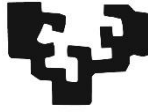


eman ta zabal zazu



Universidad del País Vasco Euskal Herriko Unibertsitatea

Department of Automatic Control and Systems Engineering

University of the Basque Country (UPV/EHU)

INDUSTRIAL AGENTS FOR RESILIENT MANUFACTURING SYSTEMS

AN I4.0 PLATFORM FOR THE MANUFACTURING DOMAIN

Phd Dissertation

Alejandro López García

Supervisors

Elisabet Estévez Estévez

Oskar Casquero Oyarzabal

Bilbao, February 2023

ACKNOWLEDGEMENT

I would like to thank my thesis directors, Marga Marcos, Elisabet Estévez and Oskar Casquero, for their patience and support, and for helping me through the first steps of my research career. I would also like to thank Professor Paulo Leitão for his kindness and dedication, and for giving me the opportunity to carry out a predoctoral stay under his supervision.

I also want to express my gratitude to all the members of the Department of Automatic Control and Systems Engineering for their support and advice during these years: Darío, Isabel, Aintzane, Nagore, Fede, Itziar, Atziber.... Thank you all very much. Nor do I want to forget all my lab colleagues, both in Bilbao and Bragança, for being an inexhaustible source of encouragement and good times: Patrick, Asier Brull and Asier Arizala, Sergio, Janire, Nerea, Lucas, Guido, Jonas, Kathleen, Raul and Regina, thank you.

I would especially like to thank all the people who have contributed with their work to shape the ideas that have emerged from this doctoral thesis: Diego, Ekaitz, Julen, Ander, Borja, Alazne, Camilo, Jon, Adrián, Ane, it has been a pleasure to work with all of you.

Finally, I would like to thank my family, my parents and my sister, who have shown infinite patience and understanding, especially in the most complicated moments. And of course, to my fiancée, Inés. Without you, this would not have been possible.

Thank you all.

This PhD was financed by Ministerio de Ciencia, Innovación y Universidades (MCIU)/Agencia Estatal de Investigación (AEI)/Fondo Europeo de Desarrollo Regional (FEDER), Unión Europea (UE), under grant PRE2019-088736.

The development of this PhD has been framed within the following projects:

Arquitectura inteligente para la industria 4.0: realimentación a través del fog (REF RTI2018-096116-B-I00), financed by Ministerio de Ciencia, Innovación y Universidades (MCIU), Agencia Estatal de Investigación (AEI), Fondo Europeo de Desarrollo Regional (FEDER), Unión Europea (UE), DESDE: 01/01/2019 HASTA: 30/09/2022.

Tecnologías base para la digitalización industrial (Elkartek 19/23 REF KK-2019-00095), financed by Gobierno Vasco (EJ/GV), FROM: 01/03/2019 TO: 31/12/2020.

Ayudas para apoyar las actividades de grupos de investigación del sistema universitario vasco (REF IT1324-19), financed by Gobierno Vasco (EJ/GV), FROM: 01/01/2019 TO: 31/12/2021.

ABSTRACT

The industry is undergoing a profound transformation caused by economic uncertainty and highly competitive markets. This translates into increased customer demands in terms of quality, cost and customization. The need to meet these demands has been the cause of a cascade of technological advances that multiple institutions have agreed to identify as the fourth industrial revolution.

In the case of the manufacturing domain, different institutions have emerged during the last decade in order to present their strategies to face the challenges of Industry 4.0. The most relevant outcome of these institutions are their reference architectures. These aim to identify the concepts and technologies to be used as a basis for the design and development of Industry 4.0-compliant solutions. However, the terms in which they are presented are generic and neutral from a technological point of view, without offering support for the practical development of solutions, which is a handicap to their adoption by companies and professionals.

In this context, this work proposes a platform for the management of manufacturing systems in the context of Industry 4.0. For this purpose, this platform makes use of the concepts and standards contained in the *Reference Architectural Model for Industrie 4.0 (RAMI 4.0)*, considered the reference architecture at European and global level. On this basis, the platform provides technological and methodological support to facilitate tasks such as asset characterization and integration through the development of the *Asset Administration Shell (AAS)*, the entity proposed in *RAMI 4.0* to provide interoperability and service management. In addition, the core of the I4.0 Platform implements the Infrastructure Services contemplated by *RAMI 4.0* for the creation, registration and exposure of AASs in the system, and extends them with additional functionalities to guarantee the operational status of the factory in the event of process incidents or unplanned events.

This I4.0 Platform has been designed for *FIPA* compliant industrial players. Industrial agents have an inherent ability to make decisions autonomously, allowing them to decide to compete and/or collaborate with each other to achieve their goals. In addition, the Agent Communication Language (*ACL*) communication protocol promoted by *FIPA* incorporates mechanisms such as the use of ontologies and performatives that facilitate the contextualization of communications. The combination of these features makes industrial agents ideal for the implementation of Industry 4.0 solutions.

This I4.0 Platform has been tested by means of three test scenarios defined to evaluate its operation under normal conditions, and its response in the event of failures affecting to physical and logical assets. These scenarios have been designed from a generic approach and then customized for the demonstrator used, which recreates a manufacturing environment using industrial equipment.

LABURPENA

Industria ziurgabetasun ekonomikoak eta merkatuen lehiakortasun handiak eraginda aldaketa sakonean murgilduta dago. Honen ondorioz, bezeroaren eskakizunak areagotu egiten dira kalitatean, kostuan eta pertsonalizazioan. Eskaera hauei erantzuteko beharra erakunde askok laugarren industria-iraultza gisa identifikatzea adostu dutela aurrerapen teknologikoen uholdearen arrazoia izan da.

Fabrikazio-domeinuaren kasuan, gobernuko edo ekimen pribatuko hainbat erakunde sortu dira azken hamarkadan, Industria 4.0aren erronkei aurre egiteko estrategiak aurkezteko. Erakunde hauen emaitzarik garrantzitsuena erreferentzia-arkitekturak dira. Soluzioen diseinuan eta garapenean erabili beharreko kontzeptuak eta teknologiak identifikatzea dute helburu, Industria 4.0aren arabera. Hala ere, erreferentzia-arkitektura hauek aurkezteko erabiltzen diren terminoak generikoak eta neutralak dira teknologiaren ikuspegitik, eta ez dute soluzioen garapen praktikorako euskarririk eskaintzen. Hau galga bat da enpresek eta profesionalek erreferentzia-arkitektura hauek har ditzaten.

Testuinguru honetan, lan honek fabrikazio-sistemak kudeatzeko plataforma bat proposatzen du, Industria 4.0aren esparruan kokatua. Horretarako, plataforma honek *Reference Architectural Model for Industrie 4.0 (RAMI 4.0)* erreferentzia-arkitekturan jasotako kontzeptuak eta arauak erabiltzen ditu, Europan eta mundu osoan garrantzitsuena dena. Oinarri honen gainean, I4.0 Plataformak euskarri teknologiko eta metodologikoa eskaintzen du, hainbat zeregin errazteko, hala nola aktiboen karakterizazioa eta *Asset Administration Shell (AAS)* integrazioa, *RAMI 4.0*n elkarrengarritasuna eta zerbitzuen kudeaketa eskaintzeko proposatutako entitatea. Gainera, I4.0 Plataformaren nukleoak *RAMI 4.0*k aurreikusitako *Infrastructure Services* zerbitzuak sisteman *AAS*ak sortzeko, erregistratzeko eta erakusteko ezartzen ditu, eta funtzionalitate gehigarriekin zabaltzen ditu, fabrikaren egoera operatiboa bermatzeko, prozesuan izandako gorabeheren edo planifikatu gabeko gertaeren aurrean.

I4.0 Plataforma hau *FIPA* estandarrarekin bateragarriak diren industria-agenteentzat diseinatu da. Industria-agenteek erabakiak modu autonomoan hartzeko berezko gaitasuna dute, eta, horri esker, beren helburuak lortzeko elkarrekin lehiatzea eta/edo lankidetzan aritzea erabaki dezakete. Gainera, *FIPA* estandarrak sustatzen duen *Agent Communication Language (ACL)* komunikazio-protokoloak ontologiaren eta performatiboen erabilera bezalako mekanismoak barneratzen ditu, komunikazioei testuingurua ematen laguntzen dutenak. Ezaugarri hauek konbinatuta, industria-agenteak ezin hobeak dira Industria 4.0rako soluzioak ezartzeko.

I4.0 Plataforma hau hiru proba-agertokiren bidez probatu da. Proba-agertoki hauek, batetik, plataformaren funtzionamendua baldintza normaletan ebaluatzeko, eta, bestetik, aktibo fisiko eta logikoei eragiten dieten akatsen kasuan duen erantzuna ebaluatzeko definituta izan dira. Agertoki hauek ikuspegi orokor batetik diseinatu dira, eta, ondoren, erabilitako saiakuntza-plataformarako bereizi dira, industria-ekipamenduaren bidez fabrikazio-ingurune bat birsortzen baitu.

INDEX

TABLE OF CONTENTS

1 INTRODUCTION

| | | |
|-----|------------------|-----|
| 1.1 | MOTIVATION..... | 1-1 |
| 1.2 | OBJECTIVES | 1-3 |
| 1.3 | STRUCTURE | 1-4 |

2 STATE OF THE ART

| | | |
|-------|--|------|
| 2.1 | INTRODUCTION | 2-1 |
| 2.2 | REFERENCE ARCHITECTURES FOR INDUSTRY 4.0..... | 2-2 |
| 2.2.1 | <i>Reference Architectural Model for Industrie 4.0</i> | 2-2 |
| 2.2.2 | <i>Industrial Internet Reference Architecture</i> | 2-8 |
| 2.2.3 | <i>Intelligent Manufacturing System Architecture</i> | 2-10 |
| 2.2.4 | <i>Industrial Value Chain Reference Architecture</i> | 2-12 |
| 2.2.5 | <i>Alignment and synergies</i> | 2-15 |
| 2.2.6 | <i>Discussion</i> | 2-17 |
| 2.3 | INTEGRATION OF ASSETS IN INDUSTRY 4.0 | 2-18 |
| 2.3.1 | <i>Related work</i> | 2-19 |
| 2.3.2 | <i>Discussion</i> | 2-23 |
| 2.4 | I4.0 SYSTEM MANAGEMENT PLATFORMS..... | 2-24 |
| 2.4.1 | <i>Related work</i> | 2-25 |
| 2.4.2 | <i>Discussion</i> | 2-30 |
| 2.5 | SUMMARY AND CONCLUSIONS | 2-32 |

3 INDUSTRIA 4.0RAKO EUSKARRI METODOLOGIKOA

| | | |
|-------|--|------|
| 3.1 | SARRERA | 3-1 |
| 3.2 | AKTIBOEN KARAKTERIZAZIOA | 3-1 |
| 3.2.1 | <i>Hierarchy Level ardatzaren estandarrak: IEC 62264 eta IEC 61512</i> | 3-2 |
| 3.2.2 | <i>Eredu-egite ikuspegia</i> | 3-7 |
| 3.3 | AKTIBOEN INTEGRAZIOA | 3-11 |
| 3.3.1 | <i>Geruza anitzeko ikuspegia</i> | 3-11 |
| 3.3.2 | <i>Integrazio-metodologia</i> | 3-16 |

| | | |
|----------|---|------|
| 3.4 | AAS GARAPENA..... | 3-18 |
| 3.4.1 | <i>Skeleton pattern txantiloia</i> | 3-20 |
| 3.4.2 | <i>Design pattern txantiloia</i> | 3-21 |
| 3.5 | LABURPENA ETA ONDORIOAK..... | 3-22 |
| 4 | I4.0 PLATAFORMAREN DISEINUA ETA GARAPENA | |
| 4.1 | SARRERA..... | 4-1 |
| 4.2 | IARMS I4.0 PLATAFORMAREN IKUSPEGI OROKORRA..... | 4-2 |
| 4.3 | AAS GARATZEKO SKELETON PATTERN TXANTILOIAK | 4-7 |
| 4.3.1 | <i>RA skeleton pattern txantiloiak</i> | 4-7 |
| 4.3.2 | <i>AppA skeleton pattern txantiloiak</i> | 4-14 |
| 4.4 | I4.0 COMPONENTS OSAGAIEN KUDEAKETA ERRESILIENTEA..... | 4-18 |
| 4.4.1 | <i>QoSA agentea: gertaeren detekzioa</i> | 4-19 |
| 4.4.2 | <i>D&DA agentea: gertaeren ebazpena</i> | 4-23 |
| 4.5 | LABURPENA ETA ONDORIOAK..... | 4-24 |
| 5 | KONZEPTU-PROBA | |
| 5.1 | SARRERA..... | 5-1 |
| 5.2 | SAIAKUNTZA-PLATAFORMA | 5-1 |
| 5.2.1 | <i>Saiakuntza-plataformaren ikuspegi orokorra</i> | 5-1 |
| 5.2.2 | <i>Fabrikazio-zelulen integrazioa</i> | 5-3 |
| 5.3 | PROBA-AGERTOKIAK DEFINIZIOA..... | 5-10 |
| 5.3.1 | <i>Proba-agertoki eta KPI orokorrak</i> | 5-10 |
| 5.3.2 | <i>Proba-agertokiak eta KPI pertsonalizatuak</i> | 5-11 |
| 5.4 | EMAITZAK ETA EZTABAIDA | 5-16 |
| 5.4.1 | <i>Hedapen-proba</i> | 5-16 |
| 5.4.2 | <i>Makinaren akats-proba</i> | 5-19 |
| 5.4.3 | <i>Prozesatze-nodoaren akats-proba</i> | 5-20 |
| 5.5 | LABURPENA ETA ONDORIOAK..... | 5-24 |
| 6 | CONCLUSIONS AND FUTURE WORK | |
| 6.1 | CONCLUSIONS | 6-1 |
| 6.2 | FUTURE WORKS..... | 6-5 |

REFERENCES

GLOSSARY

TABLE OF FIGURES

Figure 2-1 Cubic model proposed by RAMI 4.0. Adapted from (Adolphs and Epple, 2015, Figure 1)..... 2-3

Figure 2-2 Elements of the I4.0 Component positioned with respect to the RAMI 4.0 layers. Adaptation of (Schweichhart, 2016). 2-6

Figure 2-3 Interaction types of AASs. Adapted from (Miny et al., 2021, Figure 14)..... 2-6

Figure 2-4 Viewpoints proposed in IIRA. Adapted from (Lin et al., 2019, Figure 3-4)..... 2-9

Figure 2-5 Cubic model proposed by IMSA. Adapted from (Wei et al., 2017, Figure 1)..... 2-10

Figure 2-6 Model for advanced manufacturing proposed by IVRA. Adapted from (“Industrial Value Chain Reference Architecture (IVRA),” 2016, Figure 2)..2-13

Figure 2-7 View of the SMU proposed by IVRA. Adapted from (“Industrial Value Chain Reference Architecture (IVRA),” 2016, Figure 1)..... 2-13

Figure 2-8 Stages in the Industry 4.0 maturity index. Adapted from (Schuh et al., 2020a, Figure 6)..... 2-18

Irudia 3-1 Role based equipment model eredua. Egokitzapena (“IEC 62264-2,” 2013, Figure 6). 3-3

Irudia 3-2 Material model eredua. Egokitzapena (“IEC 62264-2,” 2013, Figure 9)..... 3-3

Irudia 3-3 Process model eredua. Egokitzapena (“IEC 61512-1,” 1997, Figure 1). 3-5

Irudia 3-4 Operations definition model eredua. Egokitzapena (“IEC 62264-2,” 2013, Figure 13). 3-5

| | |
|--|------|
| Irudia 3-5 Operations definition model eredua. Egokitzapena (“IEC 62264-2,” 2013, Figure 14). | 3-6 |
| Irudia 3-6 Manufacturing system meta-eredua (XMLSpyrekin sortua). | 3-7 |
| Irudia 3-7 Equipment meta-eredua (XMLSpyrekin sortua). | 3-8 |
| Irudia 3-8 Manufacturing plan meta-eredua (XMLSpyrekin sortua). | 3-9 |
| Irudia 3-9 Recipe track meta-eredua (XMLSpyrekin sortua). | 3-10 |
| Irudia 3-10 Updated plan meta-eredua (XMLSpyrekin sortua). | 3-10 |
| Irudia 3-11 Aktibo fisikoak Industria 4.0n integratzeko geruza anitzeko ikuspegia. | 3-12 |
| Irudia 3-12 I4.0 Component osagaien artean zerbitzu-eskaeretan sartu beharreko informazioa. | 3-14 |
| Irudia 3-13 Zerbitzu-eskaera Operative Layer geruzara bidaltzeko behar den informazioa. | 3-14 |
| Irudia 3-14 AASak bi etapatan garatzeko proposamena: skeleton pattern eta design pattern txantiloak. | 3-19 |
| Irudia 3-15 Skeleton pattern txantiloaren egoera finituen makina. | 3-21 |
| Irudia 4-1 IARMS I4.0 Plataformaren ikuspegi orokorra. Egokitzapena (Gangoiti et al., 2022, Fig. 1). SRA = System Repository Agent, SR = System Repository, PNA = Processing Node Agent, PLA = Planner Agent, QoSA = Quality of Service Agent, D&DA = Diagnosis and Decision Agent, MA = Machine Agent, TA = Transport Agent, BA = Batch Agent, OA = Order Agent, MPA = Manufacturing Plan Agent. | 4-4 |
| Irudia 4-2 IARMS I4.0 Plataformako RA skeleton pattern txantiloietarako egoera finituen makina. | 4-8 |
| Irudia 4-3 MA skeleton pattern txantiloaren interakzioak boot egoeran adierazten dituen sekuentzia-diagrama. | 4-10 |

| | |
|--|-------------|
| Irudia 4-4 AASen eta GA agenteen arteko interakzioak estandarizatzeko erabiltzen diren sendDataToAsset eta rcvDataFromAsset metodoen funtzionamendua adierazten duen sekuentzia-diagrama..... | 4-12 |
| Irudia 4-5 Geruza anitzeko ikuspegiaren aplikazioa tightly coupled, hybrid praktikarako. Egokitzapena (Leitão et al., 2018, Figure 5)..... | 4-13 |
| Irudia 4-6 Geruza anitzeko ikuspegiaren aplikazioa tightly coupled, on device praktikarako. Egokitzapena (Leitão et al., 2018, Figure 6)..... | 4-14 |
| Irudia 4-7 Geruza anitzeko ikuspegiaren aplikazioa loosely coupled, hybrid praktikarako. Egokitzapena (Leitão et al., 2018, Figure 8)..... | 4-14 |
| Irudia 4-8 Geruza anitzeko ikuspegiaren aplikazioa loosely coupled, on device praktikarako. Egokitzapena (Leitão et al., 2018, Figure 9)..... | 4-14 |
| Irudia 4-9 IARMS I4.0 Plataformaren AppA skeleton pattern txantiloietarako egoera finituen makina..... | 4-15 |
| Irudia 4-10 MPA, OA eta BA skeleton pattern txantiloien interakzioak boot egoeran adierazten dituen sekuentzia-diagrama. | 4-17 |
| Irudia 4-11 MPA, OA eta BA skeleton pattern txantiloien interakzioak running egoeran adierazten dituen sekuentzia-diagrama. | 4-18 |
| Irudia 4-12 QoSA agenteak denbora-muga gainditze motako gertaera nola egiaztatzen duen adierazten duen sekuentzia-diagrama. | 4-20 |
| Irudia 4-13 QoSA agenteak mezu-galera motako gertaera nola egiaztatzen duen adierazten duen sekuentzia-diagrama..... | 4-22 |
| Irudia 5-1 IARMS I4.0 Plataformaren hedapenaren xehetasuna saiakuntza-plataforman. | 5-2 |
| Irudia 5-2 Xehetasuna: a) fabrikazio-zelula; b) produktuaren zatiak; c) ondoriozko I4.0 Component osagaia..... | 5-4 |
| Irudia 5-3 Functional Layer geruza ezartzen duen robotaren azpiprogrametako baten sasikodea. | 5-7 |

| | |
|--|-------------|
| Irudia 5-4 AASaren ikuspegi orokorra (AASX Package Explorer softwarearekin sortuta)..... | 5-9 |
| Irudia 5-5 Hedapen-proban MA agenteetan oinarritutako AASetarako bildutako denbora-markak adierazten dituen denbora-diagrama..... | 5-13 |
| Irudia 5-6 Hedapen-proban MPA agenteentzat bildutako denbora-markak adierazten dituen denbora-diagrama..... | 5-13 |
| Irudia 5-7 Makinaren akats-proban jasotako denbora-markak adierazten dituen denbora-diagrama..... | 5-14 |
| Irudia 5-8 Prozesatze-nodoaren akats-proban jasotako denbora-markak adierazten dituen denbora-diagrama..... | 5-16 |
| Irudia 5-9 MA agenteetan oinarritutako AASetarako hedapen-denbora bereiziak. | 5-17 |
| Irudia 5-10 AppA agenteetan oinarritutako AASentzako hedapen-denbora bereiziak. | 5-18 |
| Irudia 5-11 Makinaren akatsa berreskuratzeko denbora bereiziak. | 5-20 |
| Irudia 5-12 Prozesatze-nodoaren akats-proba egin ondoren lortutako hedapen-denborak..... | 5-22 |
| Irudia 5-13 Prozesatze-nodoaren akats-proba egin ondoren lortutako detekzio-denborak..... | 5-23 |
| Irudia 5-14 Prozesatze-nodoaren akats-proba egin ondoren lortutako baieztatze-denborak..... | 5-23 |
| Irudia 5-15 Prozesatze-nodoaren akats-proba egin ondoren lortutako Funtzionaltasunaren berreskuratze-denborak..... | 5-23 |
| Irudia 5-16 Prozesatze-nodoaren akats-proba egin ondoren lortutako Erreduantziaren berreskuratze-denborak..... | 5-24 |

TABLE OF TABLES

| | |
|---|-------------|
| Table 2-1 Summary of compliance with integration requirements | 2-24 |
| Table 2-2 Summary of compliance with requirements..... | 2-31 |
| Taula 3-1 Equipment-baliabideen ezaugarriak. Egokitzapena (“IEC 62264-2,” 2013, Table 11). | 3-3 |
| Taula 3-2 Material-baliabideen ezaugarriak. Egokitzapena (“IEC 62264-2,” 2013, Table 24). | 3-4 |
| Taula 3-3 Operations schedule objektuaren ezaugarriak. Egokitzapena (“IEC 62264-2,” 2013, Table 58)..... | 3-6 |
| Taula 5-1 Fabrikazio-zelulak ematen dituen muntaketa-zerbitzuen xehetasuna. | 5-5 |
| Taula 5-2 Operative Layer eta Functional layer geruzen artean trukaturako aldagaien zerrenda. | 5-8 |
| Table 5-3 Zerbitzu-geruzen eta operazio-geruzen artean trukaturako datu- egiturak..... | 5-9 |
| Taula 5-4 CeDRI ikerketa-zentroko ikertzaileekin lankidetzan proposaturako proba-agertokiak..... | 5-11 |
| Taula 5-5 CeDRI ikerketa-zentroko ikertzaileekin lankidetzan proposaturako KPIen deskribapena. | 5-11 |
| Taula 5-6 Hedapen-proba egin ondoren lortutako batezbesteko hedapen- denbora (N=20)..... | 5-17 |
| Taula 5-7 Prozesatze-nodoaren akats-proba egin ondoren lortutako batez besteko denborak (N=20). | 5-21 |

1 INTRODUCTION

1.1 Motivation

The industry is undergoing a deep transformation. Economic uncertainty and highly competitive markets are leading to an increase in customer demands for higher quality products at lower cost and with a high degree of customization, both in terms of features and production volume. The need to meet these demands has led to a burst of initiatives and technological developments that many institutions agree to identify as the Fourth Industrial Revolution (Schwab, 2016).

According to the World Economic Forum, “*the Fourth Industrial Revolution can be described as the advent of cyber-physical systems involving entirely new capabilities for people and machines*” (World Economic Forum, 2016). In turn, cyber-physical systems (CPS) are systems made up of interconnected entities consisting of a physical and a virtual part. These entities collaborate with each other to create an intelligent control circuit with adaptive capacity, autonomy and improved efficiency (Monostori, 2018; Zanero, 2017).

The transversality of the Fourth Industrial Revolution, and specifically of CPS, allows their application in multiple domains, although it is known by different names in each of them. Thus, it is common to see references to *Smart city* in the urban development domain (do Livramento Gonçalves et al., 2021), *Smart grids* in the energy supply domain (Faheem et al., 2018), or *Industry 4.0* in the manufacturing domain (Colombo et al., 2017).

In the case of the manufacturing domain, the object of study of this work, different governmental or private initiative institutions have emerged in order to present their strategies to face the challenges of Industry 4.0. These institutions have been committed to proposing different reference architectures over the last decade (Nakagawa et al., 2021; Fraile et al., 2019; Moghaddam et al., 2018). These reference architectures aim to identify the concepts and technologies to be used as a basis in the design and development of Industry 4.0 compliant solutions. As they address the Industry 4.0 problem from different approaches, it is possible that these reference

architectures are either mutually exclusive or complementary. This circumstance raises the following questions regarding this issue:

- Q1** What are the similarities and differences in the interpretation of Industry 4.0 offered by the main reference architectures?
- Q2** What are the advantages and disadvantages of each of them?
- Q3** Are these approaches complementary to each other?

Reference architectures for Industry 4.0 focus on ensuring interoperability, understood as *“the ability of two or more systems or components to exchange information and to use the information that has been exchanged”* (“IEEE Standard Computer Dictionary,” 1991), among the entities participating in CPS. However, they do not address the integration between the virtual and the physical parts that make up these entities, considering it as a step prior to Industry 4.0 (Schuh et al., 2020a). This step can be challenging because, in many cases, manufacturing systems exhibit great diversity when it comes to the communication capabilities of their assets, which makes scalability and reusability of integration solutions difficult. In this situation, two new questions arise:

- Q4** Are there currently any methodologies to guide the integration of manufacturing assets in Industry 4.0?
- Q5** If yes, do they allow generic integration regardless of asset type or communication technologies?

