"] = "informational"
      if risk_value >= 3.0 and risk_value <= 4.0:
            r["https://uri.fiware.org/ns/data-models#Alert"]["severity"][
                "value"] = "low"
      if risk_value >= 5.0 and risk_value <= 6.0:
            r["https://uri.fiware.org/ns/data-models#Alert"]["severity"][
                "value"] = "medium"
      if risk_value >= 6.0:
            r["https://uri.fiware.org/ns/data-models#Alert"]["severity"][
                "value"] = "high"


response2 = requests.post(url='http://localhost:1026/ngsi-ld/v1/
    entityOperations/update', headers={
      "content-type": "application/ld+json"}, data=json.dumps(json_data))

print(risk_value)
#print(json_data)
print(response2.status_code)

response = requests.get(url)
json_data = json.loads(response.text)
print(json_data)
```

List of Figures

Bibliography

[1] M. Hankel and B. Rexroth. "Das Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0)." In: *Zentralverband Elektrotechnik-und Elektronikindustrie e. V. URL: https://www. zvei. org/fileadmin/user_upload/Presse_ und_Medien/Publikationen/2015/april/Das_Referenzarchitekturmodell_Industrie_4. 0__RAMI_4. 0_/Faktenblatt-Industrie4_0-RAMI-4_0. pdf* (2015).

[2] *IDS Reference Architecture Model 3.0*. [Online; accessed 28-Dec-2020]. International Data Spaces Assosiation, 2021.

[3] S.-W. Lin, B. Miller, J. Durand, R. Joshi, P. Didier, A. Chigani, R. Torenbeek, D. Duggal, R. Martin, G. Bleakley, et al. "Industrial internet reference architecture." In: *Industrial Internet Consortium (IIC), Tech. Rep* (2015).

[4] J. Lee, B. Bagheri, and H.-A. Kao. "A cyber-physical systems architecture for industry 4.0-based manufacturing systems." In: *Manufacturing letters* 3 (2015), pp. 18–23.

[5] M. Broy. *Cyber-physical systems: Innovation durch softwareintensive eingebettete Systeme*. Springer-Verlag, 2011.

[6] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic. "Cyber-physical systems: the next computing revolution." In: *Design automation conference*. IEEE. 2010, pp. 731–736.

[7] F. Wortmann and K. Flüchter. "Internet of things." In: *Business & Information Systems Engineering* 57.3 (2015), pp. 221–224.

[8] J. Holler, V. Tsiatsis, C. Mulligan, S. Karnouskos, S. Avesand, and D. Boyle. *Internet of Things*. Academic Press, 2014.

[9] S. Forsstroem and U. Jennehag. "A performance and cost evaluation of combining OPC-UA and Microsoft Azure IoT Hub into an industrial Internet-of-Things system." In: *2017 Global Internet of Things Summit (GIoTS)*. IEEE. 2017, pp. 1–6.

[10] B. Marr. *How BMW Uses Artificial Intelligence And Big Data To Design And Build Cars Of Tomorrow*. `https://www.forbes.com/sites/bernardmarr/2017/08/01/how-bmw-uses-artificial-intelligence-and-big-data-to-design-and-build-cars-of-tomorrow/?sh=358da8a32b91`. [Online; accessed 28-Dec-2020]. 2021.

[11] S. J. Qin and L. H. Chiang. "Advances and opportunities in machine learning for process data analytics." In: *Computers & Chemical Engineering* 126 (2019), pp. 465–473.

[12] J. Davis, T. Edgar, J. Porter, J. Bernaden, and M. Sarli. "Smart manufacturing, manufacturing intelligence and demand-dynamic performance." In: *Computers & Chemical Engineering* 47 (2012), pp. 145–156.

[13] K. Nagorny, P. Lima-Monteiro, J. Barata, and A. W. Colombo. "Big data analysis in smart manufacturing: A review." In: *International Journal of Communications, Network and System Sciences* 10.3 (2017), pp. 31–58.

[14] J. Schlick. "Cyber-physical systems in factory automation-Towards the 4th industrial revolution." In: *2012 9th IEEE International Workshop on Factory Communication Systems*. IEEE. 2012, pp. 55–55.