Another problem in the implementation of Industry 4.0 is managing the system's entities. In this sense, there are different opinions on what requirements the management platforms for this type of systems should meet, and how they should be structured. Similarly, there is no unanimity concerning which technologies to use to that end, although those recommended by the reference architectures can be considered. This lack of consensus leads to the following questions:

- Q6** What requirements must management platforms for manufacturing systems meet in the context of Industry 4.0?

- Q7** What functionalities should such platforms provide to ensure proper management of these systems?
- Q8** Which paradigms and technologies fit these requirements and functionalities?

Beyond these differences of criteria in the design and development of management platforms for Industry 4.0, it is remarkable the great interest these issues have aroused in the scientific community, considering the number of works related to them published in the last years (Pivoto et al., 2021). However, despite this fact, this interest has not yet resulted in the adoption of such solutions by the industry, at least on a general basis. This brings up the following questions:

- Q9** What obstacles hinder the transition of companies to Industry 4.0?
- Q10** What mechanisms and tools can be provided to facilitate this transition?

All these questions together summarize the most representative problems in the context of Industry 4.0, serving as a framework to pose and develop the work presented in this PhD dissertation.

1.2 Objectives

In order to provide answers to the questions raised about the reference architectures for Industry 4.0, the integration of manufacturing assets in it, its management through platforms, and the adoption of these by the industry, this work aims to design and develop a platform for the management of manufacturing systems framed in the context of Industry 4.0. To this end, the proposed platform must use concepts and technologies in line with the main reference architectures that allow it to offer the required infrastructure. In addition, it must include the required mechanisms and tools to allow a customizable and accessible development of the platform, facilitating the integration of manufacturing assets regardless of their characteristics. This way, the aim is to promote the adoption of this type of solutions in the industry, easing their implementation from a coherent, structured and replicable basis that guarantees the

fulfillment of all the necessary requirements for the management of manufacturing systems in Industry 4.0.

In order to achieve the main objective, a series of secondary objectives are proposed:

- Conduct a comparative study of the main reference architectures for Industry 4.0, analyzing their similarities and differences (Q1), in order to determine the advantages and disadvantages of each one (Q2), and to verify and if it is possible to establish synergies between their most important concepts (Q3).
- Design a methodology for the integration of manufacturing assets in Industry 4.0 (Q4), that is applicable regardless of the functional and communicative characteristics of the asset (Q5).
- Define a platform for the design and development of manufacturing systems in Industry 4.0 that guarantees the management of the operative state of the factory and its correct functioning (Q6), offering the infrastructure required for this (Q7), using appropriate technologies for this purpose (Q8).
- Analyze the causes that limit the use of manufacturing management platforms in Industry 4.0 (Q9) and provide mechanisms and tools to address this situation (Q10).
- Perform a proof of concept to validate that the proposed platform meets the requirements related to the management and monitoring of the manufacturing process (Q6), providing the necessary functionalities for the design and deployment of customized solutions (Q7). This implies defining a set of test scenarios and associated indicators and determining how to carry them out with the means available for this purpose.

1.3 Structure

Once the motivation and objectives of the PhD dissertation have been stated, the opening chapter concludes by presenting the structure of this document in the following chapters.

Chapter 2 – State of the art – is devoted to the analysis of the literature related to the design and development of manufacturing management platforms in the context of Industry 4.0. This chapter reviews the main reference architectures presented globally during the last decade to analyze the similarities and differences in their interpretation of Industry 4.0. In addition, the degree of adoption of the concepts proposed by these reference architectures in the literature is tested. To do so, an analysis of works dedicated either to the integration of manufacturing assets in Industry 4.0 or to their management through manufacturing management platforms is presented.

Chapters 3 to 5, summarized below, respond to the bulk of the secondary objectives presented in Section 1.2.

Chapter 3 – Industria 4.0rako euskarri metodologikoa – takes the reference architectures analyzed in Chapter 2 (see Section 2.2) as a starting point for developing approaches to support and guide in the adoption of Industry 4.0. First, a modeling approach is proposed to characterize the assets of the manufacturing domain within the Industry 4.0 paradigm. Then, a multi-layer approach is presented to address asset integration from a generic point of view, regardless of asset type or communication capabilities. Finally, a two-step approach for component management is proposed.

Chapter 4 – I4.0 Platformaren diseinua eta garapena – presents a platform for managing manufacturing systems. First, the platform is overviewed, describing its main elements and features. The following describes the infrastructure provided and how it supports the management of manufacturing systems under normal conditions or in the event of incidents in the manufacturing process (e.g., asset breakdown or malfunction, communications failure, etc.). Finally, the mechanisms offered by the platform for the customization of both its infrastructure and its participants (the so-called *I4.0 Components*) for a specific manufacturing application are explained.

Chapter 5 – Kontzeptu-proba – covers the procedure carried out to validate the proposed I4.0 Platform, which includes: the definition of test scenarios and associated metrics; the description of the demonstrator used, detailing the deployment of the

platform on it; the design of experiments according to the test scenarios; and, finally, the analysis of the results obtained to assess whether the platform conforms to the design requirements for which it was conceived.

Finally, to sum up, Chapter 6 – Conclusions and future works – answers the questions posed at the beginning of the PhD dissertation. The answers to these questions translate, on the one hand, into the contributions of this work, and on the other, into possible lines of future work.

2 STATE OF THE ART

2.1 Introduction

The objective of this chapter is to analyze what answers can be currently found in the literature related to the Industry 4.0 paradigm regarding the questions posed in Chapter 1. As a result of this review, it is expected to identify the gaps and/or limitations of the solutions proposed in the current literature with respect to these questions. Furthermore, this review also allows to obtain a better knowledge of the state of the art in the field, which can serve as a basis for addressing the main objective of this work: the design and development of a manufacturing management platform in the context of Industry 4.0.

The large volume and variety of works related to Industry 4.0 makes it essential to consider a limited and organized review of the literature in order to facilitate the process. For this purpose, the questions posed in Chapter 1 have been grouped into four blocks according to their affinity. Hence, Section 2.2 analyzes the main reference architectures proposed by both governmental and private institutions over the last decade, describing the concepts and standards on which they are based. This section also reviews works that make comparisons between these reference architectures to analyze their differences and possible synergies, thus answering questions Q1-Q3. These architectures establish the conceptual and normative framework for Industry 4.0, but do not provide technological support for the development of concrete solutions. The first step in this direction is to achieve interoperability among all system participants. For this reason, Section 2.3 discusses the integration of manufacturing assets in Industry 4.0, analyzing the generality and replicability of works focused on this topic, addressing questions Q4-Q5. The integration of assets in Industry 4.0 ensures effective interactions between them. However, the management of manufacturing systems usually requires additional infrastructure that enables, among others, the creation, registration, maintenance and deletion of system participants. Therefore, Section 2.4 presents an analysis of works oriented to the design and/or development of manufacturing management platforms. This analysis focuses on the services and functionalities offered by these platforms, on the technologies used for this purpose, and on their suitability to the reference architectures, covering questions

Q6-Q8. As for questions Q9-Q10, they are considered cross-cutting in nature given the problems they address (the analysis and resolution of the problems that hinder the transition of companies towards Industry 4.0), so they have been taken into account throughout the chapter. Finally, the conclusions of this analysis are presented in Section 2.5.

2.2 Reference architectures for Industry 4.0

As noted in Chapter 1, several institutions have emerged over the last decade aiming to facilitate a coherent and smooth transition towards Industry 4.0. The most relevant of the actions carried out by these institutions is the elaboration and publication of their reference architectures. These architectures propose a contextualization of Industry 4.0 to ensure an understanding among its participants, providing a starting point for the design and development of solutions on a common basis.

This section analyzes and describes the main reference architectures for Industry 4.0. The first four subsections are devoted to architectures proposed by globally relevant institutions, such as *Plattform Industrie 4.0*, *Industry IoT Consortium (IIC)*, *China Manufacturing 2025* or *Industrial Value Chain Initiative (IVI)*. In addition, a fifth subsection has been included which reviews works focused on the comparative analysis of these reference architectures. Finally, the section concludes with a brief discussion.

2.2.1 Reference Architectural Model for Industrie 4.0

The term *Industrie 4.0* is coined by Henning Kagermann, Wolf-Dieter Lukas y Wolfgang Wahlster, who introduce it for the first time at the 2011 Hannover Fair (Kagermann et al., 2011). The idea behind this term proposes the intelligent interaction of humans, machines and processes through the integration of CPS and the Internet of Things (IoT) (“Digital Transformation Monitor Germany: Industrie 4.0,” 2017). The combination of these concepts can be applied to different functions, such as flexible manufacturing, the

modular factory, customer (or product) oriented manufacturing, and logistics optimization, among others (“What is Industrie 4.0?,” n.d.).

This idea received strong political support from the first moment, and in 2013, the German Federal Government uses the same event to present the *Plattform Industrie 4.0* (“Plattform Industrie 4.0,” n.d.). This platform places the irruption of the fourth industrial revolution in the imminent future. Under this premise, current production models are obsolete, and a new reference framework is needed to enable intelligent interconnection between the elements of a system. To do so, *Plattform Industrie 4.0* presents in 2015 the *Reference Architectural Model for Industrie 4.0 (RAMI 4.0)* (Adolphs and Epple, 2015). In 2016, *RAMI 4.0* is adopted as a standard by the German Institute for Standardization (Deutsches Institut für Normung or DIN) (“DIN SPEC 91345,” 2016), and in 2017 is recognized by the International Electrotechnical Commission (IEC) (“IEC PAS 63088,” 2017). *RAMI 4.0* is a service-oriented architecture that aims to provide a structured description of manufacturing systems in Industry 4.0 (hereafter referred to as I4.0 System). To this end, it presents a cubic model that provides a framework for a common understanding among the elements of *I4.0 Systems*, called *I4.0 Components*, with respect to three axes defined from several reference standards (Figure 2-1):

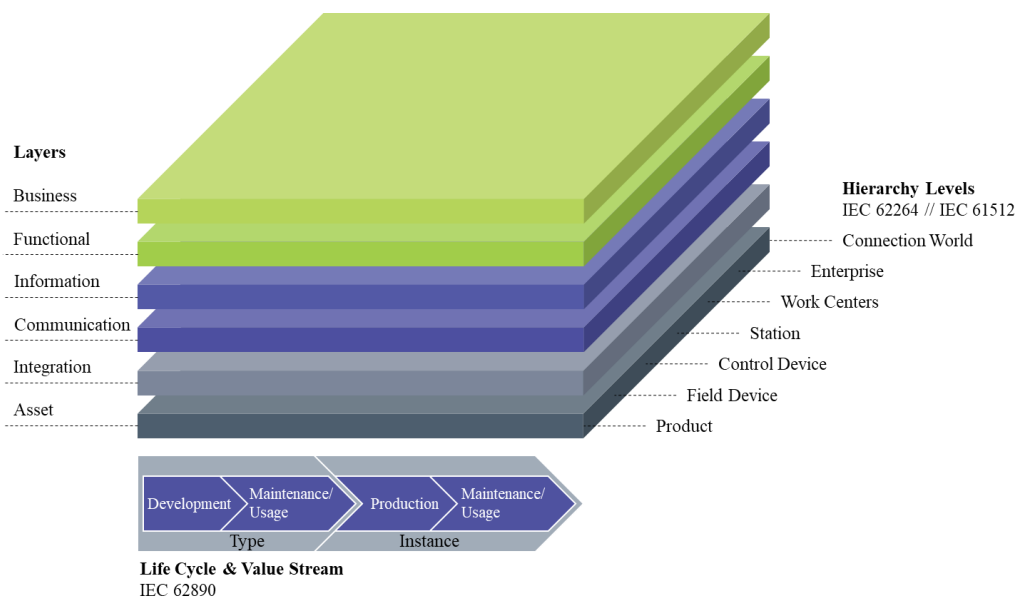


Figure 2-1 Cubic model proposed by RAMI 4.0. Adapted from (Adolphs and Epple, 2015, Figure 1).

- *Layers axis*: the vertical axis of *RAMI 4.0* represents the different functional levels involved in the implementation of manufacturing systems for Industry 4.0. These layers interact so that each provides services to the upper layer and manages the services provided by the lower layer. *RAMI 4.0* defines six layers:
 - *Business layer*: it describes the business strategy of the company (regulatory and legal framework, business strategy planning, economic aspects, etc.). It also orchestrates the services provided by the functional layer to obtain information about the state of the process at a global level.
 - *Functional layer*: it provides a formal description (functionalities and data) of the services offered by an asset and manages their access remotely.
 - *Information layer*: it is responsible for ensuring the integrity, integration and persistence of the data received from the assets. It is also responsible for acquiring, processing and adapting this data, and providing it in a structured manner through interfaces to support the services of the *Functional layer*.
 - *Communication layer*: it establishes uniform and predefined data formats to ensure interoperability between system participants, allowing assets to access the functionalities and data offered by others. It is also responsible for managing the services offered by the *Integration layer*.
 - *Integration layer*: it represents the transition between the physical and virtual worlds. This layer offers services that enable access to assets, providing information about their functionalities and data, allowing them to be used as intended.
 - *Asset layer*: it represents the real world, understood as the set of all physical or logical entities with value for a company, including human beings.
- *Life Cycle & Value Stream axis*: it is supported by IEC 62890 (“IEC 62890,” 2020). It provides models to describe the operative state of the product. The most noteworthy aspect of this standard is the differentiation between product type and product instance:

- *Product type*: it is defined by a unique identifier and all the information associated with the product development stage (i.e., from design to prototype production).
- *Product instance*: each product manufactured constitutes an instance of a product type. It is defined by a unique identifier (e.g., a serial number), and by information associated with the manufacturing process of that particular instance.
- *Hierarchy Levels axis*: it is based on IEC 62264 (“IEC 62264-1,” 2013) and IEC 61512 (“IEC 61512-1,” 1997), which correspond to ISA 95 and ISA 88, respectively. These standards set out a factory hierarchy that corresponds to the traditional interpretation of the automation pyramid (also known as Computer-Integrated pyramid or CIM pyramid). *RAMI 4.0* proposes to partially follow the hierarchy proposed in these standards, adopting some of its levels such as *Enterprise*, *Work centers*, *Stations* and *Control device*. Additional levels have been added and/or modified at both ends to make a hierarchy consistent with Industry 4.0 (“DIN SPEC 91345,” 2016, Figure 9):
 - *Connected world*: it is located above the *Enterprise* level, making it possible to represent a group of companies collaborating throughout the life cycle of a product.
 - *Field device*: it is a subcategory of the *Control device* hierarchy level. It is used to represent devices that are directly involved in the manufacturing process (e.g., a sensor).
 - *Product*: this level gives the product its own place in the Industry 4.0 hierarchy, offering a decoupled view with respect to machine planning.

Along with this cubic model, which enables understanding between the participants of a system, *RAMI 4.0* also introduces the *I4.0 Component*. This concept can be understood as a particularization of the participants of a CPS aligned with *RAMI 4.0* (Ye and Hong, 2019). As shown in Figure 2-2, *I4.0 Components* consist of an asset (physical part) and an *Asset Administration Shell* or *AAS* (virtual part), which represents the asset in the *I4.0 System*, allowing it to interact with other system participants through *RAMI 4.0* compliant communication.

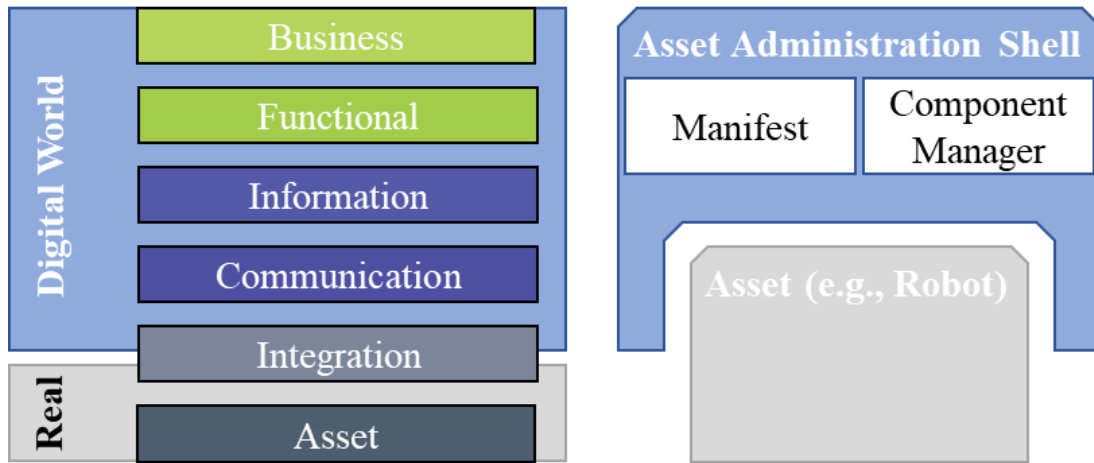


Figure 2-2 Elements of the I4.0 Component positioned with respect to the RAMI 4.0 layers. Adaptation of (Schweichhart, 2016).

Depending on the way in which this interaction takes place, AASs are classified into three categories (see Figure 2-3):

- *Passive AAS*: they consist only of their submodels (a series of structured properties that describe and/or identify the data and functionalities of the asset) and can be exchanged as a whole in the form of a file.
- *Reactive AAS*: they can attend to standardized service requests.
- *Proactive AAS*: they can establish a peer-to-peer interaction with other AAS, offering and requesting services when required to meet their own goals.

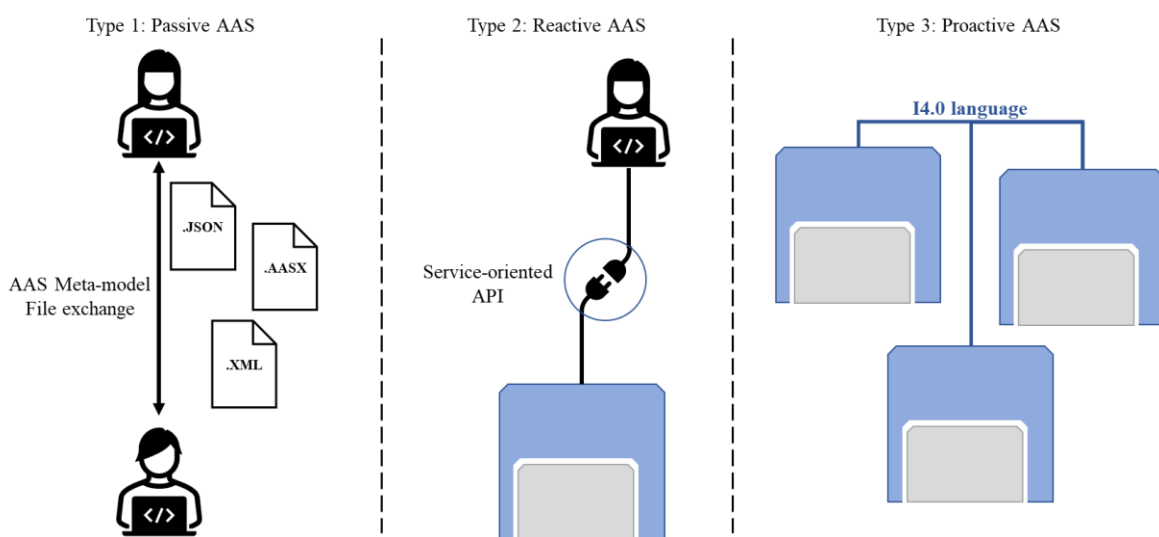


Figure 2-3 Interaction types of AASs. Adapted from (Miny et al., 2021, Figure 14).

I4.0 Components are service-oriented: they can offer services through an Application Programming Interface (*API*) provided by the *AAS*. These service requests are handled by a *Component Manager*. To that end, the *Component Manager* manages the *AAS* submodels, which are also called *Manifest*.

The services provided by *I4.0 Components* to each other can be combined to compose manufacturing applications, which is why they are called *Application Relevant Services* (Miny et al., 2021). *Application Relevant Services* can be classified into two categories:

- *Submodel Services*: they are intended to provide total or partial access to the information of a submodel. These services do not require interaction with the asset (e.g., querying the serial number of the asset, etc.).
- *Asset Related Services*: they involve an interaction with the asset to access the results of the execution of some functionality or to manage its state (e.g., start-up, controlled shutdown, etc.).

In addition to *Application Relevant Services*, *I4.0 Systems* also require an infrastructure to manage *I4.0 Components* and support them in the execution of manufacturing applications. Therefore, apart from *Application Relevant Services*, *Plattform Industrie 4.0* defines another category of services called *Infrastructure Services*. *Infrastructure Services* can be used by *Application Relevant Services*, supporting the interaction with and between *I4.0 Components*. *Infrastructure Services* are responsible for, among others, access control to *Application Relevant Services*, management of *AAS*-related information, creation, registration and deletion of *AASs*, or controlled exposure of *AASs*. In order to provide these functionalities, *Infrastructure Services* are classified into two categories:

- *AAS Services*: they are responsible for managing the information and functionalities (i.e., the *Application Relevant Services*) of the *I4.0 Components* and can therefore be performed by the *AASs* themselves.
- *AAS Infrastructure Services*: they serve to manage the *AASs* in the system as a whole (e.g., to register and create them, or to make them traceable to each

other). These services cannot be performed by the AASs themselves. Instead, they are offered by platforms in charge of managing the I4.0 Systems (hereafter referred to as I4.0 Platforms).

Consequently, from the above it can be concluded that I4.0 Systems are hybrid in nature, having a distributed part developed by the AASs themselves through the *AAS Services* (creation and management of submodels), and a centralized part performed by the I4.0 Platform through the *AAS Infrastructure Services* (creation, registration and management of AASs).

2.2.2 Industrial Internet Reference Architecture

The next initiative in terms of both age and relevance is *Industry IoT Consortium (IIC)* (“Industry IoT Consortium,” n.d.). This initiative was formed as a non-profit consortium in 2014, aiming to catalyze and coordinate the progress of industry, academia and the U.S. Government around the *Industrial Internet* concept, promoted by the General Electric Company in 2012 (Evans and Annunziata, 2012). This concept proposes the application of Internet-based technologies to industry to enable interconnected systems capable of processing and analyzing large amounts of data.

In 2015, the *IIC* releases the *Industrial Internet Reference Architecture*, or *IIRA*, to guide the development of these intelligent systems. This architecture is based on the ISO/IEC/IEEE 42010 standard on description of architectures in systems engineering. (“Systems and software engineering — Architecture description,” 2011). According to this standard, software architectures should be organized in *Viewpoints*. The *Viewpoints* frame the description and analysis of specific problems in the system. These problems, named *Concerns*, can refer to any relevant aspect of the system. On the other hand, the system participants, or *Stakeholders*, may show interest in different *Concerns*, and, by extension, in different *Viewpoints* of the system (Lin et al., 2019, p. 6). Specifically, *IIRA* is structured in four *Viewpoints* (see Figure 2-4):

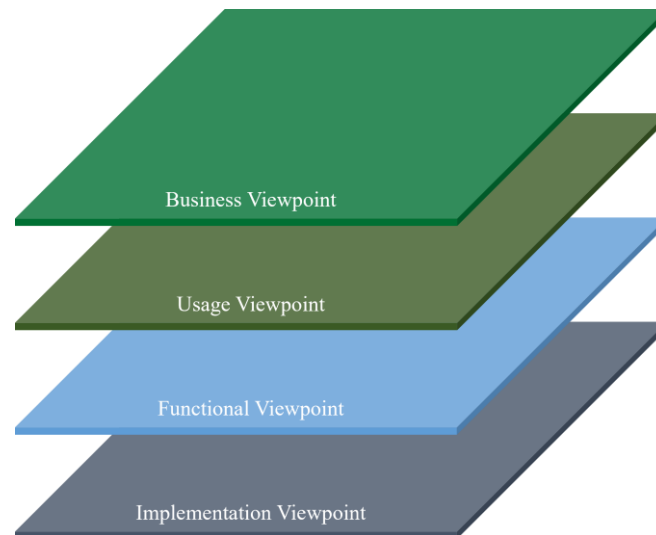


Figure 2-4 Viewpoints proposed in IIRA. Adapted from (Lin et al., 2019, Figure 3-4).

- *Business Viewpoint*: it deals with *Concerns* related to vision, values and objectives at the enterprise level (e.g., calculation of maintenance costs and expected benefits, assessment of product liability, etc.), as well as identifying the capabilities of the system to meet these objectives. *Stakeholders* interested in this *Viewpoint* are typically business decision makers, product managers and system engineers.
- *Usage Viewpoint*: it addresses *Concerns* related to system capabilities by breaking them down into sequences of smaller activities that may involve people or other system components (i.e., production planning). These activities provide functionality that can be combined to provide the system capabilities identified in the *Business Viewpoint*. *Stakeholders* interested in this *Viewpoint* are typically system engineers and product managers.
- *Functional Viewpoint*: its main *Concern* are the functional components of the IIoT system (their structure, and interfaces both with other system components and with external elements). *Stakeholders* interested in this *Viewpoint* are typically system and component architects, developers and integrators.
- *Implementation Viewpoint*: it focuses on *Concerns* involved in the implementation of the functional components of the system (e.g., the selection of technologies and devices). *Stakeholders* interested in this *Viewpoint* are typically system and component architects, developers and integrators, and system operators.

2.2.3 Intelligent Manufacturing System Architecture

In 2015, the Chinese Government presented its strategic plan for the next decade, under the name *China Manufacturing 2025* (“China Manufacturing 2025,” 2017). This strategic plan proposes different national projects, including one focused on the concept of *Intelligent Manufacturing*. This concept is proposed as a key element in transforming Chinese industry, which is characterized by the production of cheap, low value-added goods, into a cutting-edge industry. To this end, it proposes the integration of new information and communications technology (ICT), new materials, new control tools and new machinery.

That same year, in view of the need to standardize efforts dedicated to *Intelligent Manufacturing*, the *Intelligent Manufacturing System Architecture (IMSA)* was presented as part of the *China Manufacturing 2025* strategy (Wei et al., 2017). This reference architecture is strongly influenced by *RAMI 4.0*, and, like this one, consists of three dimensions (Figure 2-5):

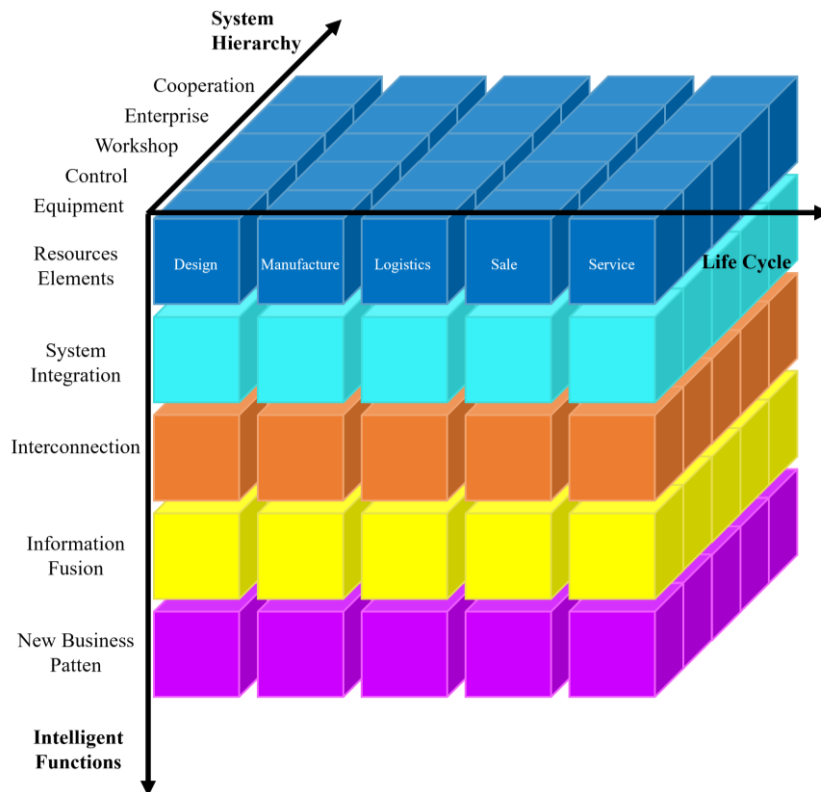


Figure 2-5 Cubic model proposed by IMSA. Adapted from (Wei et al., 2017, Figure 1).

- *Life Cycle*: it represents the chain of activities that add value to a product from design to commissioning. *IMSA* considers the following stages in the product life cycle:
 - *Design*: it covers the simulation, construction, verification and optimization of a product prototype.
 - *Manufacture*: it corresponds to the manufacturing process of the product.
 - *Logistics*: it refers to the transportation of products to final destination.
 - *Sale*: it covers the commercial activity by which products are transferred to customers.
 - *Service*: it includes product decommissioning and recycling at the end of the useful life of the product.
- *System Hierarchy*: as *RAMI 4.0*, it takes as its starting point the IEC 62264 standard (“IEC 62264-1,” 2013), from which it gets its four lower levels: *Equipment, Control, Workshop* and *Enterprise*. *IMSA* proposes to add at the top of the hierarchy the *Cooperation* level to illustrate the collaboration between companies throughout the life cycle of a product.
- *Intelligent Functions*: this axis proposes to divide the high-level functionalities provided by ICTs. *IMSA* defines five functional levels:
 - *Resources*: it corresponds to data generation, associated with the factory resources.
 - *System integration*: it manages the collection of resource data and connects them to the intelligent manufacturing system.
 - *Interconnection*: it provides the necessary means to enable communication between participants in the system.
 - *Information fusion*: it is responsible for the storage and processing of data received from the lower levels.
 - *New business pattern*: uses system data to improve business strategy from different angles such as product customization, network collaboration, or remote operation on the system.

2.2.4 Industrial Value Chain Reference Architecture

Similar to the Chinese example, the Japanese *Industrial Value Chain* initiative appeared in 2015 (“Industrial Valuechain Initiative,” n.d.). Like its predecessors, this initiative raises its own reference architecture, the *Industrial Value chain Reference Architecture (IVRA)* (“Industrial Value Chain Reference Architecture (IVRA),” 2016). *IVRA* proposes an advanced manufacturing model based on a set of functionalities called *General Function Blocks (GFB)*, distributed along three axes (Figure 2-6):

- *Knowledge and Engineering Flow*: it is focused on the product, covering its entire life cycle. It is divided into five stages: *Marketing and Design, Construction and Implementation, Manufacturing Execution, Maintenance and Repair* and *Research and Development*.
- *Demand/Supply Flow*: it focuses on the value chain, from process planning to after-sales service. It is again divided into five stages: *Master Planning, Material Procurement, Manufacturing Execution, Sales and Logistics* and finally *After Services*.
- *Hierarchy Levels*: the *GFBs* corresponding to the two previous axes can be applied at different levels of the factory hierarchy. Unlike other reference architectures, *IVRA* does not specify any standard to be followed in this respect, and proposes a four-level structure: *Enterprise, Department, Floor* and *Device*.

Another concept introduced by *IVRA* is the *Smart Manufacturing Unit (SMU)*, which defines any participant in a smart manufacturing system. *IVRA* and proposes a cubic model to define *SMUs* from three points of view (Figure 2-7):

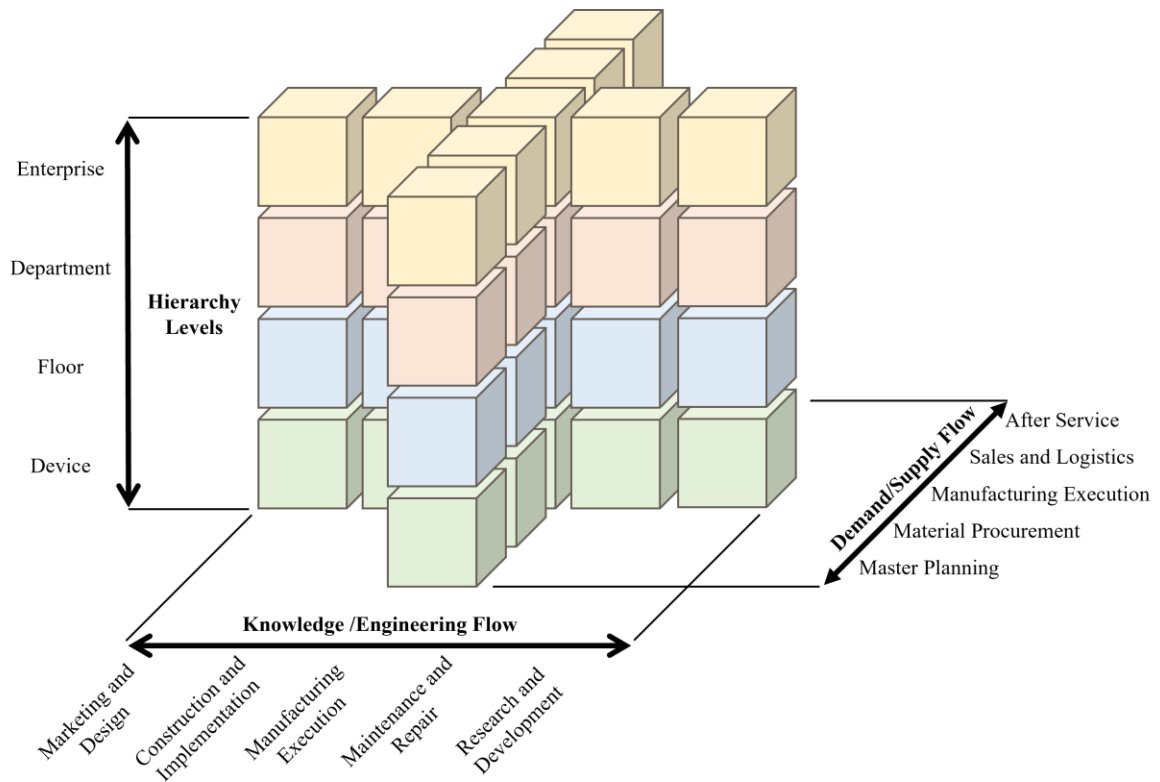


Figure 2-6 Model for advanced manufacturing proposed by IVRA. Adapted from (“Industrial Value Chain Reference Architecture (IVRA),” 2016, Figure 2).

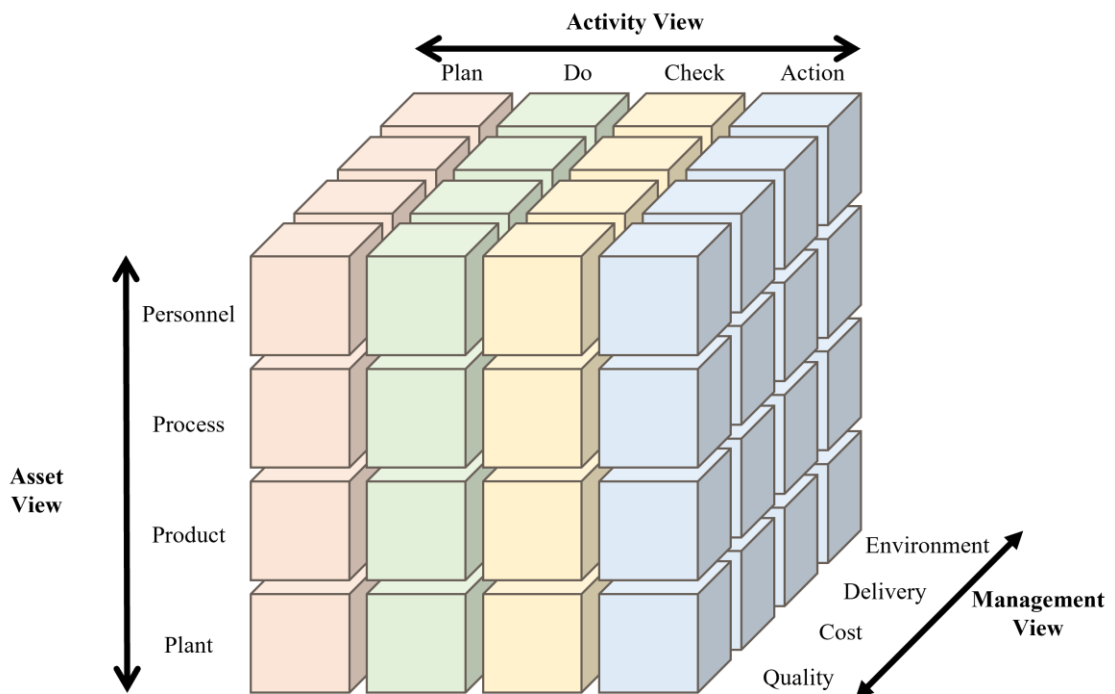


Figure 2-7 View of the SMU proposed by IVRA. Adapted from (“Industrial Value Chain Reference Architecture (IVRA),” 2016, Figure 1).

- *Asset View*: it shows the assets that are valuable to the company. An *SMU* can contain one or several assets, and these in turn can become part of other *SMUs*. Four types of assets are proposed:
 - *Personnel*: personnel working in the factory, both in performing manufacturing operations and in making decisions about the process.
 - *Process*: technical and methodological knowledge about manufacturing processes is considered a valuable asset.
 - *Product*: the products manufactured during the process and the components and materials consumed for it.
 - *Plant*: machinery and other equipment (control devices, tools, etc.) used in the manufacturing process.
- *Activity View*: it covers the activities performed by the assets comprising the *SMUs*. All activities can be classified into four basic types:
 - *Plan*: to make a list of actions to be carried out in a specific period and with a specific objective.
 - *Do*: perform certain actions in the physical world in order to achieve a particular goal.
 - *Check*: determining whether the objectives defined in the planning activities have been achieved.
 - *Action*: make decisions autonomously to improve *SMU* performance based on the results of testing activities.
- *Management View*: it focuses on the different management areas that can be worked on to optimize the performance of the *SMU*. Four areas are proposed that can be worked on independently:
 - *Quality*: set of indicators to evaluate how the characteristics of a product or service provided by *SMU* meet the client's needs. It is also possible to speak of quality when internally evaluating the equipment or personnel that form part of the *SMU*.
 - *Cost*: the sum of economic, material and personal resources used by *SMU* to provide a given product or service.
 - *Delivery*: measure of the time and manner in which the products and/or services required by a customer are delivered.

- *Environment*: assessment of the environmental impact caused by *SMU* in the course of its activities.

2.2.5 Alignment and synergies

Despite the differences they may exhibit, all the reference architectures reviewed in this chapter have the same goal: to achieve interoperability between manufacturing system participants, both internally and externally. However, in an increasingly globalized and interconnected world, the area of influence of these reference architectures and the initiatives that promote them is limited. This has motivated a series of collaborations between these initiatives in search of synergies that can lead to a global standard. Specifically, these collaborations have positioned the different reference architectures with respect to *RAMI 4.0*, due to its status as the oldest reference architecture.

The closest cooperation has been established between *Plattform Industrie 4.0* and *IIC*. In 2017, they publish together a paper in which they perform a comparative analysis between *RAMI 4.0* and *IIRA* (Lin et al., 2017). In it, they study the similarities and complementarity of the main concepts of *RAMI 4.0* and *IIRA*, reaching several conclusions:

- A relationship is established between the layers defined in the *Layers* axis of *RAMI 4.0* and the functional domains considered by *IIRA* in its *Functional Viewpoint*.
- It is confirmed that there is a solid foundation for facilitating interoperability between *RAMI 4.0* and *IIRA*-based solutions in terms of connectivity and communication, but that work remains to be done on semantic interoperability.

Subsequently, in 2020 both organizations publish a new joint paper comparing the *IIC* understanding of the *Digital Twin (DT)* with the *AAS* concept of *RAMI 4.0* (Boss et al., 2020). As a result, it is concluded that the *AAS* provides full support for the key requirements of *DTs* described and characterized by the *IIC* (connectivity, deployment,

interoperability, etc.). Following on from these comparative analyses, *Plattform Industrie 4.0* as initiated the publication of a series of papers outlining how to apply the *Viewpoints* presented in *IIRA* to the *AAS* (Braune et al., 2019; Miny et al., 2021).

Similarly, as a result of the Sino-German cooperation commission on standardization, a report analyzing the synergies between *RAMI 4.0* and *IMSA* is presented in 2018 (“Alignment Report for Reference Architectural Model for Industrie 4.0/ Intelligent Manufacturing System Architecture,” 2018). In this case, the influence of *RAMI 4.0* upon *IMSA* is clearly noticeable, since the mapping between axes is practically straightforward. The only exception is on the *Life Cycle & Value Stream* axis, where both partners agree to work on a consensus proposal. However, to date, they have not published any progress in this regard.

Regarding cooperation with Japanese institutions, a collaboration between *Plattform Industrie 4.0* and *IVI* was initially envisioned to realize a harmonization between *RAMI 4.0* and *IVRA* (Nishioka, 2017). However, that collaboration has not progressed, and in more recent publications *Plattform Industrie 4.0* has collaborated with the *Robot Revolution & Industrial IoT Initiative (RRI)* (“Robot Revolution Initiative,” n.d.). However, all work resulting from this collaboration has been directly contextualized within *RAMI 4.0* (Braune et al., 2019).

In addition to these collaborations, reference architectures have also been studied by independent researchers. Several works are devoted to performing comparisons focusing on different technical or formal aspects. For example, (Zeid et al., 2019) analyze how different reference architectures treat the concept of interoperability in the context of Industry 4.0, determining that they are all still in a conceptual stage. In a similar analysis, (Moghaddam et al., 2018) conclude that reference architectures should enable the capabilities of participants in I4.0 Systems to be standardized and published as services (both features inherent to *RAMI 4.0* and *I4.0 Components*). (Han, 2020) compares eight reference architectures focusing on the lifecycle axes of the reference architectures and how they address the *DT* concept. (Nakagawa et al., 2021) focus on the impact of different reference architectures in the literature, highlighting

that *RAMI 4.0* and *IIRA* are by far the most cited. (Bader et al., 2019) argue that there is a strong need to align the different reference architectures in order to create a complete picture of the state of the art. To this end, they position all the analyzed reference architectures with respect to *RAMI 4.0* and *IIRA*, as they consider that together they cover most of the aspects required in Industry 4.0. Finally, (Soares et al., 2021) made a comparison with respect to 24 aspects organized in 5 categories (cyber, physical, lifecycle, human and comprehensive), with *RAMI 4.0* being the most complete reference architecture.

In contrast, other works analyze reference architectures as a starting point from which to make new contributions. (Li et al., 2018) discusses the set of standards and regulations on which several reference architectures are based, and develop a compendium of standards to serve as a guide for companies and help researchers to identify areas to work on. (Fraile et al., 2019) propose a new reference model based on the analyzed reference architectures, which they call the Industrial Internet Integrated Reference Model. This reference model is based on the *IIRA Viewpoints*, in which it integrates functionalities of *RAMI 4.0* and standards of the *National Institute of Standards and Technology (NIST)*. In the same way, (Cañas et al., 2022) propose a framework for production planning and control systems in Industry 4.0, analyzing several reference architectures and adopting the *RAMI 4.0* axes for their proposal.

2.2.6 Discussion

The analysis of these works shows that particularly *RAMI 4.0*, and to a lesser extent *IIRA*, are prevailing as the reference architectures to follow, both because of their age and the completeness of their proposal. The interest of the scientific community is evidenced by the large number of research papers focused on this topic, as shown in the previous subsection.

However, the follow-up of these reference architectures is moderate or non-existent in many companies (World Economic Forum and McKinsey & Company, 2018). This is because, as they aim to as generic as possible (to be applicable to the entirety of their

application domain), *RAMI 4.0* and *IIRA* are presented from a technologically neutral point of view. However, this approach may be too abstract for companies and practitioners, who lack the methodological and technological support to translate it into real-world solutions (Nakagawa et al., 2021; Trunzer et al., 2019). This circumstance is a brake on the transition of companies towards Industry 4.0.

2.3 Integration of assets in Industry 4.0

Institutions interested in fostering the leap to Industry 4.0 do not limit themselves only to proposing the reference architectures seen in the previous section, they also offer mechanisms to facilitate the adoption of the main precepts of Industry 4.0. In this regard, the German National Academy of Science and Engineering (Acatech) proposed in 2017 the Industrie 4.0 maturity index (Schuh et al., 2020a). This document proposes a roadmap consisting of six stages to reach the full development of the Industry 4.0 paradigm (see Figure 2-7): *computerisation*, *connectivity*, *visibility*, *transparency*, *predictive capacity* and *adaptability*. The first two stages constitute a previous digitization phase that should ensure the basic conditions for the adoption of Industrie 4.0: the *computerisation* stage is related to the automation of manufacturing services, while the *connectivity* stage focuses on communication with and between assets to provide accessibility to these services.

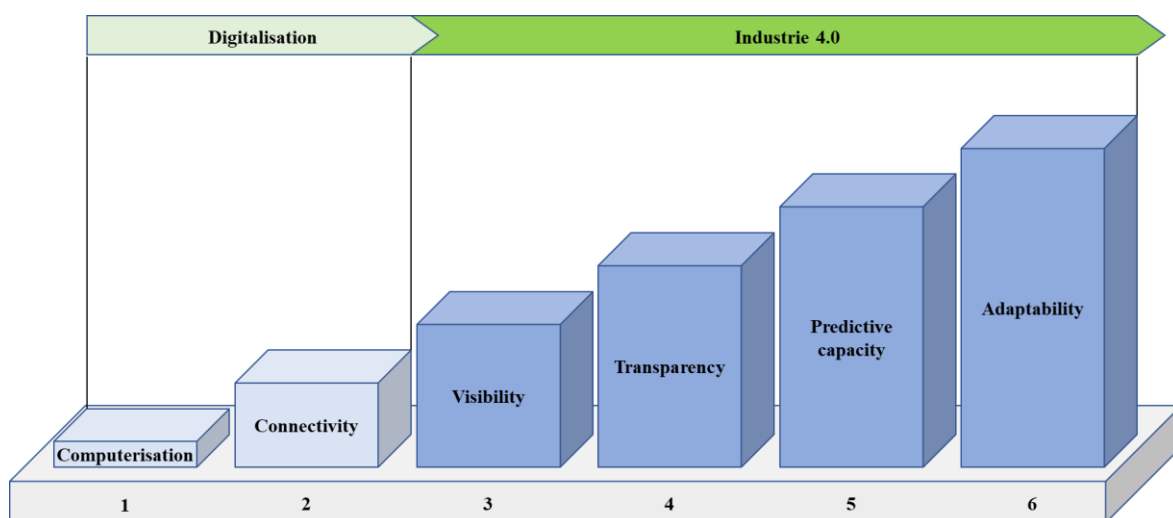


Figure 2-8 Stages in the Industry 4.0 maturity index. Adapted from (Schuh et al., 2020a, Figure 6).

With this in mind, it is worth pointing out the results of the survey conducted by Acatech on the current situation of companies with respect to the maturity index (Schuh et al., 2020b): by April 2020, only 4% of the participants had reached the first stage of the Industrie 4.0 phase (*visibility*). Therefore, 96% of the companies surveyed had not passed the stages of computerization (16%) and connectivity (80%). Given these percentages, it can be concluded that asset connectivity is a source of problems for companies. The reason is that factories often have a great diversity of physical assets, with significant differences in their functional and communication capabilities between them. This explains why, despite significant efforts in this field, solutions are often ad hoc, focused on specific types of assets or integration technologies (de las Morenas et al., 2017; Karnouskos and Leitão, 2017; Martin et al., 2019).

This section is devoted to review works focused on the integration of assets in Industry 4.0 according to the precepts of *RAMI 4.0*, the most widespread reference architecture. The first subsection describes how the search for papers was carried out and proceeds with their analysis. In the second, the requirements that have been established to compare these jobs are defined and their conformity with respect to the requirements is briefly discussed.

2.3.1 Related work

The problem of manufacturing asset integration has long been of interest to researchers in the context of holonic manufacturing systems (Leitao and Restivo, 2006; Van Brussel et al., 1998), and more recently, in the context of CPS (Karnouskos et al., 2020; Leitão et al., 2022) and *I4.0 Components* (Arm et al., 2021; Ye et al., 2021). As seen in Section 2.2, *RAMI 4.0* is the predominant reference architecture for Industry 4.0 at this time. Thus, it has been decided to focus the search for papers around the *Asset Administration Shell* concept, since it is the designated entity in *RAMI 4.0* to achieve asset integration. However, the search has been extended to works dealing with asset integration from the CPS perspective because of the similarity between this concept and the *I4.0 Component* concept. The search for works was conducted in the

bibliographic database Web of Science¹, combining the results of two search strings “Asset Administration Shell” and “integration”, and another with the concepts “asset”, “integration” and “Cyber Physical Systems”). After discarding the works prior to 2015 (the year in which *RAMI 4.0* was presented) and/or those that did not fit the subject, a selection of 28 articles was obtained, which are reviewed below.

Some of these works are aimed at solving specific integration problems, and propose ad-hoc implementations for particular case studies (Neal et al., 2021; Marcon et al., 2018). Others, although not ad-hoc, focus on a specific type of asset. For example, several papers focus on the integration of Programmable Logic Controllers (PLC). Most of them are oriented to standards such as IEC 61131-3 (Cavalieri and Salafia, 2020; de las Morenas et al., 2017; Pethig et al., 2017) or IEC 61499 (Dai et al., 2017; Wenger et al., 2018). These works also present different alternatives on where to deploy the AAS: inside the PLC as a function block (Pethig et al., 2017), or on an external server, outside the PLC (Cavalieri and Salafia, 2020; Wenger et al., 2018). The latter argue that AAS interactions can compromise the real-time constraints of PLC programs. Another type of asset that is getting a lot of attention is robots. The Robot Operating System (ROS) is commonly used at different levels to integrate robots (Martin et al., 2019; Schou and Madsen, 2017): some researchers use ROS as a means of communication between the AAS and the assets (Martin et al., 2019). Others, instead, propose to use ROS for the interactions between AASs, choosing the most suitable communication protocol to interact with each asset (Schou and Madsen, 2017). These solutions are suitable for solving specific problems, as they propose the most appropriate technologies and integration techniques for each type of asset. However, this also limits them in terms of reusability and scalability.

In other works, the focus is not on the type of asset to be integrated, but on which technologies to use for this purpose. In this regard, two approaches stand out above the rest: OPC Unified Architecture (OPC UA) and the industrial agent paradigm.

¹ <https://www.webofscience.com/wos/woscc/basic-search>

The use of OPC UA is promoted in different documents published by *Plattform Industrie 4.0*, and there are several researchers presenting solutions based on this technology. For example, (Contreras et al., 2017) propose the use of OPC UA for communication between the asset and the AAS, and a combination of AutomationML and OPC UA supported by the IEC 62769 standard for communications between the AASs. (Fuchs et al., 2019) propose an AAS submodel to represent field-level devices based on the OPC UA Part 100: Devices specification standard. Similarly, in (Zimmermann et al., 2019) a skill-oriented metamodel is proposed, where skills are understood as the potential services that an asset can offer. (Ye et al., 2021) presents a three-layer architecture (Field assets, edge AAS deployment, and cloud AAS management), in which OPC UA is used both to integrate manufacturing assets (establishing a communication bus) and to implement AASs (deployed together in an aggregation server).