[15] F. Ansari, P. Hold, and W. Sihn. "Human-Centered Cyber Physical Production System: How Does Industry 4.0 impact on Decision-Making Tasks?" In: *2018 IEEE Technology and Engineering Management Conference (TEMSCON)*. IEEE. 2018, pp. 1–6.

[16] A. Ustundag and E. Cevikcan. *Industry 4.0: managing the digital transformation*. Springer, 2017.

[17] X. Xu. "From cloud computing to cloud manufacturing." In: *Robotics and computer-integrated manufacturing* 28.1 (2012), pp. 75–86.

[18] B. Huang, C. Li, C. Yin, and X. Zhao. "Cloud manufacturing service platform for small-and medium-sized enterprises." In: *The International Journal of Advanced Manufacturing Technology* 65.9-12 (2013), pp. 1261–1272.

[19] D. Romero, P. Bernus, O. Noran, J. Stahre, and A. Fast-Berglund. "The operator 4.0: human cyber-physical systems and adaptive automation towards human-automation symbiosis work systems." In: *IFIP international conference on advances in production management systems*. Springer. 2016, pp. 677–686.

[20] D. Romero, J. Stahre, T. Wuest, O. Noran, P. Bernus, A. Fasth Fast-Berglund, and D. Gorecky. "Towards an Operator 4.0 Typology: A Human-Centric Perspective on the Fourth Industrial Revolution Technologies." In: Oct. 2016.

[21] J. Ortiz, S. Toxiri, and D. G. Caldwell. "Beyond Robo-Mate: Towards the Next Generation of Industrial Exoskeletons in Europe." In: *International Symposium on Wearable Robotics*. Springer. 2018, pp. 365–369.

[22] *Augmented Reality*. `https://www.illogic.xyz/technologies/augmented-reality/`. [Online; accessed 02-Jan-2021]. 2021.

[23] *Satisfactory (a collaborative and augmented-enabled ecosystem for increasing satisfaction and working experience in smart factory environments)*. `http://www.satisfactory-project.eu/`. [Online; accessed 02-Jan-2021]. 2021.

[24] J. Zheng, K. Chan, and I. Gibson. "Virtual reality." In: *Ieee Potentials* 17.2 (1998), pp. 20–23.

[25] N. Ordaz, D. Romero, D. Gorecky, and H. R. Siller. "Serious games and virtual simulator for automotive manufacturing education and training." In: *Procedia Computer Science* 75 (2015), pp. 267–274.

[26] P. Maurice, V. Padois, Y. Measson, and P. Bidaud. "Human-oriented design of collaborative robots." In: *International Journal of Industrial Ergonomics* 57 (2017), pp. 88–102.

[27] K. Darvish, F. Wanderlingh, B. Bruno, E. Simetti, F. Mastrogiovanni, and G. Casalino. "Flexible human–robot cooperation models for assisted shop-floor tasks." In: *Mechatronics* 51 (2018), pp. 97–114.

[28] K. Myers, P. Berry, J. Blythe, K. Conley, M. Gervasio, D. L. McGuinness, D. Morley, A. Pfeffer, M. Pollack, and M. Tambe. "An intelligent personal assistant for task and time management." In: *AI Magazine* 28.2 (2007), pp. 47–47.

[29] Q. P. He and J. Wang. "Statistical process monitoring as a big data analytics tool for smart manufacturing." In: *Journal of Process Control* 67 (2018), pp. 35–43.

[30] T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben. "Machine learning in manufacturing: advantages, challenges, and applications." In: *Production and Manufacturing Research* 4.1 (2016), pp. 23–45.

[31] T. C. Ford, J. M. Colombi, S. R. Graham, and D. R. Jacques. *Survey on interoperability measurement*. Tech. rep. AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH, 2007.

[32] E. Morris, L. Levine, C. Meyers, P. Place, and D. Plakosh. "System of Systems Interoperability (SOSI): final report." In: *Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA* (2004).

[33] M. Kasunic. *Measuring systems interoperability: Challenges and opportunities*. Tech. rep. Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 2001.

[34] A. Zeid, S. Sundaram, M. Moghaddam, S. Kamarthi, and T. Marion. "Interoperability in smart manufacturing: Research challenges." In: *Machines* 7.2 (2019), p. 21.

[35] S. J. Kemmerer and S. J. Kemmerer. *Manufacturing Interoperability Program, a Synopsis*. US Department of Commerce, National Institute of Standards and Technology, 2009.