Another option is using industrial agents for asset integration. Industrial agents (or simply agents) are software entities with an inherent ability to compete and/or collaborate with each other to achieve their goals, and that naturally approach asset integration (Karnouskos et al., 2020). For instance, (Leitão et al., 2018) propose a classification for what they call integration patterns (different strategies for integrating assets using industrial agents), attending to the location of the agent and its interaction mode with the asset and discuss the benefits and drawbacks of the different alternatives. (Sakurada et al., 2022) propose an agent-based AAS concept in which functionalities are coded as microservices, including an agent-based AAS interface (to interact with other agent-based AASs), an asset interface (to interact with the asset) and asset information (to store static asset information). (Leitão and Barbosa, 2016) proposes an agent-based solution to establish standard interfaces, regardless the concrete communication protocols, presenting an approach based on the use of the ISO 9506 Manufacturing Message Specification international standard. (Bakliwal et al., 2018) propose a three-level approach, in which the first layer receives the information from the asset and provides it a standardized format, so upper data processing layers can be generic. These solutions can be applied to different assets, but still depend on the characteristics of the technology used to implement the integration solution. This

can be a limitation in case of having to integrate an asset that is not compatible with the technologies used in the solution.

Beyond the works oriented to specific assets or technologies, there is another group of works, with a more generic approach, that share their concern for heterogeneity in AAS development (which includes asset integration). In this regard, (Ashtari Talkhestani et al., 2019) propose a software module composed of an interface adapted to the capabilities of the asset it integrates, and a generic interface to facilitate interoperability. (Seif et al., 2019) propose a methodology for the implementation of AASs. Within this methodology, the third step corresponds to asset integration. This step comprises the creation of databases, the implementation of a communication protocol and the integration of the IoT. (Tantik and Anderl, 2017) make a proposal to define the technical structure of the AAS, according to which the AAS is subdivided into six parts: external interface (towards the I4.0 System), authentication and security, data management, functionality, administration and internal interface (towards the asset). However, as far as the internal interface is concerned, they claim that it does not make sense to propose a standardization, given the diversity of AAS integration solutions. In (Orio et al., 2019) an implementation of the AAS concept, called NOVAAS, is proposed. This work also points out that an internal interface is required to connect the asset to the AAS. For this purpose, a Façade software design pattern is proposed. (Ye et al., 2020) present a methodology for implementing AAS which supports the plug and produce (PnP) concept. This methodology consists of five steps (physical configuration, pre-integration, digital representation, AAS communication and system evaluation), of which the first three are related to asset and AAS integration. Finally, (Dorofeev et al., 2017) propose a very similar approach, also oriented to PnP, in which they propose a methodology based on four phases (discovery, configuration, production and reconfiguration). All these works propose methodologies for the implementation of AAS, including the integration of their assets. However, none of them provides guidelines to facilitate this integration; at most, they propose the use of some technologies for this purpose.

2.3.2 Discussion

Once the review of related work has been completed, and taking into account the questions on which this section focuses (Q4 and Q5), these will be evaluated according to the following requirements:

- First, they should be based on generic concepts applicable to any type of asset (R1, asset independence).
- Second, they should be, as far as possible, independent of any specific technology (R2, technology independence).
- Third, they should include a roadmap for carrying out the integration process (R3, asset integration methodology).

A solution that meets these requirements would help companies to integrate their manufacturing assets in Industry 4.0.

Table 2-1 shows the conformity of the works analyzed with respect to the requirements. As can be seen, some of these works meet two out of three requirements. Some works are too specific and address integration by focusing only on one type of asset or using a particular technology. Others, in contrast, present a generic approach but do not provide details on asset integration. This is not incompatible with the fact that some of the works reviewed present interesting concepts that may be useful for future solutions. For example, the use of internal interfaces to enable communication between the AAS and the asset (Ashtari Talkhestani et al., 2019; Leitão and Barbosa, 2016; Tantik and Anderl, 2017), that are generic and reusable for assets of the same type (Ye et al., 2020). However, as far as can be seen from this literature review, there is no integration proposal that meets all three requirements at the same time.

Table 2-1 Summary of compliance with integration requirements

| Integration proposal | R1: Asset independence | R2: Technology independence | R3: Asset integration methodology |
|--|------------------------|-----------------------------|-----------------------------------|
| (Ye et al., 2021), (Neal et al., 2021), (Cavaliere and Salafia, 2020), (de las Morenas et al., 2017) | ✘ | ✘ | ✘ |
| (Arm et al., 2021), (Contreras et al., 2017), (Fuchs et al., 2019), (Sakurada et al., 2022), (Leitão and Barbosa, 2016), (Bakliwal et al., 2018) | ✔ | ✘ | ✘ |
| (Dai et al., 2017) | ✘ | ✔ | ✘ |
| (Martin et al., 2019), (Schou and Madsen, 2017) | ✘ | ✘ | ✔ |
| (Ashtari Talkhestani et al., 2019), (Seif et al., 2019), (Tantik and Anderl, 2017), (Ye et al., 2020) | ✔ | ✔ | ✘ |
| (Leitão et al., 2018) | ✔ | ✘ | ✔ |

2.4 I4.0 System Management Platforms

Asset integration in Industry 4.0 through AASs enables interoperability between assets with different functional and communicative capabilities. This allows *I4.0 Components* to collaborate by offering and requesting services that support the execution of manufacturing applications (i.e., *Application Relevant Services*). However, as discussed in Subsection 2.2.1, additional infrastructure is required to support the creation, registration, deployment and management of *I4.0 Components* (i.e., *Infrastructure Services*). Although part of this infrastructure (the internal management of the information and functionalities of the *I4.0 Components*) can be assumed by the AASs themselves through the *AAS Services*, most of these services (creation, registration, exposure of the AASs of the system, among others) fall on *AAS Infrastructure Services*, offered by the so-called I4.0 Platforms (manufacturing management platforms in the context of Industry 4.0).

This section analyzes the literature regarding I4.0 Platforms. The objective is to check how they organize factory management, what services they offer and whether their solutions are fixed or facilitate their extension and customization according to user needs. The rest of this section has been divided into two parts: the first shows how the literature search has been conducted and reviews the selected works; the second

presents the criteria that will be used for the comparative analysis of works and discusses briefly how these works meet the established requirements.

2.4.1 Related work

This subsection shows how the search for works related to I4.0 Platforms has been conducted. As in the previous search, we have used the bibliographic database Web of Science. In this case, a search has been carried out using the concepts “Industry 4.0” or “Industrie 4.0” and also including the concepts “platform” or “architecture” (this distinction has been included because different authors use one term or the other indistinctly to refer to the same thing. In this PhD dissertation, the term platform will be used in accordance with *RAMI 4.0* terminology and to differentiate our work from the reference architectures.). However, the number of works obtained from this search is still too high, so it was decided to narrow it down by application paradigm the search to the implementation technology that best suits the needs of I4.0 Systems (interoperability, adaptability, flexibility, etc.).

As seen in Subsection 2.3.1, from a technology point of view, there are two alternatives that stand out above the rest when it comes to developing solutions for Industry 4.0: industrial agents and OPC UA. Industrial agents have an inherent capacity to make decisions autonomously, which allows them to decide to compete and/or collaborate with each other to achieve their objectives. The Agent Communication Language (*ACL*), a protocol developed by the Foundation for Intelligent Physical Agents (*FIPA*²), contemplates the use of ontologies, which facilitates the definition of different scenarios to ensure interoperability (“Standard Status Specifications,” n.d.). However, they can also fit other protocols and build well-known *APIs* such as REST. In addition, they also stand out for naturally addressing asset integration, which is fundamental in I4.0 Systems, as supported by the IEEE 2660.1-2020 standard (“IEEE Recommended Practice for Industrial Agents,” 2021). All these features make industrial agents ideal for the implementation of Industry 4.0 solutions (Fay et al., 2019; Karnouskos et al.,

² <http://www.fipa.org/>

2020; Vogel-Heuser et al., 2020). In comparison, OPC UA allows transferring different types of data structures with ease, but does not innately offer the autonomy and decision-making capabilities inherent to agents. With this in mind, it has been decided to focus primarily on agent-based work.

In many of these works, the management of the manufacturing process is based almost exclusively on two types of agents (usually called by this or similar names): Product Agent, in charge of managing one or more products, and Resource Agent, in charge of representing the assets in the factory. In (Bennulf et al., 2020), Product Agents manage process plans, understood as sequences of skills they require to fulfil their goals. These skills are offered by the Resource Agents, who fulfill their designated demands (specific, parameterized skill requests), collaborating with each other if necessary. PROSA architecture (Van Brussel et al., 1998) also proposes using Product Agents and Resource Agents, and incorporates Order Agents for handling scheduling and logistical aspects. This work also considers incorporating Staff Agents to supervise the system and assist the other agents with expert knowledge. ADACOR architecture (Barbosa et al., 2015) follows a similar approach and proposes a Product Agent, Task Agent and Operational Agent corresponding to the Product Agent, Order Agent and Resource Agent in PROSA, respectively. In addition, ADACOR includes a Supervision Agent that provides coordination and optimized scheduling capabilities. Another example is CASOA architecture (H. Tang et al., 2018), where there are four different agents: Product Agents, Machine Agents, Conveying Agents, in charge of managing the manufacturing process; and Suggestion Agents, whose function is to assign order tasks to Product Agents and subsequently suggest updates based on data processing in the cloud. Further examples can be found in (Kovalenko et al., 2022; D. Tang et al., 2018). All these works identify the basic entities present in a manufacturing system and define agents associated with them. Thus, the management of the system is based on the interactions between these agents for the supply and demand of services, generally with the Resource Agents satisfying the service requests required by the Product Agents to meet their goals. However, these works do not point to any additional infrastructure, or the one they have focuses exclusively on task scheduling. This makes

system management depend only on the partial system information that each agent has, with no way to check if it is out of date or incorrect.

Other works present solutions that go deeper into the functionalities that support the management of the manufacturing process, with services such as the creation and registration of components. (Peres et al., 2018) propose the IDARTS framework, which consists of several parts, including an agent-based manufacturing management system. This system has a data model that records, among other things, information about existing plant assets and their relevant information. This information is periodically checked by a set of deployment agents, which launch and remove agents as needed to match the plant topology. (Munkelt and Krockert, 2018) present a concept based on eight different types of agents that are divided into transient agents (linked to the execution of tasks) and persistent agents (alive during the whole process). In the second group there are two agents that provide infrastructure necessary to manage the system: the system agent (responsible for registering production plans and launching them by creating the necessary agents) and the directory agent (manages a directory of the agents available in the system). Similarly, MAS-RECON architecture (Gangoiti et al., 2022) proposes a set of agents that monitor the system and manage the creation and registration of agents associated with the resources and applications within the system. MAS-RECON does not focus on the manufacturing domain but provides mechanisms to customize the resources and applications of a specific domain by means of information models. These works provide essential functionalities and means to facilitate the management of the manufacturing process. However, they do not follow *RAMI 4.0* or any of the other reference architectures discussed in Section 2.2, which makes interoperability with other systems difficult.

Among the works that are aligned with *RAMI 4.0*, many focus on the concepts of *AAS* and *I4.0 Component* from different points of view. For example, (Sakurada and Leitão, 2020) focuses on the implementation of the *AAS*, identifying the following challenges: supporting interoperability, interconnecting *AASs* and assets, and providing distributed intelligence to *I4.0 Components*. Similarly, (Baumgärtel and Verbeet, 2020) propose the use of agents to implement the *AAS* due to their negotiation and decision

making characteristics, coining the term Active Component Shell. In turn, (Contreras et al., 2017) proposes an implementation of industrial agent-based AASs based on three types of agents (resources, products and coordinators), as well as which technologies to use to implement the different layers of the cubic model (OPC UA for the communication layer, IEC 62769 as reference for the information and functional layers, and AutomationML for end-to-end engineering). These works are interesting because they identify the features of *I4.0 Components* and which technologies fit with them, but they do not go into detail on how to develop *I4.0 Components* or what services they should offer.

(Cruz Salazar et al., 2019) propose design patterns (i.e., categories of agents with a specific name and functionality) to implement an agent-based architecture aligned with *RAMI 4.0*. Specifically, in this proposal the AAS is interpreted as a combination of agents that perform specific functions in the different layers of *RAMI 4.0*. This work is reviewed and extended in the MARIANNE architecture (Cruz Salazar and Vogel-Heuser, 2022), focused on providing a solution aligned with both *RAMI 4.0* and the main standards related to industrial agents (“IEEE Recommended Practice for Industrial Agents,” 2021; “VDI/VDE. 2021. 2653 Sheet 4: Multi-agent systems in industrial automation – Selected patterns for field level control and energy systems,” 2022). These works focus more on the type of components that should form the system than on the services to be offered by the components and the I4.0 Platform.

Other works make concrete proposals for implementing the AAS concept. (Arm et al., 2021) propose a procedure to facilitate the creation of AASs through a configuration wizard. In this proposal, two basic types of AASs are defined, service requesters and service providers, with specific functionalities. However, they share a common code that provides the ability to compete for service offerings and structured access to AAS data. Instead, (Cavalieri and Salafia, 2020) focus on modeling AASs to represent PLCs programmed according to IEC 61161-3. This work proposes to implement AASs as OPC UA nodes, accompanied by a web application that allows to perform AASs creation and update services manually. These works present implementations of AASs with specific services, together with a complementary infrastructure, designed to create or update

AASs manually. However, the infrastructure they offer does not contemplate an active and automated management of AASs.

Other works focus on developing I4.0 Platforms. The work presented in (Leitão et al., 2016) presents a platform for managing manufacturing reconfiguration. This platform is based on a network of heterogeneous components interconnected by a middleware that guarantees their interoperability. To this end, the middleware offers functionalities such as service registration and exposure, exception handling and data persistence, etc. (Trunzer et al., 2019) define the requirements for I4.0 Platforms based on the analysis of three research projects that approach the topic from different perspectives: data analytics and security, service discovery and real-time management. As a result, a platform is proposed that combines elements from each of the projects to meet all these requirements. These works corroborate that there is a real need to develop platforms for managing I4.0 Systems, and indicate which aspects to focus on. However, after identifying these needs, they do not indicate how to implement their platforms to address them.

There are other works that present specific proposals oriented to different types of services. (Pisching et al., 2018) propose a platform based on the *Layers* axis of *RAMI 4.0* that focuses on the discovery of manufacturing assets. To do so, two databases are proposed: Virtual Entity, which maintains a virtual representation of any physical asset in the system, and Hierarchical Equipment Network, which indicates the operations available in the system, which machine can perform them and with what priority. These priorities are used as a fixed criterion for allocating the operations to be done. (Ye et al., 2021) presents a platform comprising three layers: the lower one, where the plant assets are located, the middle one, where the AASs are deployed, and the upper one, in the cloud, where functionalities for managing the AASs are executed. In this approach, AASs are manually modeled and then transformed into OPC UA nodes that can exchange data with each other to perform application-specific functions. AASs are managed through two cloud applications: a web browser to view and edit AAS submodels and a user interface to monitor asset data acquired by AASs. These two works propose I4.0 Platforms to address the problems they aim to solve from different

perspectives. However, although they show implementation examples, they do not seem to provide guidelines from which to follow or replicate their proposal.

2.4.2 Discussion

Taking into account the questions related to this section (Q6–Q8), it is checked if the reviewed works meet the following criteria:

- First, if works do follow *RAMI 4.0* or not (R1, *RAMI 4.0* compliance).
- Second, if they are focused on the development of components, infrastructure, or both (R2, focus).
- Third, if they propose design patterns, and if so, if they are fixed or customizable (R3, design patterns).
- Fourth, if they provide infrastructure at component level (i.e., the *AAS Services* considered by *Plattform Industrie 4.0*), which shall include, at least, submodel registry, exposure and discovery, and meta information management (R4, component infrastructure).
- Fifth, if they provide infrastructure at platform level (i.e., the *AAS Infrastructure Services* considered by *Plattform Industrie 4.0*), which shall include, at least, component creation, registry and exposure and discovery (R5, platform infrastructure). Note that, in those works that were not aligned with *RAMI 4.0*, functionalities equivalent to *AAS Services* and *AAS Infrastructure Services* have been considered.

Some of the works reviewed in the previous subsection present concepts and ideas that can be very useful for the development of customizable I4.0 Platforms. For example, the distinction between transitory and permanent agents, as proposed by (Munkelt and Krockert, 2018) or de use of design patterns proposed by (Cruz Salazar et al., 2019; Cruz Salazar and Vogel-Heuser, 2022) are innovative measures that can increase the modularity and customization possibilities of I4.0 Platforms. In this regard, the contribution of MAS-RECON architecture (Gangoiti et al., 2022) is also noteworthy. MAS-RECON is not *RAMI 4.0* compliant, but the agent core it provides has the potential

to manage service-oriented components and provides customization mechanisms. However, as can be seen in Table 2-2, none of these works meet all the requirements due to at least one of the following reasons:

- They do not offer some or many of the services indicated.
- They are not aligned with *RAMI 4.0*.
- They do not offer customization mechanisms.

Table 2-2 Summary of compliance with requirements.

| Proposal | R1: RAMI 4.0 compliance | R2: Focus | R3: Design patterns | R4: Component infrastructure | | | R5: Platform infrastructure | | |
|---------------------------------------|-------------------------|-----------|---------------------|------------------------------|------------------------|-----------------------------|-----------------------------|----------|------------------------|
| | | | | Registry | Exposure and Discovery | Meta information management | Creation | Registry | Exposure and Discovery |
| (Bennulf et al., 2020) | No | C | F | Ms | Ms | Yes | No | No | No |
| (H. Tang et al., 2018) | No | C+I | F | Yes | Yes | Yes | No | No | Yes |
| (Munkelt and Krockert, 2018) | No | C+I | F | Yes | Yes | Yes | Yes | Yes | Yes |
| (Peres et al., 2018) | | | | | | | | | |
| (Gangoiti et al., 2022) | No | C+I | Cu | Yes | Yes | Yes | Yes | Yes | Yes |
| (Sakurada and Leitão, 2020) | Yes | C | U | No | No | No | No | No | No |
| (Baumgärtel and Verbeet, 2020) | | | | | | | | | |
| (Contreras et al., 2017) | Yes | C | F | No | No | No | No | No | No |
| (Arm et al., 2021) | Yes | C | F | Yes | Yes | Yes | Ms | No | No |
| (Cavaliere and Salafia, 2020) | Yes | C | U | Mp | Mp | Yes | Ms | No | No |
| (Leitão et al., 2016) | Yes | C+I | U | No | No | No | No | No | No |
| (Trunzer et al., 2019) | | | | | | | | | |
| (Pisching et al., 2018) | Partially | I | F | No | No | Yes | No | Yes | Yes |
| (Ye et al., 2021) | Yes | C+I | U | Mp | Mp | Yes | Ms | Yes | Yes |
| (Cruz Salazar et al., 2019) | Yes | C+I | F | Yes | Yes | Yes | Ms | No | Yes |
| (Cruz Salazar and Vogel-Heuser, 2022) | | | | | | | | | |

C = Focus on components, I = Focus on infrastructure, C+I = Focus on components and infrastructure; F = Fixed, Cu = Customizable, U = Unespefied; Ms = Manually, before startup; Mp = Manually, during process

2.5 Summary and conclusions

In this chapter, a comprehensive review of the Industry 4.0 problem has been carried out from different perspectives to check the answers available in the literature to the questions posed in Chapter 1, reaching the following conclusions:

- After comparing the main reference architectures for Industry 4.0, it was found that *RAMI 4.0* is the most relevant one (Q1). *RAMI 4.0* has received great attention from the scientific community, standing out from other reference architectures thanks to its age, its industrial perspective, proposing concepts such as the *I4.0 Component* and incorporating standards such as IEC 62264 and IEC 61512, and the support provided by *Plattform Industrie 4.0* in the form of several documents (Q2). In addition, it has been verified that there is an active collaboration between *Plattform Industrie 4.0* and the *IIC*, the institution behind *IIRA*, due to the complementary nature of both reference architectures (Q3). However, *RAMI 4.0* and *IIRA* provide just a starting point, based on standards and key concepts such as *AAS*, but they lack a technological base from which to design and develop customized I4.0 Systems. This lack of tools and mechanisms for the development of I4.0 solutions is slowing down their transfer to companies (Q9).
- The integration of manufacturing 4.0 assets has been identified as another major constraint in the transition to Industry 4.0 (Q9). After analyzing how this problem is addressed in the literature, it has been found that there are many papers devoted to the integration of assets in Industry 4.0 from different perspectives (Q4). However, most of them present an ad-hoc perspective and focus on specific asset types or technologies, which hinders the scalability of these solutions at the factory level. On the other hand, those papers that approach the problem from a more general perspective are limited to identifying the problems inherent to the integration process, without providing a methodology to guide the steps to follow to carry out the process (Q5).
- With respect to the management of I4.0 Systems, it has been determined that, although the I4.0 Components are the ones that offer the services required for

the execution of manufacturing applications, they require an I4.0 Platform that provides the necessary support to guarantee the correct operation of the system (Q6). This support is provided through the *Infrastructure Services*, which are responsible for managing the creation, registration, editing and exposure of the AASs in the system, among others (Q7). In addition, it has been found that the industrial agents have several qualities (negotiation capacity, autonomous decision making, use of a language that incorporates ontologies, etc.) that make them suitable for the implementation of both the *I4.0 Components* and the I4.0 Platform (Q8). As seen in the literature review, it is still common to find manufacturing management platforms based on industrial agents that are not aligned with *RAMI 4.0* or other reference architectures. Regarding those works that do adhere to *RAMI 4.0*, most of them cover only partially the infrastructure services identified as necessary by *Plattform Industrie 4.0*. In addition, they propose predefined and unchangeable types of components for their systems. Incorporating customization mechanisms on a functional basis to these solutions would facilitate their use by professionals (Q10).

For these reasons, the main goal of this work is to develop an I4.0 Platform that is based on the reference architectures; that develops its axes through the use of the appropriate technologies, justifying their choice and evaluating its performance in a reproducible way; and that also includes the methodological and technological resources that allow the implementation of customized solutions from a solid base, avoiding stages in the process of adoption of Industry 4.0 by the industry. To this end, the following chapters present the steps in which this goal has been developed:

- Chapter 3 focuses on providing methodological support to facilitate the design of Industry 4.0 solutions aligned with *RAMI 4.0*. To this end, it presents a modeling proposal based on the standards supporting the *hierarchy levels* axis of *RAMI 4.0*. This modeling allows defining I4.0 Systems, as well as the relationships established between their participants. Next, a generic multi-layer approach for asset integration is proposed. This proposal includes a methodology to facilitate the decision-making on the implementation and

deployment of the layers. This chapter also presents a proposal for the development of AASs in two stages oriented to the two service categories offered by AASs: *AAS Services* (i.e., *Infrastructure Services* for the internal management of the AAS) and *Application Relevant Services*.

- Chapter 4 addresses the design and development of an I4.0 Platform based on industrial agents for managing manufacturing systems aligned with *RAMI 4.0*. This platform provides a set of agents for implementing the AASs in the system based on patterns that can be extended and customized according to application requirements. In addition, the platform also provides a core that implements the *AAS Infrastructure Services* for AAS creation and registration, and provides additional means to ensure the operative state of the factory.

3 INDUSTRIA 4.0RAKO EUSKARRI METODOLOGIKOA

3.1 Sarrera

Kapitulu honen helburua, enpresei eta profesionalei *RAMI 4.0* erreferentzia-arkitekturarekin bat datozen Industria 4.0rako soluzioen diseinuarentzako eta garapenarentzako gida ematean datza. Horretarako, *RAMI 4.0n* bildutako estandarretan eta kontzeptuetan oinarritutako ikuspegi metodologikoak aurkezten dira, Industria 4.0ren hasierako faseetako hiru alderdi nagusiak jorratuz: aktiboen karakterizazioa, aktiboen integrazioa eta *AASen* garapena. 3.2 atalak aktiboen karakterizazioa aztertzen du. Horretarako, IEC 62264 (“IEC 62264-1,” 2013) eta IEC 61512 (“IEC 61512-1,” 1997) estandarren analisia egiten da, *RAMI 4.0ren Hierarchy Levels* ardatza arautzen dutenak, alegia. Analisi horren ondorioz, I4.0 Sistemetak parte-hartzaileen (hau da, fabrikazio-aktiboak) ezaugarriak definitzeko eredu-egite ikuspegi bat aurkezten da. 3.3 atala aktiboen integrazioari buruzkoa da. Aktiboak integratzeko prozesuaren alderdi nagusiak abstraitzeko, geruza independenteetan oinarritutako geruza anitzeko ikuspegia proposatzen da, arazoa zeregin txikiago eta kudeagarriagoetara murriztuz. Proposamen hori integrazio metodologia generiko batekin osatzen da. Metodologia horrek urrats batzuk proposatzen ditu, aktibo mota eta integratzeko beharrezkoa den teknologia edozein direla ere aplikatu daitezkeenak. 3.4 atala *AASen* garapenari dagokio. Horretarako, *AASek* eman beharreko zerbitzu moten ezaugarrien azterketatik abiatuz, *IIRA* kontzeptuetan oinarritutako bi etapako proposamena eratortzen da. 3.5 atalak kapituluaren ondorioak aurkezten ditu.

3.2 Aktiboen karakterizazioa

Atal honetan, fabrikazio-prozesuan zehar I4.0 Sistemen parte diren aktiboen ezaugarriak zehazteko eredu-egite proposamen bat aurkezten da. Zehazki, *RAMI 4.0n* bildutako estandarrekin eta terminologiarekin bat datozen meta-ereduak proposatzen dira fabrikazio-sistema baten aktiboak deskribatzeko. Ikuspegi horrek IEC 62264 eta IEC 61512 estandarren gidei jarraitzen die (horiek baitira *Hierarchy Levels* ardatzaren erreferentzia), eta *Life Cycle & Value Stream* ardatzaren ekoizpen-fasean kokatzen da.

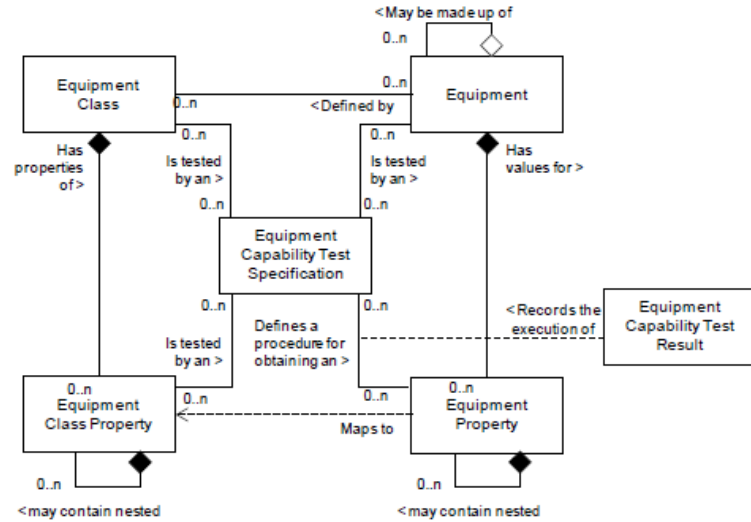
Hurrengo azpiatalean, IEC 62264 eta IEC 61512 estandarretan bildutako kontzeptu garrantzitsuenen azterketa labur bat egiten da, ondoren eredu-egite proposamena azalduz.

3.2.1 *Hierarchy Level* ardatzaren estandarrak: IEC 62264 eta IEC 61512

Azpiatal hau *RAMI 4.0* hierarkia-mailak gidatzen dituzten estandarretan oinarritzen da, produkzio-fasean *I4.0 Component* osagaien sistemako aktiboak karakterizatzeko behar den informazioa identifikatuz. Informazio horri esker, I4.0 Plataformak erabakiak hartu ahal izango ditu fabrikazio-planean akatsik edo aldaketarik izanez gero. Horretarako, I4.0 Plataformak fabrika nola antolatzen den (fabrikan erabilgarri dauden baliabideak), zer egin behar den (fabrikazio plana) eta denbora errealean zer gertatzen ari den jakin behar du. IEC 62264 eta IEC 61512 estandarrek lehenengo bi galderei erantzuten diete.

Lehenik eta behin, fabrika batean zer baliabide mota dauden eta nola antolatuta dauden identifikatu da. IEC 62264k bost baliabide mota bereizten ditu: *personnel*, *equipment*, *physical asset*, *material*, eta *process segment*. Azpimarratu behar da *equipment* eta *physical asset* baliabideen arteko erlazioa, lehena motarekin bat datorrena eta bigarrena instantziarekin (adibidez, *equipment* funtzionalitate espezifikoak dituen makina mota bati buruzkoa izan daiteke, eta *physical asset*, berriz, serie-zenbaki bakarra duen makina jakin bati buruzkoa). Baliabide horien informazio-ereduak IEC 62264 estandarren 2 zatian bilduta daude. Eredu-egite proposamen horrek bi baliabide mota erabiltzen ditu deskribatutakoen artean: *equipment* eta *material*. Horrela, estandarrean proposatutako *role based equipment model* ereduaren arabera (Irudia 3-1 irudia), *equipment* baliabideen ezaugarriak identifikadorea, deskribapena eta maila hierarkikoa dira (Taula 3-1 taula). *Material* baliabideei dagokienez (lehengaiak, kontsumigarriak, amaitu gabeko produktuak eta produktu bukatuak), *material model* ereduan (Irudia 3-2 irudia) identifikadore batez, deskribapen batez eta, aukeran, muntaketa mota batez (fisikoa edo logikoa) eta

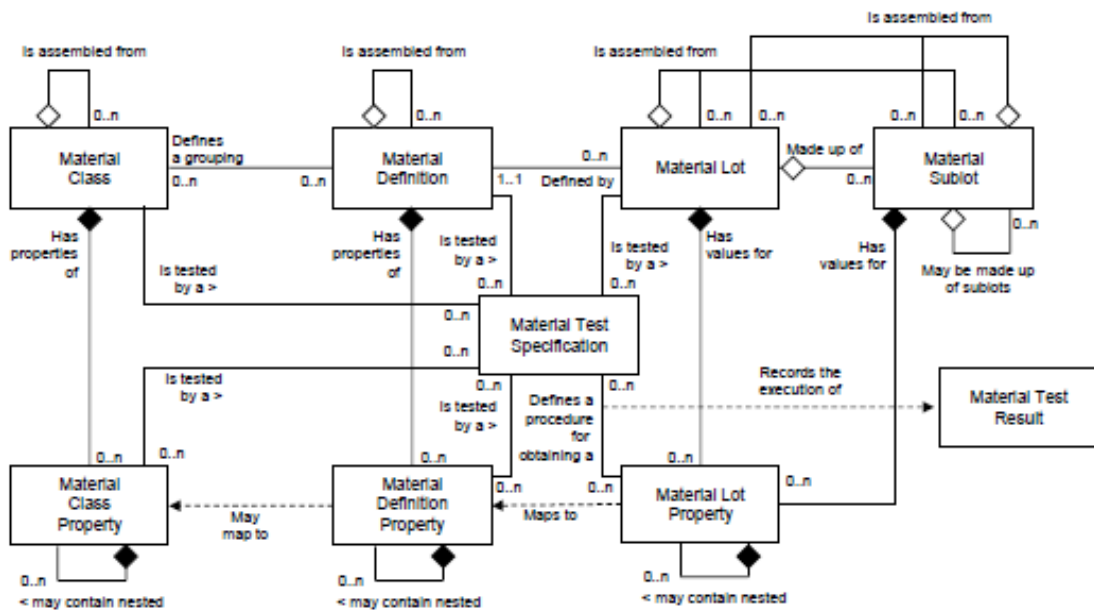
muntaketa erlazio batez (iraunkorra edo iragankorra) bereizten dira, Taula 3-2 taulan adierazten den bezala.



Irudia 3-1 Role based equipment model eredua. Egokitzapena (“IEC 62264-2,” 2013, Figure 6).

Taula 3-1 Equipment-baliabideen ezaugarriak. Egokitzapena (“IEC 62264-2,” 2013, Table 11).

| Ezaugarriaren izena | Deskribapena |
|---------------------|--|
| Identifikadorea | Ekipamendu-pieza jakin baten izendapen bakarra. |
| Deskribapena | Ekipamenduari buruzko informazio gehigarria. |
| Ekipamendu-maila | Roletan oinarritutako ekipoen hierarkiaren maila identifikatzea. |

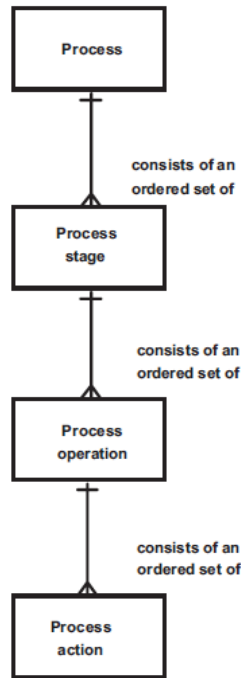


Irudia 3-2 Material model eredua. Egokitzapena (“IEC 62264-2,” 2013, Figure 9).

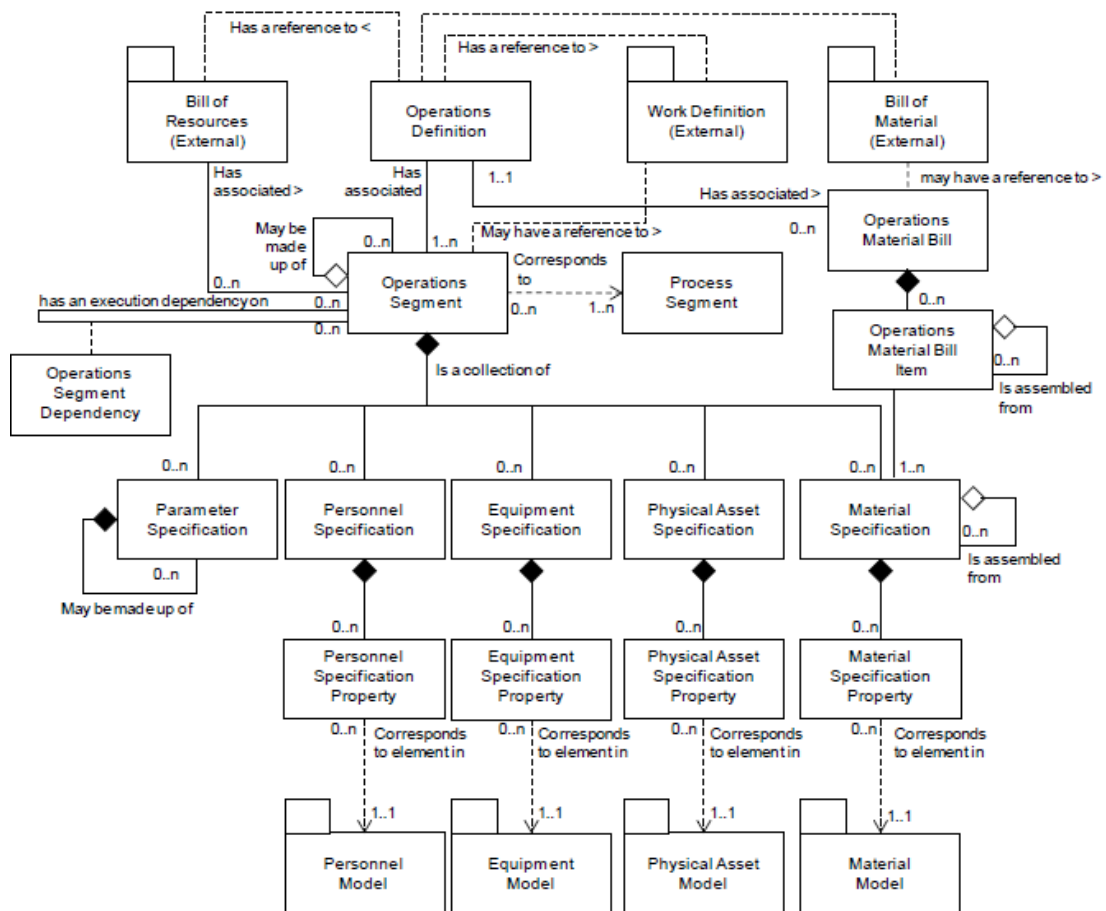
Taula 3-2 Material-baliabideen ezaugarriak. Egokitzapena (“IEC 62264-2,” 2013, Table 24).

| Ezaugarriaren izena | Deskribapena |
|---------------------|---|
| Identifikadorea | Material mota jakin baten izendapen bakarra. |
| Deskribapena | Material klaseari buruzko informazio gehigarria. |
| Muntaketa mota | Aukerakoa: Muntai mota definitzen du, hots, osagaiak fisikoki edo logikoki lotuta dauden. |
| Muntaketa erlazioa | Aukerakoa: Muntaiaren erlazio mota definitzen du, hots, iraunkorra edo iragankorra den. |

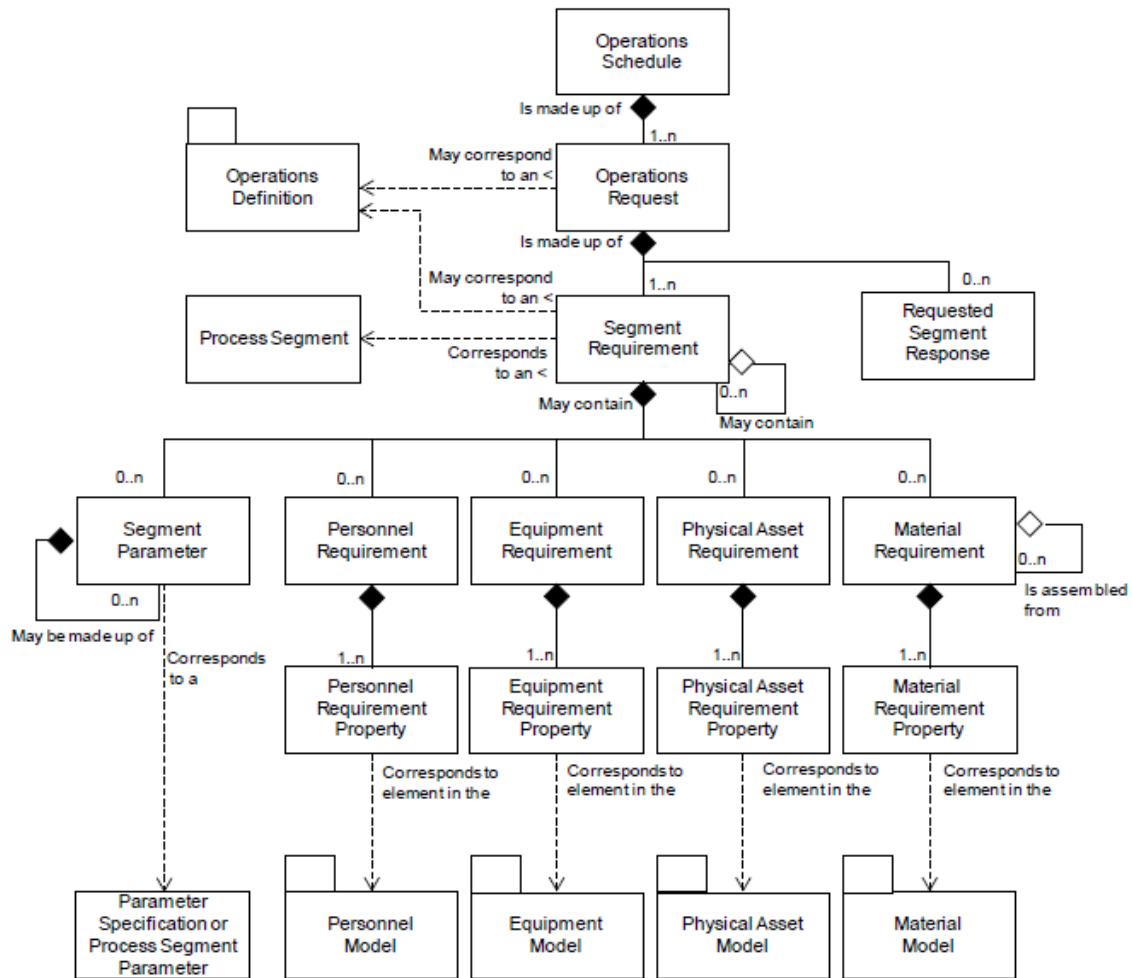
Fabrikazio planari dagokionez, zer, noiz eta non ekoitzi behar den jakin behar da. Horrek produktu zehatzei eta fabrikazio planari eragiten die. Produktua bere fabrikazio-prozesuan parte hartzen duten eragiketekin erlazionatuta dago. Horri dagokionez, IEC 61512 estandarraren 1 zatiak *process model* eredu ematen du, sorta-prozesuak granularitate-maila batean edo gehiagotan azpibanatzeko aukera ematen duena (Irudia 3-3 irudia). Fabrikazio planak ondorengo informazioa biltzen du: eragiketa zein makinak egin behar duen, zein produktuentzat eta zein ordenatan. Estandarrak bi alderdi desberdin deskribatzen ditu: eragiketa baten definizioa (*operations definition model*, Irudia 3-4 irudia) eta bere programazioa (*operations schedule model*, Irudia 3-5 irudia). IEC 62264 estandarraren 2 zatiak *operations definition* objektua ezaugarritzen du, besteak beste, identifikadore batekin, deskribapen batekin, identifikadore operazional batekin (kasu honetan, bere balioa *production* izango da beti) eta informazioa trukatzeko deneko hierarkia-mailaren erreferentzia batekin. *Operations schedule* objektuak ezaugarri horiek ditu, eta beste batzuk ere, hala nola argitalpen-data eta programatutako egoera (pausagunean edo iragarrita). Taula 3-3 taulan eredu-egite ikuspegiarako erabiliko diren *operations schedule* objektuaren ezaugarriak adierazten dira.



Irudia 3-3 *Process model* eredua. Egokitzapena (“IEC 61512-1,” 1997, Figure 1).



Irudia 3-4 *Operations definition model* eredua. Egokitzapena (“IEC 62264-2,” 2013, Figure 13).



Irudia 3-5 *Operations definition model* eredua. Egokitzapena (“IEC 62264-2,” 2013, Figure 14).

Taula 3-3 *Operations schedule* objektuaren ezaugarriak. Egokitzapena (“IEC 62264-2,” 2013, Table 58).

| Ezaugarriaren izena | Deskribapena |
|-----------------------|--|
| Identifikadorea | Elementuaren identifikazio bakarra. |
| Deskribapena | Elementuaren informazio eta deskribapen gehigarriak. |
| Eragiketa mota | Eragiketaren kategoria deskribatzen du. Definitutako balioak <i>Production, Maintenance, Quality, Inventory</i> eta <i>Mixed</i> dira. |
| Hasiera-denbora | <i>Operations schedule</i> hasiera-denbora. |
| Amaiera-denbora | <i>Operations schedule</i> amaiera-denbora. |
| Argitalpen-data | <i>Operations schedule</i> objektua argitaratu edo sortu zen eguna eta ordua. |
| Hierarkia-eremua | Trukatutako informazioa roletan oinarritutako ekipoen hierarkiaren barruan kokatzen du. |
| Programatutako egoera | <i>Operations schedule</i> egoera adierazten du. Egoera hau <i>forecast</i> (iragarrita) edo <i>released</i> (pausagunean) izan daiteke. |

3.2.2 Ereduegite ikuspegia

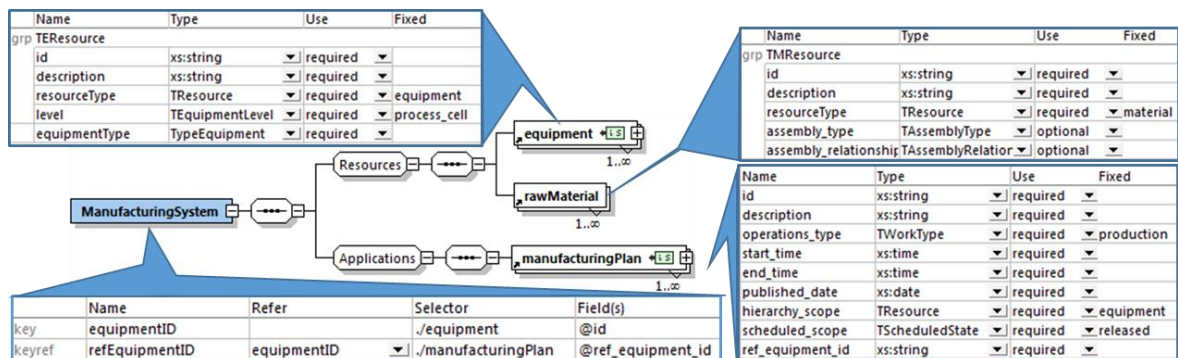
Lehen egindako analisisian oinarrituta, azpiatal honetan Industria 4.0rako ereduegite proposamen bat aurkezten da.

Fabrikazio-sistemaren ereduegitea bi meta-eredu nagusitan banatzen da: *resources* eta *applications*. *Resources* meta-eredua bat dator IEC 62264 eta IEC 61512 estandarretan deskribatutako baliabide-kategorien arabera fabrikatzeko moduarekin. *Applications* meta-ereduek ekoiztu nahi dena definitzen duten fabrikazio plan posible guztiak biltzen ditu. Eredue horrek estandarretatik hautatutako informazio-ereduak zabaltzen ditu, eskatutako malgutasuna lortzeko ezinbestekoa den informazio dinamikoa gehituz. *Manufacturing system*aren meta-eredua Irudia 3-6 irudian agertzen da. Irudi hau eta atal honetan aurkezten diren guztiak XMLSpy³ ereduegite tresnaren bidez sortu dira.

Jarraian, *Resources* eta *Applications* estandarretara nola egokitzen diren zehazten da.

3.2.2.1 Resources

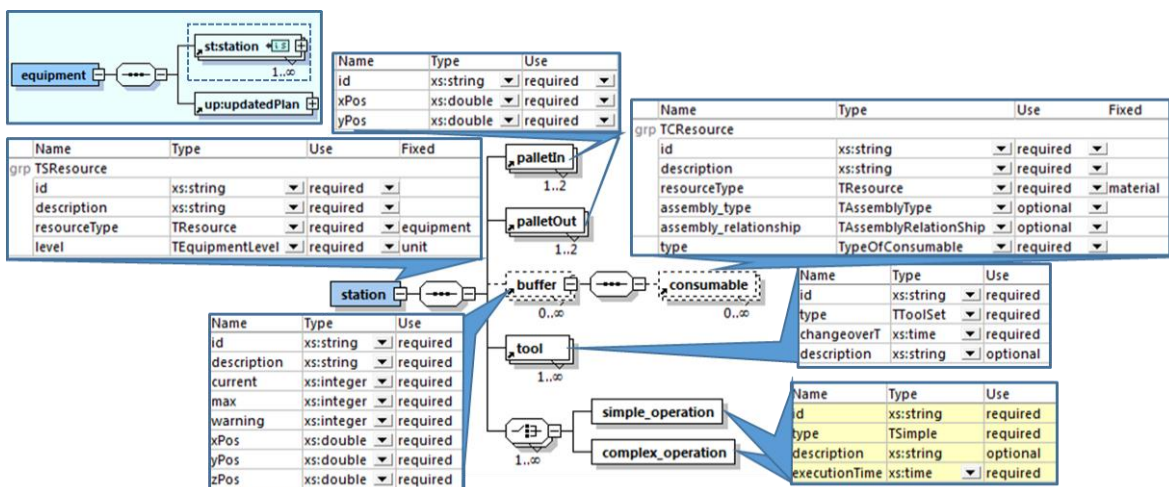
3.2.1 azpiatalean egindako analisiaren ildotik, proposamen honetan fabrikazio baliabide (*resources*) gisa hartzen dira produktuak fabrikatzeko erabiltzen diren lehengaiak (*raw material*) eta fabrika dagoen ekipamendua (*equipment*).



Irudia 3-6 Manufacturing system meta-eredua (XMLSpyrekin sortua).

³ <https://www.altova.com/xmlspy-xml-editor>

- equipment*: role based equipment model ereduaren arabera, *equipment* motako balibideak errekursiboak izan daitezke (hau da, *equipment* motako elementuak beste *equipment* batzuez osaturik egon daitezke). Horrela, azpi-produktu desberdinetan eragiketa paraleloak egiteko gaitasuna irudikatzeko, bi maila definitu dira: *equipment RAMI 4.0* hierarkiaren *work center* mailan dago, eta *station*, berriz, *station* mailan. *Equipment* motako elementu bakoitza *station* elementu azpimultzo batek osatzen du, eta, beraz, paraleloan eragiketa multzo bat egin daiteke. Horrela, *equipment* batek bere *station*en fabrikazio-eragiketak ditu ezaugarri. Eragiketa bakoitzak, berriz, bere iraupena du ezaugarri. Gainera, *station* bakoitzak eragiketak egiteko behar den material kontsumigarria eta eskura dagoen tresneria ditu ezaugarri. Dagokion meta-eredua Irudia 3-7 irudian adierazita dago. Bi *equipment* mota hartu dira kontuan (*equipmentType* ezaugarria Irudia 3-6 irudian): *machine* eta *transport*. *Transport* unitateek ez dute fabrikazio-eragiketarik egiten; aldiz, *material* balibideak *machine*-en artean dinamikoki garraiatzeko aukera ematen dute, sistemak ustekabeko egoerei erantzuteko beharrezko duen malgutasuna gehituz. Hala ere, *equipment* mota horiek zabaldu eta/edo aldatu egin daitezke, egoki iritziz gero (adibidez, *machine* mota desberdinak bereizteko, (Kovalenko et al., 2022) edo (D. Tang et al., 2018) ikus daitezkeen moduan).
- rawMaterial*: elementu honek eragiketak egiteko erabiltzen den lehengaiaren zati bat ezaugarritzen du (ikus Irudia 3-6 irudia).

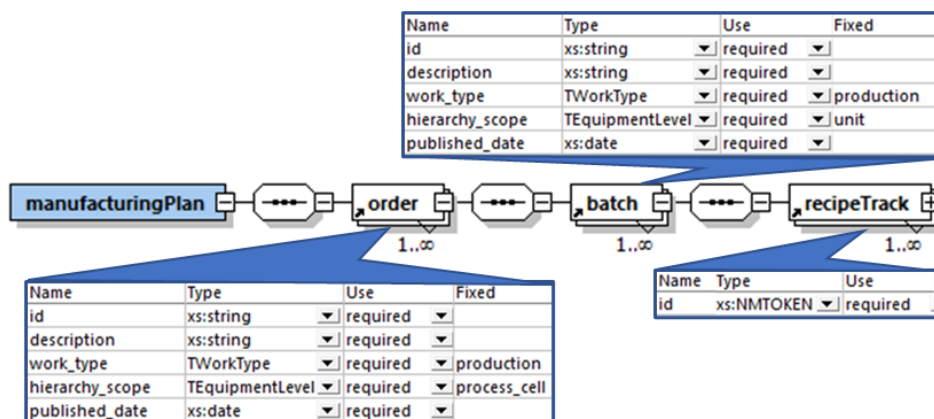


Irudia 3-7 Equipment meta-eredua (XMLSpyekin sortua).