[36] B. Solomon, D. Ionescu, M. Litoiu, and M. Mihaescu. "Towards a real-time reference architecture for autonomic systems." In: *International Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS'07)*. IEEE. 2007, pp. 10–10.

[37] N. Bui. *An analysis of the benefits of EULYNX-style requirements modeling for ProRail*. 2017.

[38] M. Moghaddam, M. N. Cadavid, C. R. Kenley, and A. V. Deshmukh. "Reference architectures for smart manufacturing: A critical review." en. In: *Journal of Manufacturing Systems* 49 (Oct. 2018), pp. 215–225. ISSN: 02786125.

[39] 2021.

[40] *INDUSTRIAL INTERNET CONSORTIUM*. 2021.

[41] Y. Bai. "Industrial Internet of things over tactile Internet in the context of intelligent manufacturing." In: *Cluster Computing* 21.1 (2018), pp. 869–877.

[42] "DIN SPEC 91345. Reference Architecture Model Industrie 4.0 (RAMI4.0)." In: (2016).

[43] S. Han. "A review of smart manufacturing reference models based on the skeleton meta-model." In: *Journal of Computational Design and Engineering* 7.3 (2020), pp. 323–336.

[44] *RAMI 4.0 – Ein Orientierungsrahmen für die Digitalisierung*. 2021.

[45] W. J. O'Brien. "Fiatech: the next generation of the capital projects technology roadmap." In: *Journal of Construction Engineering and Management* 143.9 (2017), p. 02517003.

[46] S. Wei, J. Hu, Y. Cheng, Y. Ma, and Y. Yu. "The essential elements of intelligent manufacturing system architecture." In: *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*. IEEE. 2017, pp. 1006–1011.

[47] I. IVCI. "Industrial Value Chain Reference Architecture." In: *Hannover, Germany* (2017).

[48] *FIWARE Webinars - Academy*. `https://fiware-academy.readthedocs.io/en/latest/integrated-courses/webinars/index.html`. [Online; accessed 08-Jan-2021]. 2021.

[49] F. Cirillo, G. Solmaz, E. L. Berz, M. Bauer, B. Cheng, and E. Kovacs. "A standard-based open source IoT platform: FIWARE." In: *IEEE Internet of Things Magazine* 2.3 (2019), pp. 12–18.

[50] *Step-by-Step for NGSI-V2*. `https://fiware-tutorials.readthedocs.io/en/latest/iot-agent/index.html`. [Online; accessed 01-March-2021]. 2021.

[51] *Data Models*. https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Data+Models. [Online; accessed 15-April-2021]. 2021.

[52] *FIWARE Data Models*. https://github.com/FIWARE/data-models. [Online; accessed 01-March-2021]. 2021.

[53] *dataModel.Alert*. https://github.com/smart-data-models/dataModel.Alert. [Online; accessed 01-March-2021]. 2021.

[54] *List of data models and repositories*. https://smartdatamodels.org/index.php/list-of-data-models/. [Online; accessed 15-April-2021]. 2021.

[55] *Create data model from your google spreadsheet*. https://smartdatamodels.org/index.php/create-data-model-from-your-google-spreadsheet/. [Online; accessed 15-April-2021]. 2021.

[56] *Wie funktionieren Docker Container?* https://www.claranet.de/blog/wie-funktionieren-docker-container. [Online; accessed 13-April-2021]. 2021.

[57] *How to use Orion Context Broker with Docker*. https://fiware-orion.readthedocs.io/en/1.4.1/user/docker/index.html. [Online; accessed 22-Feb-2021]. 2021.

[58] V. Code. *Working with MongoDB in Visual Studio Code*. 2021.

[59] *FIWARE Getting Started*. https://documenter.getpostman.com/view/513743/fiware-getting-started/RVu5kp1c. [Online; accessed 22-Feb-2021]. 2021.

[60] *FIWARE-NGSI v2 Specification*. http://telefonicaid.github.io/fiware-orion/api/v2/stable/. [Online; accessed 22-Feb-2021]. 2021.

[61] *Introduction to NGSI-LD Entities and Attributes*. https://github.com/FIWARE/context.Orion-LD/blob/develop/doc/manuals-ld/entities-and-attributes.md. [Online; accessed 22-Feb-2021]. 2021.

License