3.2.2.2 Applications

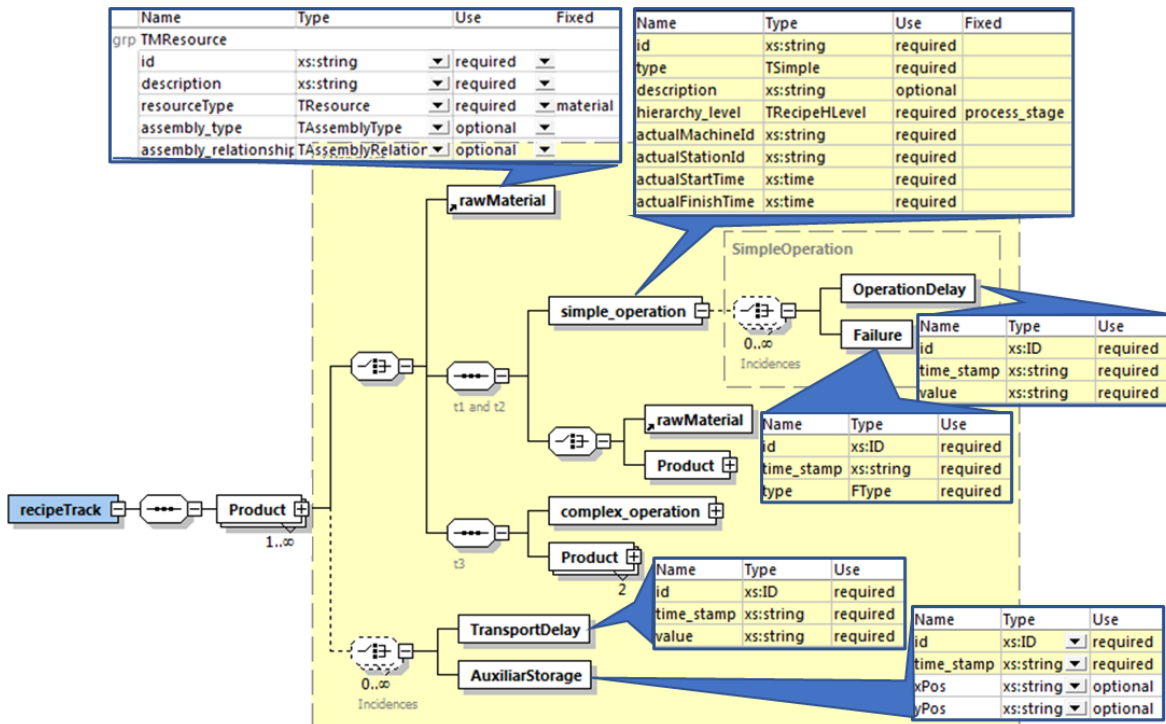
I4.0 Sistemetan, *applications* elementuek zer ekoitzi behar duten zehaztu behar dute, eskatutako *equipment* eta *material* baliabideei eta produktu bakoitzaren plangintzari buruzko informazioa emanaz. Proposamen honetan, *applications* meta-eredua hiru mailatan egituratzen dira: *manufacturing plan* elementu batek fabrika-mailako produkzioari buruzko informazio guztia gainbegiratzen du. *Manufacturing plan* elementu bat *order* elementu batek edo gehiagok osatzen dute, eta horiek prozesuei buruzko informazioa bezero-mailan erregistratzen dute (hau da, helburu bera partekatzen duten produktuak elkarrekin monitorizatzen ditu). Azkenik, *order* elementuak *batch* elementu multzo bat osatzen du, fabrikatu beharreko produktu jakin baten elementu multzo bat adierazten duena. Irudia 3-8 irudiak *manufacturing plan*aren meta-eredua erakusten du.

IEC 62264 eta IEC 61512 estandarrek emandako ereduak ez dute fabrikazio-sistemaren egoera une oro irudikatzeko ezaugarririk. Horretarako, proposamen honek *recipeTrack* elementua barne hartzen du, estandarraren *process model* ereduan oinarrituta (kasu honetan, lehengaiak, azpiproduktuak eta amaierako produktuak aipatzeko), bai eta *operations definition model* eta *operations schedule model* ereduetan ere. *RecipeTrack* elementuak trazabilitatea eskaintzen du; izan ere, uneko fabrikazio-prozesuaren bilakaera jasotzen du (hau da, produktuari eragiketak noiz eta non egiten zaizkion), ustekaberik gertatuz gero hasierako plangintzarekiko aldaketak izan ditzakeena (ikus Irudia 3-9 irudia).

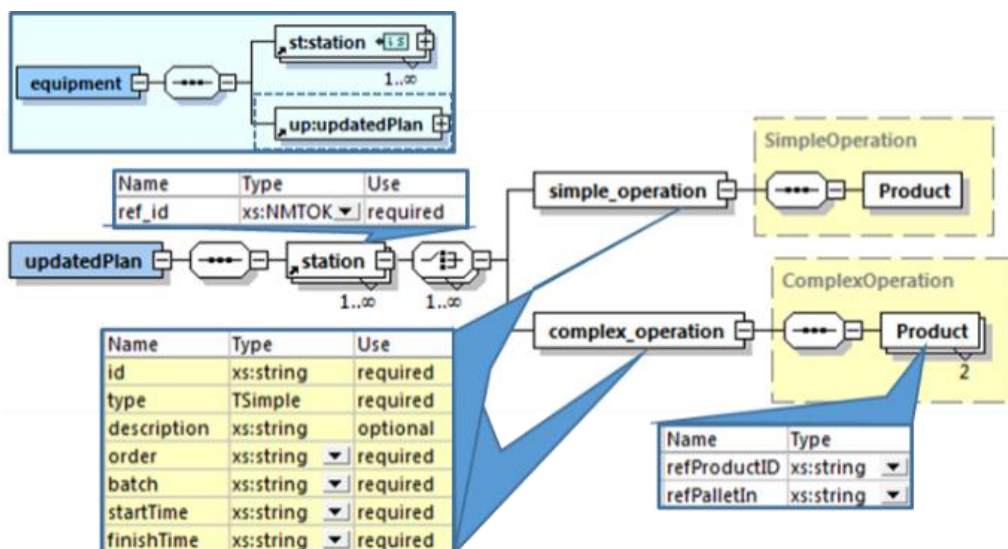


Irudia 3-8 *Manufacturing plan* meta-eredua (XMLSpyrekin sortua).

Equipment elementuak bere plana ere jasotzen du, hau da, stations baliabideek egin beharreko eragiketa aurreikusien sekuentzia, updatedPlan elementuan gordeta (ikusi Irudia 3-10 irudia). Hasiera batean plan estatikoa jasotzen duen arren, aldaketak izan ditzake fabrikazio sistemaren uneko egoerara egokitzeko.



Irudia 3-9 Recipe track meta-eredua (XMLSpyrekin sortua).



Irudia 3-10 Updated plan meta-eredua (XMLSpyrekin sortua).

3.3 Aktiboen integrazioa

I4.0 Sistemen eta horien parte-hartzaileen ezaugarriak zehaztu ondoren, atal honek fabrikazio-aktiboak Industria 4.0n zelan integratu daitezkeen azaltzea du ardatz. Horregatik, 3.3.1 azpiatalak aktiboak *I4.0 Component* osagaietan integratzeko geruza anitzeko ikuspegia aurkezten du. Ikuspegi horrek aktiboen integrazioa erraztea eta *I4.0 Components* osagaiei *Asset Related Services* (hau da, aktiboari lotutako funtzionalitateak ematen dituzten *Application Relevant Services* zerbitzuak) zerbitzuetarako sarbidea ematea ditu helburu. Horretarako, aktiboen integrazioarekin erlazionatutako alderdi desberdinak geruzetan abstraitu dira, modu generikoan (hau da, integratu beharreko aktibo mota eta behar diren teknologiak alde batera utzita) zein independentean (hau da, elkarrengandik bananduta) jorratuak izan daitezten. Geruza anitzeko ikuspegiarekin batera, 3.3.2 azpiatalak integrazio-metodologia bat proposatzen du, geruza bakoitza garatzeko jarraibideak ematen dituena, erabiltzaileen beharretara egokitutako soluzio bat lortzeko asmoz.

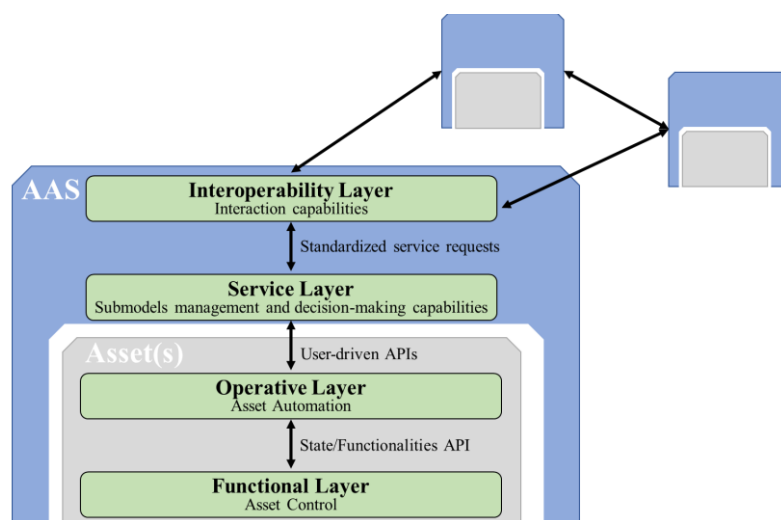
3.3.1 Geruza anitzeko ikuspegia

Industria 4.0n aktiboak integratzea zeregin konplexua da, besteak beste, aktiboari lotutako informazioaren kudeaketa, aktiboen automatizazioa edo zerbitzuen ezarpena barne hartzen dituena. Hala ere, alderdi horiek behar bezala identifikatuta eta bereizita badaude, modu generikoan aurre egin ahal izango zaie, edozein motatako aktibori aplikatzeko gaitasuna izanez, haien komunikazio-gaitasunak edozein direla ere.

Azpiatal honek Industria 4.0n aktiboak integratzeko geruza anitzeko ikuspegi bat aurkezten du, alderdi horiek geruza abstraktutan isolatzen dituena. Balizko fabrika-aktiboen barruan, aktibo fisikoen integrazioak (adibidez, mekanizazio-estazioak edo garraio-robotak) aktibo logikoen integrazioak baino erronka handiagoa suposatzen du (adibidez, aktibo ukiezinak, hala nola trazabilitate-prozesuari buruzko informazioa duten datu-baseak edo enpresaren ezagutza), fabrika baten instalazioetan ohikoa baita aktibo fisiko legatuak edo komunikazio gaitasun desberdinak dituzten aktibo fisikoak aurkitzea. Horregatik, ikuspegi hau aktibo fisikoen integrazioan oinarritzen da.

Horrela, hemendik aurrera kontrakoa esaten ez bada, "aktibo" terminoa aktibo fisikoei erreferentzia egiteko erabiliko da soilik. AAS motari dagokionez, geruza anitzeko ikuspegi horrek aktiboen eta *proactive AAS*en integrazioan oinarritzen da, Industria 4.0 kontzeptua garatzeko potentzialik handiena aurkezten baitute. Gainera, aktiboak AAS mota horrekin integratzera bideratutako soluzioa sinplifikatu daiteke, *reactive AAS*ak edo *passive AAS*ak ere integratzeko aukera eskainiz.

Faktore horiek kontuan hartuta, ondoriozko proposamena, 3-11 irudian adierazita dagoena, lau geruzatan egituratuta dago. Goiko geruzak (*Interoperability Layer*) beste *I4.0 Component* osagai batzuekin elkarrengaitzeko behar diren komunikazio-gaitasunak ezartzen ditu zerbitzu-eskaera estandarizatuak trukatzuz. Bigarren geruzak (*Service Layer*) zerbitzu-eskaera horiek kudeatzen ditu, eta, hala badagokio, erantzun edo baztertu egiten ditu, eta aktiboekin elkarrengaitzen du eskaerek *Asset Related Services* zerbitzuei dei egiten dietenean. Hirugarren geruzak (*Operative Layer*) *Asset Related Services* zerbitzuak osatzen dituzten funtzionalitateen exekuzio ordenatua gainbegiratzen eta kudeatzen du. Geruza horrek *API*etan antolatutako metodoak ere eskaintzen dizkio *Service Layer* geruzari, kasu bakoitzean dagokionaren arabera. Azkenik, beheko geruzak (*Functional Layer*) *Asset Related Services* zerbitzuak osatzen dituzten funtzionalitateak ezartzen ditu eta aktiboaren egoera kudeatzen du. Funtzionalitate horiek *Operative Layer* geruzari beste *API* baten bitartez ematen zaizkio. Jarraian, geruza hauek xehetasunez deskribatuko ditugu.



Irudia 3-11 Aktibo fisikoak Industria 4.0n integratzeko geruza anitzeko ikuspegia.

3.3.1.1 *Interoperability Layer geruza*

Geruza honek *I4.0 Component* osagaiaren kanpoko *APIa* ezartzen du, sistemaren gainerako *I4.0 Component* osagaiekin elkarreragiteko eta *Application Relevant Services* zerbitzu eskaerak kudeatzeko erabiltzen dena. *I4.0 Component* osagaien arteko elkarreragite egokirako, ezinbestekoa da zerbitzu-eskaera horiek alde aurretik zehaztutako formatu jakin bat jarraitzea. 3-12 irudian ikus daitekeen bezala, lan honek zerbitzuaren informazioa bi kategoriatan banatzea proposatzen du:

- *Meta information*: zerbitzu-eskaera baten exekuzioan eraginik ez duten, baina bere kudeaketarako beharrezkoak diren datuez osaturik dago. Honako hauei buruzko informazioa jaso behar du: zerbitzu-eskatzailea (nori erantzun behar dion jakiteko), honek sisteman duen rola (zerbitzurako atzipen-baimena duela egiaztatzeko) eta zerbitzu mota (nola kudeatu behar den zehazteko).
- *Core information*: zerbitzua egiteko behar diren datuak dira. Eskatutako zerbitzua identifikatzeko eta parametrizatzeko aukera eman behar du.

Datu-eremu horien artean, zerbitzu-eskatzailearen identifikadorea da *Interoperability Layer* geruzarentzako adierazgarriena, jasotako zerbitzu-eskaeretan parte hartzen duten hizketakideak ezagutzeko. Era berean, mezuren bat bidaltzean, geruza horrek *AASa* eta berari lotutako aktiboa identifikatzeko informazioa bete behar du, beste *I4.0 Component* osagai batzuek ezagutu dezaten. Zerbitzu-eskaeran jasotako gainerako informazioa *Service Layer* geruzari bidaltzen zaio, *interoperability-service* interfazearen bidez.

3.3.1.2 *Service Layer geruza*

Geruza hau *Interoperability Layer* geruzatik jasotako zerbitzu-eskaerak prozesatzeaz eta *Application Relevant Services* zerbitzuetarako atzipena kudeatzeaz arduratzen da. Beraz, geruza *Service Layer* geruzan *Component Manager*-a ezartzen du, *Application Relevant Services* zerbitzuei erreferentzia egiten dieten *AAS* azpi-ereduak kudeatzeko (kontuan izan behar da, geruza honetan edozein zerbitzuren exekuzioa *AAS* azpi-ereduen informaziora abstraituta dagoela).

(Requester Id, Requester role, Service type, Service Id, Input parameters)

Meta information

Core information

Irudia 3-12 I4.0 Component osagaien artean zerbitzu-eskaeretan sartu beharreko informazioa.

Kudeaketa hori egiteko, zerbitzu motari buruzko informazioa bereziki garrantzitsua da. Azpi-ereduen (*Submodel Service*) erabilpena eskatzen duten zerbitzu-eskaerak zuzenean *Service Layer* geruzan ebatzi ahal izango da. Aldiz, aktiboarekin elkarreragitea (*Asset Related Service*) eskatzen duten zerbitzu-eskaerentzako, zerbitzuaren *core information* datuak *Operative Layer* geruzari bidaltzea beharrezkoa izango da *service-operative* interfazea erabiliz. Interfaze hau *API*tan antolatuta dago, zerbitzuetarako atzipena modu eraginkorrean kudeatzeko. Beraz, eskatzaileak dagokion *API*rako atzipena duen egiaztatzea beharrezkoa da. Horretarako, *API* horretara atzipena duten rolen zerrenda egiaztatzen da, eta eskatzailearen rolaekin alderatzen da. Rol kopurua eta mota aplikazio konketuaren araberakoa da, baina rol generiko batzuk identifikatu daitezke, hala nola aktiboen agintaria, produkzio-arduraduna edo aktiboen integratzailea (Braune et al., 2018, Figure 9). Horrela, 3-13 irudian adierazten den bezala, eskatzaileak atzipena badu, oinarrizko informazioa *Operative Layer* geruzara bidaltzen da; eskatzaileak atzipenik ez badu, eskatutako zerbitzua eskuragarri ez dagoela jakinarazten zaio.

AAS azpi-ereduak kudeatzeaz gain, *Service Layer* geruzak *proactive AAS*ei dagozkien erabakiak hartzeko gaitasunak ezartzen ditu. Horrek beste *I4.0 Component* osagai batzuei zerbitzu-eskaerak noiz egin zehazteko aukera ematen dio, bere aktiboari (AAS azpi-ereduen bidez) eta inguruneari (*Interoperability Layer* geruzatik jasotako datuen bidez) buruzko informazioan oinarrituta.

(API Id, Service Id, Input parameters)

Meta information

Core information

Irudia 3-13 Zerbitzu-eskaera *Operative Layer* geruzara bidaltzeko behar den informazioa.

3.3.1.3 *Operative Layer geruza*

Geruza hau *Service Layer* geruzak *Asset Related Services* zerbitzu-eskaerak jasotzen dituen beharrezkoa duen euskarria emateko diseinatuta dago. Horretarako, *Operative Layer* geruzak aktiboa automatizatzen du, aktiboaren egoera kontrolatuz eta, kasuan-kasuan, haren gainean jardunez edo egokiak diren gertaerak jakinaraziz. Gainera, *Asset Related Services* zerbitzu-eskaerak kudeatzeaz ere arduratzen da. Horretarako, zerbitzu jakin bati dagozkion aktibo-eragiketak identifikatu behar ditu (funtzionalitate-zati txikiagoak eta maneigarriagoak) eta modu antolatuan abiarazi. Eragiketa horiek *operative-functional* interfazearen bidez deitzen dira.

Maila honetan, aktibo-eragiketa bakoitza *Functional Layer* geruzaren unitate funtzional bati (adibidez, robot-programa bati) dagokio. Eragiketa bakoitzari dagozkion sarrera-parametroak *operative-functional* interfazearen bidez bidaltzen dira, eta erantzun gisa eragiketa bakoitzaren emaitzak jasotzen dira. Emaitza partzial horiek zuzenean edo eragiketa guztiak amaitu ondoren (hau da, zerbitzua amaitzean) bidal daitezke *Service Layer* geruzara, dagokionaren arabera.

3.3.1.4 *Functional Layer geruza*

Geruza hau *Asset Related Services* zerbitzuak burutzeko behar diren eragiketa guztiak kontrolatzeaz arduratzen da. Eragiketa horiek bi taldetan bana daitezke: aktiboari gauzak egiteko eskatzeko erabiltzen direnak (adibidez, sartutako parametroak kontuan hartuta, funtzionalitate jakin bat betetzen duten eragiketak) edo aktiboari informazioa eskatzeko erabiltzen direnak (adibidez, bere egoerari buruzko galderei erantzuteko). Azkenik, aipatzekoa da eragiketa horien exekuzioa *Functional Layer* geruzatik *Operative Layer* geruzara doan erantzun batekin amaitu behar dela beti, bai ondo burutu direnean (eragiketaren emaitzak bidaliz edo baieztapen-mezu bat bidaliz) bai burutu ezin izan direnean (errorearen berri emanez, eta, ahal bada, beraren zergatiaren berri emanez).

3.3.2 Integrazio-metodologia

Azpiatal honek lau urratseko integrazio-metodologia proposatzen du, aurreko atalean aurkeztutako geruza anitzeko ikuspegiaren aplikazioa arintzeko argibideak eta jarraibideak emateko. Jarraian, metodologia osatzen duten urratsak zehazten dira.

3.3.2.1 Antolamendua: aktibo erabilgarrien analisia

Lehenik eta behin, kontuan hartu behar da lan-espazioaren antolamendua eta integratu nahi dugun aktiboari (aurrerantzean aktibo nagusia deituko duguna) lotutako aktibo osagarriak daudela. Kasu gehienetan, Industria 4.0n aktibo bat integratzeak beste aktibo batzuk kontuan hartzera behartzen du. Horren adibide da aktibo nagusia (adibidez, robot bat) beste aktibo batek (adibidez, PLC batek) kudeatzen duenean. Gainera, aktibo nagusiak beste aktibo batzuk behar izaten ditu bere *Asset Related Services* zerbitzuak egiteko (adibidez, matxarda bat, material kontsumigarriak hornitzen dituen elikagailu bat, eta abar). Azkenik, aktibo nagusiaren komunikazio-gaitasunen arabera, aktibo gehigarriak beharrezko izango dira I4.0 Sistemara konektatzeko (adibidez, zirkuitu-plaka bakarreko ordenagailu bat, Raspberry Pi bezala). Aktibo osagarri guzti horiek identifikatzea garrantzitsua da, *Asset Related Services* zerbitzuak gauzatzeko edo geruzak hedatzeko balio baitezakete.

3.3.2.2 Zerbitzuen definizioa: geruzen arteko interfazeen zehaztapena

Hurrengo urratsa aktibo nagusiak eskaini beharreko *Application Relevant Services* zerbitzuak zehaztea da. Lehenago azaldu den bezala, zerbitzu horiek fabrika osorako adostutako formatu baten arabera definitu behar dira, hots, *meta information* informazioa zerbitzua nola prozesatu behar den zehazteko, eta *core information* informazio hura identifikatu eta parametrizatzeko. Zerbitzuak ezaugarritzeko formaturik definitu ez bada, fabrikaren aktibo guztiak kontuan hartuta definitzea gomendatzen da, etorkizunean aldaketak edo eguneratzeak egin behar ez izateko.

Definitutako zerbitzuak *Asset Related Services* direnean, geruza guztiek parte hartzen dutela nabarmendu behar da. Beraz, geruza batek goiko geruzari interfazeen bidez

eskaini beharreko metodoak definitu behar dira. Oro har, *Operative Layer* geruzak metodo bat eskaini behar dio *Service Layer* geruzari *Asset Related Service* zerbitzu bakoitzeko. *Functional Layer* geruzari dagokionez, *Operative Layer* geruzari metodo bat edo gehiago eskaini behar dizkio *Asset Related Service* zerbitzu bakoitzerako (zerbitzua egiteko behar den eragiketa edo kontsulta kopuruaren arabera).

3.3.2.3 Hedapena: gailua eta teknologia aukeratzea

Hirugarren urratsa geruza bakoitza non hedatu eta zein teknologia erabiliz zehaztea da, batetik, aktibo erabilgarrien gaitasun funtzionalak eta komunikatiboak (hau da, aktibo nagusia gehi aktibo osagarriak) eta, bestetik, geruza bakoitzaren eskakizunak kontuan hartuta. Adibidez, kasu gehienetan, *Operative Layer* eta *Functional Layer* geruzek denbora errealeko eskakizunak dituzte. Bestalde, *Service Layer* geruzak AAS azpi-ereduak biltegitratzeko gaitasuna izan behar du, edo, gutxienez, AAS azpi-ereduak atzipena izan behar du. Azkenik, *Interoperability Layer* geruzak komunikazio-gaitasun onak behar ditu.

Halaber, aktibo berean geruza bat baino gehiago zabal daitekeela azpimarratzea garrantzitsua da, hala badagokio (adibidez, PLC batek berak kontrolatutako eta kudeatutako eragingailu baten *Operative Layer* eta *Functional Layer* garuzak ezar ditzake). Hala ere, erabaki hori hartzen bada, oso komenigarria da geruzak bereiztea, soluzioaren berregingarritasuna eta moldagarritasuna zaintzeko.

3.3.2.4 Garapena: behetik gorako ezarpena

Azkenik, azken urratsa geruzak eta horiek konektatzen dituzten interfazeak garatzea da. Behetik gorako ezarpena egitea proposatzen da, geruza bakoitza azpian dagoen geruzaren gainean eraikitzen baita. Horrela, lehenik eta behin, *Functional Layer* geruza ezarri behar da *Asset Related Services* zerbitzuak eskaintzeko oinarri diren eragiketak emanaz. Jarraian, *Operative Layer* geruza ezartzen da. Puntu honetan, garrantzitsua da geruza honen eta *Functional Layer* geruzaren arteko konektibitatea bermatzea. Gainera, *Operative Layer* geruza hartzen duen aktiboak direktorio bat kudeatu behar du zerbitzu bakoitza egiteko behar diren eragiketen zerrendarekin. Azkenik, *Service*

Layer eta *Interoperability Layer* geruzak ezartzen dira, aktibo nagusiaren AASa osatzen dutenak. Urrats honen barruan AAS azpi-ereduak garatzea sartzen da, *I4.0 Component* osagaien *Application Relevant Services* zerbitzu guztiak adierazteko, bai eta horien egoera ere.

3.4 AAS garapena

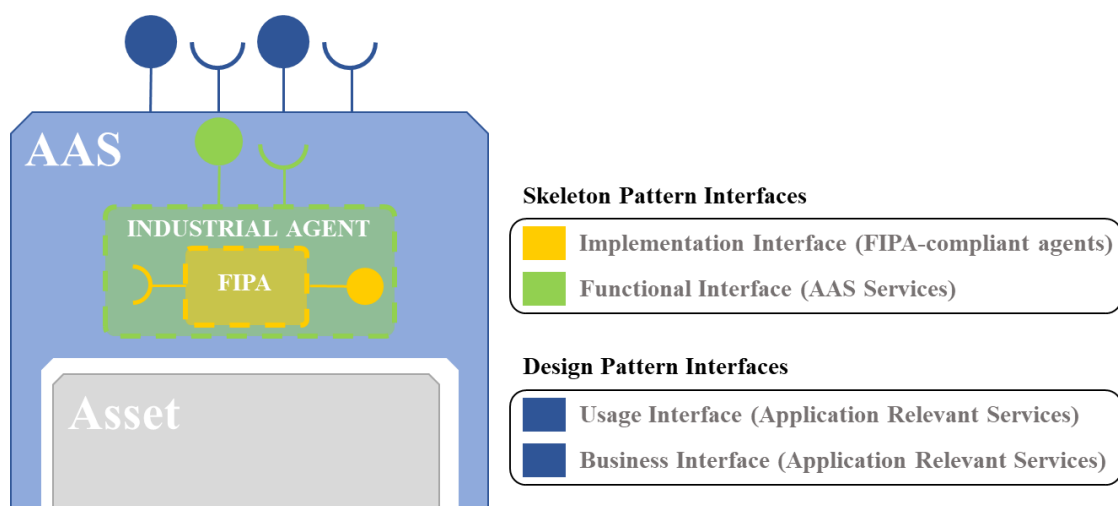
Atal honetan, AASen garapenerako hurbilketa bat aurkeztu nahi da, *RAMI 4.0* erreferentzia-arkitektura jarraituz, *IIRA* erreferentzia-arkitekturatik datozen kontzeptuak ere barne hartzen dituenak. Ikuspegi hori *Plattform Industrie 4.0*k jasotzen dituen zerbitzu-kategorietan oinarritzen da. 2.2.1 azpiatalean aipatu den bezala, AASEk *Application Relevant Services* zerbitzuak elkarri eskaintzen dizkiote, *Infrastructure Services* izeneko beheagoko mailako zerbitzuek lagunduta. Era berean, *Infrastructure Services* zerbitzuak honela banatzen dira: *AAS Services* eta *AAS Infrastructure Services*. Lehenengoak AASEk berek egiten dituzte beren barne-kudeaketarako, eta bigarrenak I4.0 Plataformak eskaintzen ditu AASak sortzen, erregistratzen eta kudeatzen laguntzeko. Zerbitzu-kategoria horien dependentziak aztertuta, argi dago *Application Relevant Services* zerbitzuak aplikazioaren menpe daudela, aktiboaren gaitasunen arabera *I4.0 Component* osagai bakoitzerako desberdinak baitira. Aldiz, *Infrastructure Services* zerbitzuek *I4.0 Component* osagaien arteko elkarreragingarritasuna bermatzen dute, osagaiak integratzen duten aktiboa edo eskaintzen dituen *Application Relevant Services* zerbitzuak direnak direla. Horregatik, bideragarria ez ezik, oso gomendagarria ere ematen du *Infrastructure Services* zerbitzuak modu generikoan ezartzea, erabiltzen diren aplikazio zehatza edozein dela ere.

Analisi horren emaitza bezala, I4.0 Plataforma fabrikazio-domeinurako modu generikoan diseinatu daitekeela ondoriozta daiteke. Aldiz, *I4.0 Component* osagaiak aplikazioarekiko zati independente bat dute (*Application Relevant Services* zerbitzuetarako atzipenez eta AASaren interakzioak kudeatzeaz arduratzen dena), baita aplikazioaren menpeko zati bat ere (AASaren eta aktiboaren ezaugarriak zehazten dituenak, *I4.0 Component* osagaiak eskainitako *Application Relevant Services* zerbitzuak barne). Horregatik, AASen garapena bi fasetan egitea proposatzen da:

- Lehenik eta behin, *skeleton pattern* txantiloia, AASEk behar dituzten *Infrastructure Services* zerbitzuen ezarpen generikoa (hau da, *AAS Services*) eskaintzen duena.
- Ondoren, *skeleton pattern* txantiloa *design pattern* txantiloia desberdinak sortzeko zabal daiteke, AASak fabrikako aktiboetara egokitzeko.

Proposamen horri koherentzia emateko, txantilo horiek interfazeetan antolatu dira IIRAn proposatutako *Viewpoints* ikuspuntuen arabera. (Lin et al., 2019). Horregatik, *skeleton pattern* txantiloak *implementation interface* interfaze bat du, azpiko teknologia interfaze estandarizatu baten bidez abstraitzen duena (adibidez, *REST API*). Horren gainean *functional interface* interfaze bat eraikitzen da, *AAS Services* zerbitzuak eskaintzen dituena.

Design pattern txantiloietan, *usage interface* eta *business interface* interfazeak daude. Interfaze horiek *functional interface* interfazeak ematen dituen *AAS Service* zerbitzuetan oinarritzen dira, *Application Relevant Services* zerbitzuak fabrika edo negozio mailako beste *I4.0 Component* osagai batzuei eskaintzeko. 3-14 irudiak AASak garatzeko proposamena erakusten du, *skeleton pattern* txantiloia batean oinarrituta, *design pattern* txantiloia baten bidez zabaldu daitekeena. Hurrengo azpiataletan txantilo horiei buruzko xehetasun gehiago ematen dira.



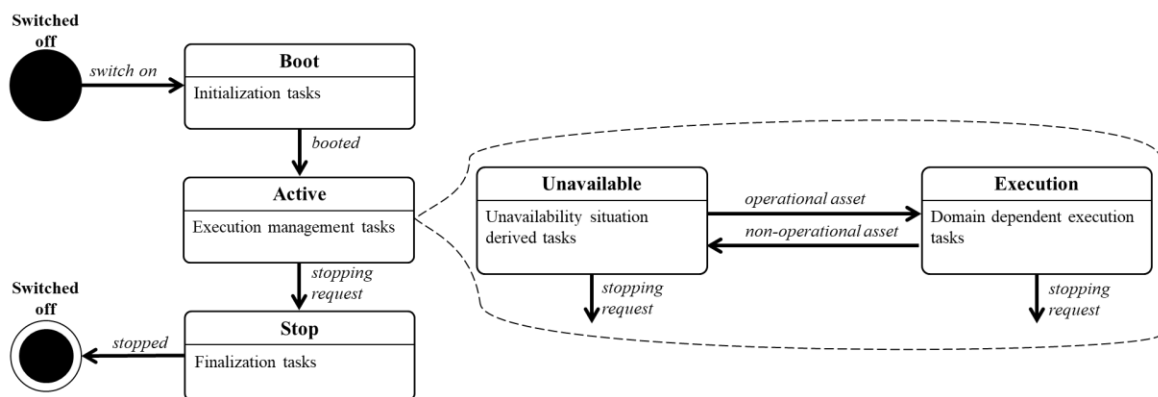
Irudia 3-14 AASak bi etapatan garatzeko proposamena: *skeleton pattern* eta *design pattern* txantiloak.

3.4.1 *Skeleton pattern* txantiloia

Skeleton pattern txantiloiak AASa kudeatzeko beharrezkoa den azpiegitura eskaintzen du *AAS Service* zerbitzuen bidez. Hau da, *skeleton pattern* txantiloiak AASeko *Component Manager* ezartzen du. Horretarako, *implementation interface* interfazean oinarritzen da, maila baxuko zerbitzu generikoak eskaintzen dituen, erabilitako oinarritzko teknologia abstraituz. Interfaze hau *functional interface* interfazearen oinarria da, informazioa erregistratzeko zerbitzuak, *Application Relevant Service* zerbitzuetarako atzipena edo erabakiak hartzeko gaitasunak eskaintzen dituen, besteak beste.

Component Manager-ak *I4.0 Component* osagaiaren bizi-zikloan *Application Relevant Services* zerbitzuak noiz eta zergatik gauzatu behar diren ere kudeatzen du. Zentzu honetan, *skeleton pattern* txantiloiak egoera finituen makina baten eredua jarraitu dezala proposatzen da, 3-15 irudian agertzen dena:

- *Boot*: AAS sortu osteko hasierako egoera da. Egoera horretan, abiarazte-lanak egiten dira (adibidez, aldagaiak abiaraztea, AASaren erregistroa eskatzea, etab.). Lan horiek ondo burutzen badira, *active* egoerara pasatzen da.
- *Active*: egoera honek, batetik, fabrikazio-prozesuan zehar *Application Relevant Services* zerbitzuen exekuzioa, eta, bestetik, aktiboa aldi baterako erabilgarri ez dagoen egoerak kudeatu behar ditu. Gertaera horiek elkarrekiko eksklusiboak dira, eta, beraz, *active* egoera bi egoera dituen azpiegoera-makina gisa definitu da (ikus 3-15 irudia):
 - *Unavailable*: AASak egoera honetan bere aktiboak funtzionatzen ez duen bitartean eta arazoa konpontzen den arte, edo *stop* egoerara pasatzera behartuta agertzen den arte jarraitzen du.
 - *Execution*: Azpiegoera honen edukia *design pattern* txantiloian zehazten da, bere edukia aplikazioaren menpe baitago. Egoera honetatik *unavailable* edo *stop* egoeraetara pasa daiteke.
- *Stop*: egoera honetan, AASen bukaera lanak egiten dira (adibidez, sistemako erregistrotik bere informazioa ezabatuz).

Irudia 3-15 *Skeleton pattern* txantiloia-aren egoera finituen makina.

3.4.2 *Design pattern* txantiloia

Design pattern txantiloiek *skeleton pattern* txantiloien *application interface* eta *functional interface* interfazeak erabiltzen dituzte beren aktiboei dagozkien *Application Relevant Services* zerbitzuak ezartzeko, bai eta zerbitzu horiek ezaugarritzeko beharrezko diren azpi-ereduak ere. Zerbitzu horiek bi interfazetan banatzen dira: *usage interface* eta *business interface*. Bi interfazeak hierarkikoki maila berean daude (hau da, sistemako beste *I4.0 Component* osagai batzuei zerbitzuak eskaintzen eta eskatzen dizkiete). Hala ere, bi interfaze desberdin proposatzen dira, eskaintzen/eskatzen dituzten zerbitzuen izaera desberdina dela eta. *Usage interface* interfazeak fabrikazio-prozesuaren garapenean inplikaturako zerbitzuak ematen edo eskatzen ditu. Aldiz, *business interface* interfazeak planifikazioarekin eta negozio-estrategiarekin erlazionaturako erabakiak hartzeko goi-mailako prozesuetara bideraturako zerbitzuak ematen edo eskatzen ditu. *I4.0 Component* osagai bakoitzak bi interfaze horien zerbitzuak eska ditzake eta horietako baten bidez, gutxienez, zerbitzuak eman.

Skeleton pattern txantiloiatik abiatuta *design pattern* txantiloia-ak garatzeak aplikazio jakin bati egokitzeko pertsonalizazioa eskatzen du. Batetik, aktiboa integratzea beharrezkoa da *Application Relevant Services* zerbitzuak oinarri dituzten funtzionalitateak eta datuak eskuratu ahal izateko. Bestetik, *I4.0 Component* osagaiek eta *I4.0* Plataformak uler ditzaketen azpi-ereduetan funtzionalitate eta datu horiek ezaugarritzeko beharrezkoa da. Funtsezkoa da pertsonalizazio hau modu generikoan

egitea sistemako *I4.0 Component* osagai guztientzat, elkarreragingarritasuna eta soluzioaren eskalagarritasuna indartzeko. Horregatik, pertsonalizazio hau 3.2 atalean ikusitako aktiboen karakterizazioan eta 3.3 atalean ikusitako aktiboen integratzeko ikuspegiari oinarritzea proposatzen da.

3.5 Laburpena eta ondorioak

Kapitulu honek hiru proposamen metodologiko aurkezten ditu *RAMI 4.0* erreferentzia-arkitekturan oinarrituta, Industria 4.0 testuinguruan enpresei eta erabiltzaileei aktiboen karakterizazioari, aktiboen integrazioari eta *AASen* garapenari aurre egiten laguntzeko.

3.2 atalak Industria 4.0rako fabrikazio-sistemarako eredu-egite ikuspegia aurkezten du, *RAMI 4.0ren hierarchy levels* ardatza arautzen duten estandarren arabera. Eredu-ikuspegi honekin, fabrikazioaren trazabilitatea ahalbidetu nahi da, bai *resources* (prozesuaren ikuspegitik trazabilitatera bideratuak), bai *applications* elementuekin (produktuaren ikuspegitik trazabilitatera bideratuak). Hala ere, ez da *personnel* baliabidea sartu, eta horregatik etorkizuneko lanerako kontuan hartu beharko da.

3.3 atalak aktibo fisikoen integrazioa jorratzen du Industria 4.0n, *proactive AASen* bidez, eta geruza anitzeko ikuspegia proposatzen du, zeregin zehatz eta kudeagarrietan integrazioarekin zerikusia duten alderdiak uztartzeko. Ikuspegi hau integrazio-metodologia batekin osatzen da. Metodologia honek zerbitzuak definitzeko euskarria ematen du, besteak beste, ezarpen-teknologiak aukeratzeko edo geruzak hedatzeko. Ikuspegi honek ikuspuntu berri bat eskaintzen du artearen egoeran eztabaidatutako antzeko lanei dagokienez (ikus 2.3 atala): kasu honetan, aktiboen integrazioa ez da *AASen* garapenaren beste alderdi bat, alderdi nagusia da baizik. Proposamen honen beste alderdi aipagarri bat generikoa dela da, beraz aktibo mota eta aktiboak komunikatzeko erabiltzen duen teknologia kontuan hartu gabe aplikatu daiteke.

3.4 atalean, *Plattform Industrie 4.0* k aintzat hartutako AASen zerbitzu-kategoriak aztertzen dira, AASen garapena nola bideratzen den zehazteko. Azterketa horretatik ondorioztatzen da *Application Relevant Services* zerbitzuen aldean *Infrastructure Services* zerbitzuak oro har fabrikazio-domeinurako defini daitezkeela. Horregatik, bi urratseko soluzio bat proposatu da: *skeleton pattern* eta *design pattern* txantiloiak. Ikuspegi honek aukera ematen du *Infrastructure Services* zerbitzu guztiak aldi berean ezartzeko, *I4.0 Component* osagaien eta I4.0 Plataformaren arteko elkarreragingarritasuna bermatuz. Gainera, AAS pertsonalizatuak garatzeko oinarri bat ematen du, ezarpen-teknologiari buruzko ezagutza sakonik izan gabe, 3.2 eta 3.3 ataletan aurkeztutako ikuspegi metodologikoetan oinarrituta.

Kapitulu honetan aurkeztutako ikuspegi metodologikoak Industria 4.0rako soluzioak garatzeko oinarri sendoak azaltzen dituzte, eta, beraz, doktorego-tesi honen emaitza nagusi bezala aurkezten den I4.0 Plataforma diseinatzeko eta garatzeko oinarri gisa erabili izan dira (ikus 4. kapitulua).

4 I4.0 PLATAFORMAREN DISEINUA ETA GARAPENA

4.1 Sarrera

Kapitulu honek doktorego tesi honen emaitza nagusia den I4.0 Plataformaren diseinua eta garapena deskribatzen ditu. Plataforma honen helburua I4.0 Sistemak *RAMI 4.0* erreferentzia-arkitekturaren arabera kudeatzea da, honako alderdi hauetan ardaztuz:

- Sistematan dauden *AAS*en (eta, beraz, *I4.0 Component* osagaien) arteko elkarreragingarritasuna euskarritzea, baita I4.0 Plataformarekin ere.
- Fabrikazio-prozesuaren operazio-egoeraren kudeaketa eta funtzionamendu zuzena bermatzea.
- Plataformaren pertsonalizazioa errazteko oinarri teknologiko eta metodologiko bat eskaintzea, oinarrizko teknologia menperatzea eskatzen ez duena.

Helburu horiek betetzeko, I4.0 Plataforma honek 3 kapituluan proposatutako ikuspegi metodologikoak erabiltzen ditu. Beraz, *skeleton pattern* txantilo multzo bat eskaintzen du *AAS*ak garatzeko oinarri gisa, 3.4 atalean aurkeztutako proposamena jarraituz. *Skeleton pattern* txantilo hauek *AAS Services* zerbitzuak ezartzen dituzte, *AAS*en arteko elkarreragingarritasuna bultzatuz eta *Application Relevant Services* zerbitzuen atzipena kudeatuz. I4.0 Plataforma honetan jasotako *skeleton pattern* txantiloak 3.2 atalean aurkeztutako eredu-egite ikuspegiaren arabera ezaugarritu dira. *Skeleton pattern* txantiloietan oinarritutako *AAS*en garapena duten *customization mechanisms* mekanismoetan oinarritzen da, 3.3 atalean aurkeztutako geruza anitzeko ikuspegiari jarraituz. Mekanismo horiek aktiboen integrazioa modu generikoan errazten dute.

Skeleton patterns txantiloiez gain, I4.0 Plataforma honek *I4.0 Platform core* nukleo bat eskaintzen du, *AAS Infrastructure Services* zerbitzuak ezartzen dituena. Zerbitzu hauek aplikazio mailako kudeaketan laguntzen dute, *AAS* berrien sorrera eta erregistroa berehalakoan gaituz. *I4.0 Platform core* nukleoa fabrikazio-aplikazioetan gertatzen diren gorabeheraz ere arduratzen da (adibidez, mezu-galerak, denbora-mugan gainditzea, etab.), eta neurrira egindako erantzunak integrazteko *customization mechanisms* mekanismoak eskaintzen ditu.

I4.0 Plataformaren diseinua *FIP*Arekin bat datozen industria-agenteei zuzenduta dago. *ACL* komunikazio-protokoloak, *FIP*Ak sustatua, hainbat mekanismo biltzen ditu, hala nola ontologiak eta performatiboak erabiltzea komunikazioen testuingurua errazteko. Honek bat egiten du industria-agenteeek berez negoziatzeko eta erabakiak modu autonomoan hartzeko dituzten ahalmenekin. Ezaugarrien konbinazioa aproposa da *I4.0 Platform core* nukleoak eta *skeleton pattern* txantiloiek eman beharreko *Infrastructure Services* zerbitzuak garatzeko.

Kapitulu honen gainerakoak ondorengo egitura jarraitzen du. 4.2 atalak I4.0 Plataformaren proposamena gainbegiratzen du, bere barne egitura eta ezaugarri nagusiak deskribatuz. 4.3 atalak xehetasunez erakusten du *AAS*en garapenera nola burutu daitekeen *skeleton pattern* txantiloietatik abiatuta, eta *AAS*en funtzionamendua baldintza normaletan deskribatzen du. 4.4 atala I4.0 Plataformak fabrikazio-prozesuen jarraipena egiteko eta horiek behar bezala betetzen direla bermatzeko eskaintzen dituen baliabideetan ardatzen da. Azkenik, 4.5 atalak kapitulua ondorioekin ixten du.

4.2 IARMS I4.0 Plataformaren ikuspegi orokorra

Atal honetan *Industrial Agents for Resilient Manufacturing Systems (IARMS)* I4.0 Plataforma gainbegiratzen da. Plataforma hau MAS-RECON arkitekturan oinarrituta eraikita dago (Gangoiti et al., 2022, Fig. 1). MAS-RECON UPV/EHUko Sistemen Kontrol eta Integrazio Taldeko (GCIS⁴) beste ikertzaile baten doktoretza-tesiaren emaitza da. Arkitektura hau ez dago fabrikazio domeinurako karakterizatua, ezta *RAMI 4.0*rekin lerrokatua ere. Hala ere, domeinuaren pertsonalizazioa onartzen du eredu-egiteak bultzatutako ingeniarietza (*Model-Driven Engineering* edo *MDE*) teknikekin eta domeinuaren menpeko agenteak garatzeko bi txantilo, *Resource Agents (RAs)* edo *Application Agents (AppAs)* izan daitezkeenak. Horrez gain, *System Supervisory Agents (SSAs)* agenteak eskaintzen ditu, domeinuaren menpeko agenteak kudeatzeko ezaugarri erabilgarriak dituztenak (adibidez, egoera finituen makinan oinarritutako

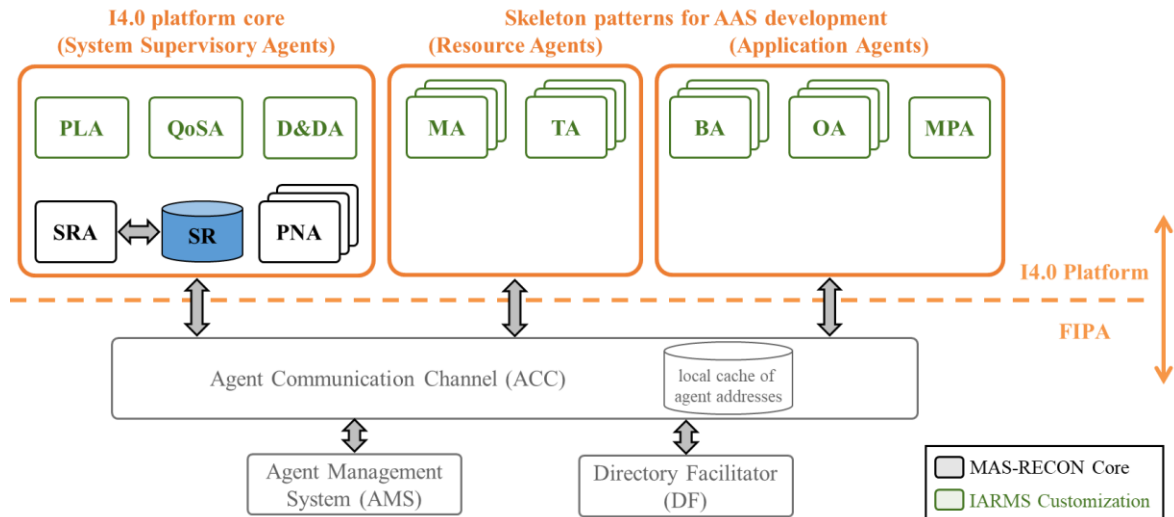
⁴ <https://ehu-gcis.org/>

bizi-zikloaren kudeaketa edo zerbitzuetara bideratutako elkarrekintzak). SSA agenteak nahi beste aldatu eta/edo zabaldu daitezke.

IARMS I4.0 Plataformak bereak egiten ditu MAS-RECONek emandako baliabideak eta tresnak, eta aldaketa garrantzitsuak aplikatzen dizkio I4.0 Sistemen beharrak betetzeko, *RAMI 4.0* erreferentzia-arkitektura betez. Beraz, 4-1 irudian adierazten den bezala, *IARMS* I4.0 Plataforma bi ezarpen-mailatan egituratuta dago: alde batetik, MAS-RECON nukleoak, beltzez nabarmendua, baldintza normaletan I4.0 Sistemaren kudeaketari laguntzeko funtzionalitateak ematen ditu; bestetik, *IARMS* pertsonalizazioak, berdez nabarmenduta, funtzionalitate gehigarriak ematen ditu (adibidez, I4.0 Sistemaren erabiltzaile-interfaze bat edo fabrikazio-prozesuan gertaerak kudeatzeko gaitasuna), eta AASak garatzeko *skeleton pattern* txantiloit multzo bat.

IARMS I4.0 Plataformako agenteak bi taldetan bana daitezke, funtzionalitateen arabera: *skeleton pattern* txantiloiak eta *I4.0 Platform core* nukleoa.

Lehenengo taldean I4.0 Sistemako AASak garatzea helburu duten *RA* eta *AppA* agenteak daude. *RA* eta *AppA* agenteak bi alderditan bereizten dira: integratu nahi diren aktibo mota eta haien bizi-zikloa. *RA* agenteek aktibo fisikoak osatzen dituzte, *AppA* agenteek, ordea, aktibo logikoak osatzen dituzte. Beraz, aldea dago aktiboak eskaintzen dituen datuetara eta funtzionalitateetara atzitzeko moduan: *RA* agenteen kasuan, integrazioa aktiboen komunikazio-gaitasunek mugatzen dute; *AppA* agenteen kasuan, berriz, agenteen integrazioa zuzenean egiten da. Bizi-zikloari dagokionez, *RA* agenteek ordezkatzeko dituzten aktiboek hainbat fabrikazio-aplikaziotan parte har dezakete, eskaeraren arabera eta eskuragarri dauden bitartean, aplikazioek beren helburuak betetzeko behar dituzten zerbitzuak emanaz. Horregatik, *RA* agenteak iraunkorrak dira: behin sortuta, denbora mugarik gabe sisteman abiarazita geratzen dira arazoren bat azaldu edo erabiltzaileak nahita ezabatu ezean. *AppA* agentei lotutako aktiboen bizi-zikloa, aldiz, fabrikazio-aplikazio baten barruko zeregin jakin bati lotuta dago. Horregatik, *AppA* agenteak iragankorrak dira: lana amaitutakoan, ez dute beste funtziorik sisteman.



Irudia 4-1 IARMS I4.0 Plataformaren ikuspegi orokorra. Egokitzapena (Gangoiti et al., 2022, Fig. 1). SRA = System Repository Agent, SR = System Repository, PNA = Processing Node Agent, PLA = Planner Agent, QoS SA = Quality of Service Agent, D&DA = Diagnosis and Decision Agent, MA = Machine Agent, TA = Transport Agent, BA = Batch Agent, OA = Order Agent, MPA = Manufacturing Plan Agent.

RA skeleton pattern txantiloiei dagokienez, soluziorik errazena mota bakarra definitzea da, edozein fabrikazio-aktiboarentzat (Bennulf et al., 2020; Munkelt and Krockert, 2018). Hala ere, granularitate handiagoa ere kontuan hartu daiteke (adibidez, Robot Agents (Kovalenko et al., 2022), Conveyor Agents (Kovalenko et al., 2022; H. Tang et al., 2018) edo AGV Agents (D. Tang et al., 2018) agenteak, besteak beste). Lan horiek kontuan hartuta, soluzio generikoegi baten (hau da, fabrikazio-baliabide mota bakarra kontuan hartzea) eta espezifikoegi baten (hau da, fabrikazio-baliabide mota gehiegi definitzea) arteko oreka bilatu da. Ondorioz, eta 3.2 atalean proposatutako eredu-egite ikuspegi batzuz, IARMS I4.0 Plataformak fabrikazio-baliabideak oinarritzko bi motatan sailkatzen ditu: makinak (hau da, produktuak eraldatzen dituzten baliabideak) eta garraioak (hau da, produktuak mugitzen dituzten baliabideak). Ondorioz, bi RA skeleton pattern txantiloiei definitu dira: Machine Agent (MA) eta Transport Agent (TA).

Bestetik, AppA skeleton pattern txantiloiei dagokienez, fabrikazio-aplikazioak produktu-eskaeretara mugatzen dira, Product Agent (PA) agentea horrelako agente bakartzat hartuz (Bennulf et al., 2020; Kovalenko et al., 2022; H. Tang et al., 2018). Ikuspegi horiek trazabilitate unitarioa ahalbidetzen dute, eta hori funtsezkoa da Industria 4.0n, baina trazabilitatea beste granularitate-maila batzuetan baztertzen dute. Hori dela eta, IARMS I4.0 Plataformak hiru aplikazio-entitate zehaztu ditu (batch,

order eta *manufacturing plan*) fabrikazio-prozesua maila desberdinetan monitorizatzeko, 3.2 atalean aurkeztutako eredu-egite ikuspegiaren arabera. Entitate horietako bakoitzak bere *skeleton pattern* txantiloia du: *Batch Agent (BA)* agenteak enpresak produktu-sortaren (bat ego gehiago) trazabilitatea kudeatzen du; *Order Agent (OA)* agenteak bezero mailako trazabilitatea eskaintzen du (hau da, bezero-eskaera beraren parte diren sorta guztiak monitorizatzen ditu); azkenik, *Manufacturing Plan Agent (MPA)* agenteak fabrikazio plan osoaren garapenaren jarraipena egiten du. Aplikazio-entitate horiek (eta, hedaduraz, horien *skeleton pattern* txantiloiek) harreman hierarkikoa dute, eta, beraz, *BA* agente bat edo gehiago *OA* agente bakar baten menpe daude, eta *OA* agenteek *MPA* agente bakarraren menpe daude. *RA* eta *AppA* agenteen diseinua eta ezaugarriak 4.3 atalean zehatz-mehatz deskribatzen dira.

Bigarren taldean *System Supervisory Agents (SSA)* agenteak daude, *I4.0 Platform core* nukleoa osatzen dutenak. Prozesua ikuspegi global batetik kudeatu eta gainbegiratzeaz arduratzen dira, euskarri funtzionala zein konputazionala emanez. Talde honen barruan, agenterik nabarmenena *System Repository Agent (SRA)* agentea da. *SRA* agentea *System Repository (SR)* biltegia kudeatzeaz arduratzen da, sisteman dauden *I4.0 Component* osagai guztiei buruzko funtsezko informazioa baitu. *Plattform Industrie 4.0k* definitzen dituen *Infrastructure Services* zerbitzuak kontuan hartuta (Miny et al., 2021, Fig. 18), *SRA* agentea *AAS Infrastructure Services* zerbitzuak emateaz arduratzen da: *I4.0 Components* osagaiak sortu eta erregistratzean, identifikadore bakarra esleitu eta informazioa *SR* biltegian erregistratzeaz arduratzen da (horrela, *AAS Create Service* eta *AAS Registry Services* zerbitzuak ematen ditu). *SRA* agenteak *I4.0 Component* osagaiak *SR* biltegitik ere kontsulta, edita eta ezaba ditzake. Gainera, informazio honetarako atzipena kudea dezake: sistemako parte-hartzaile orok beste *I4.0 Component* osagai bateko informazioa eskura dezake bere identifikadorea eskatuta edo eskatutako ezaugarri jakin bat betetzen badu (atributuak, eskaintako zerbitzuak eta abar). Funtzionalitate hori bat dator *AAS Exposure and Discovery Services* zerbitzuekin.

SRA agentea osatzeko, *Planner Agent (PLA)* agentea garatu da. *PLA* agenteak erabiltzaile-interfaze gisa funtzionatzen du, eta *SRA* agenteak eskaintzen dituen *AAS Infrastructure Services* zerbitzuetara sartzeko aukera ematen dio erabiltzaileari. Beraz, *IARMS I4.0* Plataformako erabiltzaile batek *I4.0 Component* osagaiak sortu eta erregistratu ditzake, eta *PLA* agentea erabiliz horien egoera banaka edo taldean kontsultatu.

AAS Infrastructure Services (I4.0 Platform core nukleoan ezarria) eta *AAS Services (skeleton pattern* txantiloetan ezarria) zerbitzuen batuketak *Plattform Industrie 4.0* delakoak aintzat hartutako *Infrastructure Services* zerbitzuen erabateko ezarpena ematen du. Hala ere, zerbitzu hauek ez dute fabrikazio-aplikazioen erresilientzia bermatzen, hau da, "*sistema batek aldaketara egokitzeko eta desiratu gabeko egoeratik desiratutako egoerara itzultzeko duen gaitasuna*" (Alexopoulos et al., 2022). Horregatik, *IARMS I4.0* Plataformak bere *I4.0 Platform core* nukleoan *Quality of Service Agent (QoSA)* eta *Diagnosis and Decision Agent (D&DA)* agenteak barne hartzen ditu. Agente hauek harreman estua dute beraien artean, fabrikazio-prozesuan gertaerak kudeatzeaz arduratzen baitira. *QoSA* agentea sistemako agente guztiak zaintzeaz arduratzen da, sistemaren funtzionamenduan arazoak detektatzeko, iragazteko eta baieztatzeko helburuarekin. Arazoren bat egiaztatuz gero, *QoSA* agenteak *D&DA* agenteari jakinaraziko dio, eta hark sistemaren egoera berrezartzeko beharrezko ekintzak egingo ditu. Agente horrek interfaze pertsonalizatu bat du, erabiltzaileei fabrikazio-aplikazioei ondoen egokitzen zaizkien ekintzak definitzeko aukera ematen diena. 4.4 atalean *QoSA* eta *D&DA* agenteak xehetasun handiagoz aztertzen dira.

Azkenik, *Process Node Agent (PNA)* agenteak sisteman dauden prozesatze-nodoak adierazten ditu, sistemaren konputazio gaitasuna behar bezala aitortzen eta eskuratzen dela bermatuz. *PNA* agenteek haien artean hainbat irizpideren arabera negoziatzea dezakete (esaterako, erabilgarri dagoen memoria, prozesadorearen karga, etab.), sistemako agenteei konputazio eta komunikazio-gaitasunak emanez.

4.3 AAS garatzeko *skeleton pattern* txantiloiak

Atal honetan, *RA* eta *AppA skeleton pattern* txantiloien diseinua eta garapena deskribatzen dira, bi ikuspuntutatik:

- Egiturari dagokionez, *skeleton pattern* txantiloiak nola garatu diren azaltzen da, *MAS-RECON*ek horretarako emandako txantiloietan oinarrituta. Azken horiek, *SRA* agentearekin elkarreragiteko oinarritzko metodoak eta egoera finituen makinak (ikus Gangoiti et al., 2022, Fig. 4) ematen dituzte.
- Funtzionaltasunari dagokionez, *skeleton pattern* txantiloiek *AAS*en funtzionamendua baldintza normaletan nola bermatzen duten erakusten da, *AAS Services* zerbitzuak gauzatuz.

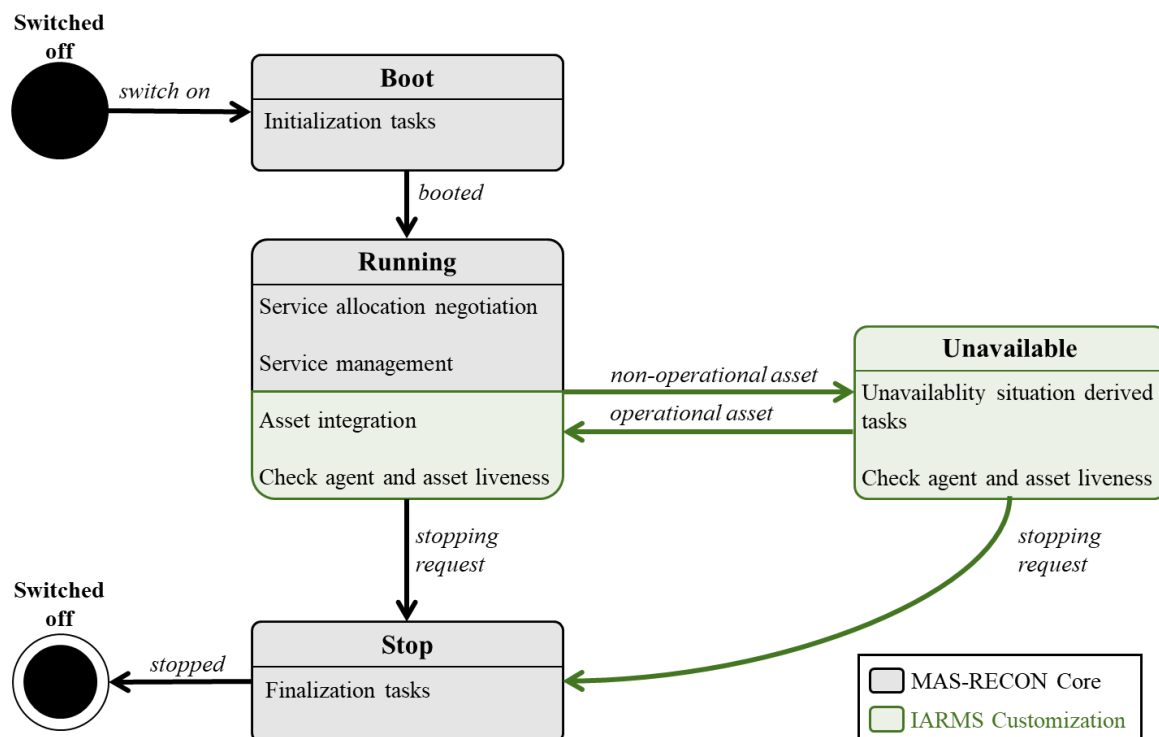
Hurrengo azpiataletan bi puntu hauetan sakontzen da *RA* eta *AppA skeleton pattern* txantiloietarako, hurrenez hurren.

4.3.1 *RA skeleton pattern* txantiloiak

Azpiatal honetan, *RA skeleton pattern* txantiloien diseinua eta garapena zehazten da, egiturazko ikuspegia eta ikuspegi funtzionala kontuan hartuz. Horrez gain, aktiboen integrazioan (eta, horrekin batera, *AAS*en garapenean) laguntzeko *customization mechanisms* mekanismoak azaltzen dira.

4.3.1.1 *Egiturazko garapena*

RA agenteen kasuan, *MAS-RECON*ek emandako txantiloia-egoera finituen makinak hiru egoera ditu: abiarazte-lanak egiten dituen *boot* egoera; zerbitzuak esleitzeko negoziazioak eta esleitutako zerbitzu-eskaerak kudeatzen dituen *running* egoera; eta bukaerako lanak egiten dituen *stop* egoera. Egoera finituen makina hori *RA skeleton patterns* txantiloietan hedatu da, ondorengo aldaketak aplikatuz (ikus 4-2 irudian):



Irudia 4-2 IARMS I4.0 Plataformako RA skeleton pattern txantiloietarako egoera finituen makina.

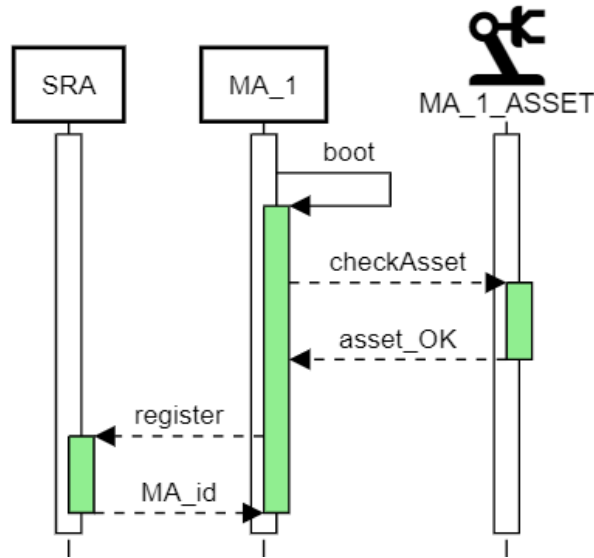
- *Running* egoeraren gaitasunak zabaldu dira fabrikako aktibo fisikoekiko elkarreragiteak modu generikoan kudeatzeko. Gaitasun berri horiek zerbitzu-eskaeren kudeaketari dagozkionekin konbinatuz, 3.3.1 azpiatalean aurkeztutako geruza anitzeko ikuspegiaren *Service Layer* geruzaren ezarpena lortzen da. *Running* egoerak dituen gaitasun berriek agentearen eta aktiboaren bizitasuna egiaztatze aukera ematen dute, baita mezuen jasotzea ere.
- *Unavailable* egoera berria sartu da, 3-15 irudian *skeleton pattern* txantiloietarako proposatutako estatu finituen makinaren *unavailable* egoerarekin bat datorrena. Egoera honek *I4.0 Component* osagaia bizirik egon arren aktiboa erabilgarri ez dagoen kasuak adierazten ditu (adibidez, matxura mekaniko bat edo mantentze-gelditze bat gertatzen bada). Erabiltzaileak behar dituen aplikazioen eragiketak ezartzeaz gain, egoera honek agentearen eta aktiboaren bizitasuna eta mezuen jasotzea egiaztatu ahal izango ditu, *running* egoerak bezalaxe. *Unavailable* egoerara *running* egoeratik irits daiteke, eta *unavailable* egoeratik *running* edo *stop* egoerara joan daiteke.

4.3.1.2 Garapen funtzionala

RA skeleton pattern txantiloien deskribapen funtzionalak *AAS Service* zerbitzuen ezarpena erakusten du, *AASek* baldintza normaletan funtziona dezaten lagunduz.

Boot egoeran, *RA skeleton pattern* txantiloiek konfigurazio-fitxategi batetik (hau da, *AAS* azpi-ereduetatik) jasotzen dute beren aktiboari buruzko informazio guztia, eta, beraz, *Submodel Registry Services* zerbitzuak betetzen dituzte. Ondoren, sisteman erregistro bat dago, 4-3 irudian *MA_1*en kasurako agertzen den bezala. Aldez aurreko baldintza gisa, aktibo fisikoaren erabilgarritasuna egiaztatzen da mezu bat bidaliz (ikus *checkAsset* 4-3 irudian). Erantzunik ez badago, edo aktiboak lanerako prest ez dagoela adierazten badu, erregistro prozesua bertan behera geratuko da. Bestela, *SRA* agenteari eskatzen zaio *AAS* berriari identifikadore bakarra esleitzeko eta bere informazioa *SR* biltegian erregistratzeko (ikus *register* 4-3 irudian). Erregistroaz gain, erabiltzaileak nahi beste hasierako eragiketa gehitu ditzake.

Abiaraztea amaitutakoan, hots, *Boot* egoera amaitutakoan, *RA skeleton pattern* txantiloiak *running* egoeran sartzen dira. Hemendik aurrera, *Application Relevant Services* zerbitzu-eskaerak kudeatzeko prest egon beharko lukete. Eskaera horiek *AAS* jakin bati zuzendu dakizkioke (adibidez, bere aktiboaren egoerari buruz galdetzeko, hala nola uneko tenperaturari edo abiadurari buruz), edo negoziazio baten ondorio izan daitezke (esaterako, eskatzaileak *Application Relevant Service* zerbitzu bat behar duenean eta hori egiteko hautagairik egokiena zein den ez dakienean). Azken kasu horretarako, *running* egoerak negoziazio-mekanismo bat ezartzen du, modu deszentralizatuan eskatutako *Application Relevant Services* zerbitzuak nork bete behar dituen erabakitzeko aukera ematen duena. Mekanismo hori *AAS* bakoitzaren negoziazio-balioa kalkulatzeko moduarekiko independentea da, erabiltzaileak kasu bakoitzean dituen irizpideen arabera delarik. Era berean, *AAS* irabazleak egin beharreko ekintzak negoziazioaren helburuaren arabera dira. Bi kasuetan, erabiltzaileak alderdi horiek pertsonalizatu ahal izango ditu bi metodori esker: *calculateNegotiationValue* eta *checkNegotiation*.



Irudia 4-3 MA skeleton pattern txantiloiaeren interakzioak boot egoeran adierazten dituen sekuentzia-diagrama.

AASek zuzenean edo negoziazio baten ondorioz jasotzen dituen *Application Relevant Services* zerbitzu-eskaerak *running* egoeran kudeatzen dira. Zehazki, *Submodel Services* zerbitzu-eskaerak zuzenean kudeatzen dira. *Asset Related Services* zerbitzu-eskaerei dagokienez, azpi-eredu batean erregistratzen dira, ordenan kudeatu ahal izateko. Horrela, *Submodel Exposure and Discovery Services* zerbitzuak eskaintzen dira.

4.3.1.3 Agenteetan oinarritutako aktiboak integratzen laguntzeko customization mechanisms mekanismoak

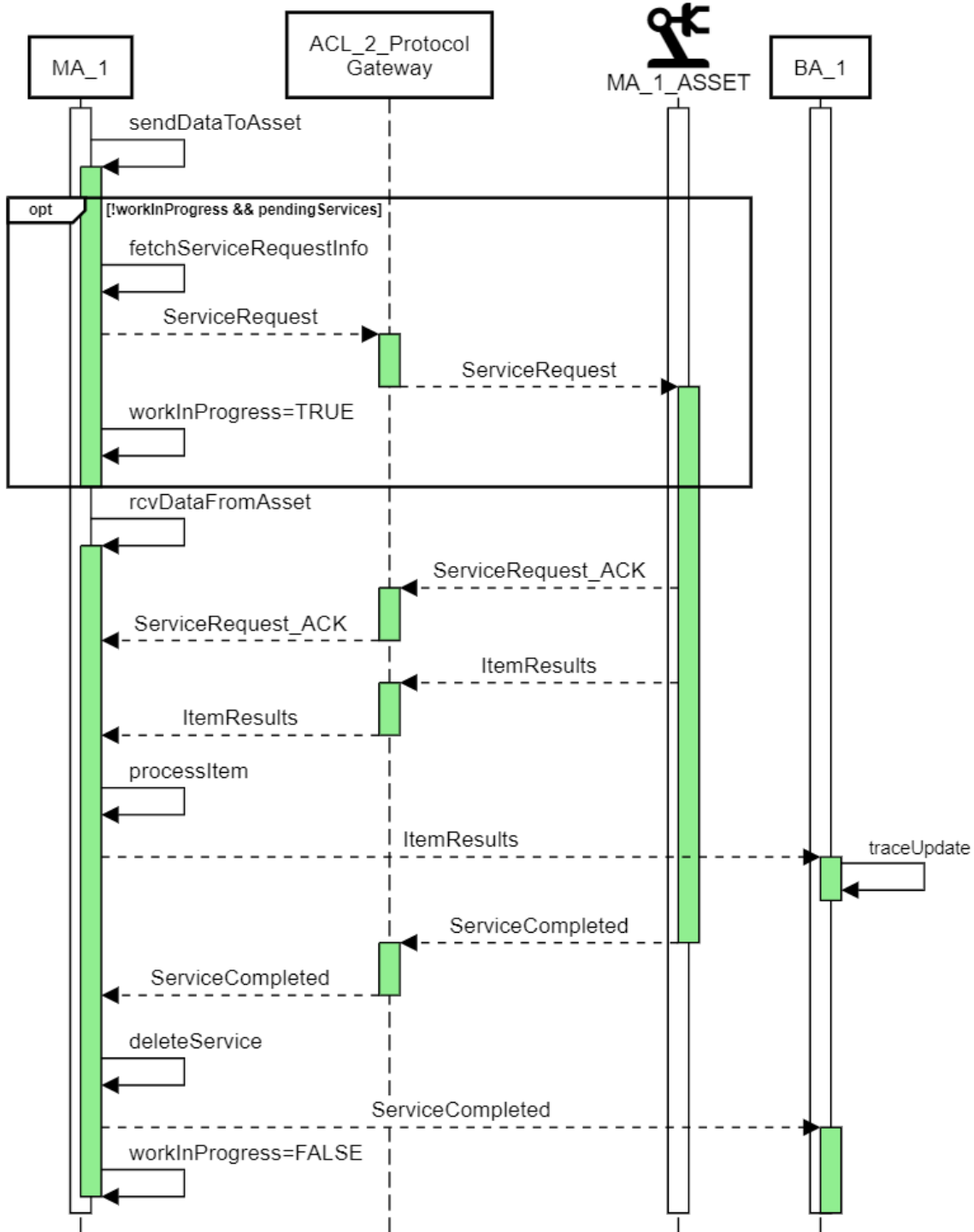
RA skeleton pattern txantiloien analisia aktiboaren integrazioa errazteko aurreikusitako *customization mechanisms* mekanismoen deskribapenarekin amaitzen da. Mekanismo hauek 3.3 atalean aurkeztutako geruza anitzeko ikuspegia aplikatzen dute. Gainera, soluzio horrek industria-agenteentzat gomendatutako praktikei buruzko 2660.1-2020 IEEE estandarrean jasotako integrazio-mota guztiak betetzen dituela egiaztatu da (“IEEE Recommended Practice for Industrial Agents,” 2021).

Fabrika bateko aktiboak integratzean direnean, komunikatzeko gaitasun desberdinak dituzten aktibo desberdinek *Application Relevant Services* zerbitzu berberak eskaintzea gerta daiteke. Ondorioz, mota bereko bi aktibo integratzeko (hau da, *Application Relevant Services* zerbitzu berak eskaintzeko), bi *design pattern* txantilo

desberdin ezarri beharko lirateke. Eragozpen hau saihesteko, aktiboarekiko komunikazioak agente laguntzaile batean abstraitzea proposatzen da, AASen funtzionaltasunetik bereizteko (hau da, batetik, *Application Relevant Services* zerbitzuen kudeaketa eta, bestetik, AAS batzuekiko interakzioa bereizteko). Agente laguntzaile honek, *Gateway Agent (GA)* izenekoa, zubi lana egiten du AASaren (*FIPA*ko agenteei datxekien *ACL* protokoloa erabiltzen duena) eta aktiboaren (beste komunikazio protokolo bat erabiltzen duena) artean.

RA skeleton pattern txantiloiak interfaze bat eskaintzen du AASen eta *GA* agenteen arteko interakzioak estandarizatzeko erabiltzen diren bi metodoekin: *sendDataToAsset* eta *rcvDataFromAsset*. Metodo hauek 4-4 irudiak adierazten duen bezala funtzionatzen dute:

- *sendDataToAsset*: hasieran, aktiboa aske dagoen ala ez erabakitzeko *workInProgress* seinale bat ebaluatzen da (ikus *workInProgress* 4-4 irudian). Jarraian, *Asset Related Services* zerbitzu-eskaerarik egiteke dagoen egiaztatzen da (ikus *pendingServices* 4-4 irudian). Baldintza hauek betetzen badira, hurrengo *Asset Related Service* zerbitzuari buruzko informazioa dagokion azpieren biltzen da (ikus *fetchServiceRequestInfo* 4-4 irudian) eta *GA* agenteari bidaltzen zaio. Informazio hau bidali ondoren, *workInProgress* seinalea aktibatzen da zerbitzua amaitu arte, informazio berriaren bidalketa blokeatzeko (*workInProgress=TRUE* 4-4 irudian).
- *rcvDataFromAsset*: metodo honek *GA* agentetik jasotako mezuak bakarrik prozesatzen ditu. Mezu horiek aurreko mezu bat jaso dela baieztatu dezakete edo *Asset Related Service* zerbitzu baten exekuzioaren emaitza partzialak edo osoak izan ditzakete. Azken kasu honetan, jasotako emaitzak inplikaturako produktuaren edo sortaren jarraipena egiten ari den *BA* agenteari bidaltzen zaizkio. Horrez gain, zerbitzua amaitzen denean, informazioa *Asset Related Services* zerbitzuaren zerrendatik kentzen da, eta *workInProgress* seinalea desaktibatzen da (ikus *workInProgress=FALSE* 4-4 irudian).



Irudia 4-4 AASen eta GA agenteen arteko interakzioak estandarizatzeko erabiltzen diren *sendDataToAsset* eta *rcvDataFromAsset* metodoen funtzionamendua adierazten duen sekuentzia-diagrama.

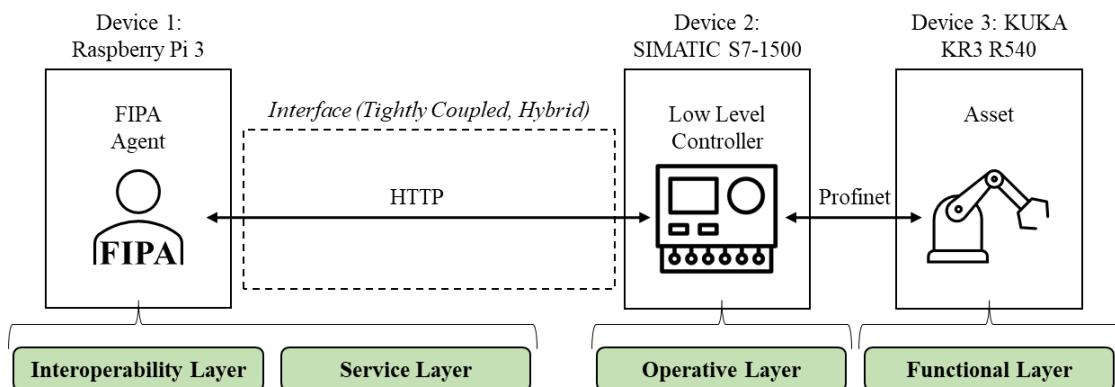
Azkenik, *customization mechanisms* mekanismo hauek IEEE 266.1-2020 estandarrarekin bat datozen aztertu da, industria-agenteak erabiltzen dituzten aktiboen integrazioan oinarritzen dena. Estandarrak lau integrazio-praktika aurkezten

eta alderatzen ditu industria-agentearen eta aktiboaren arteko elkarrengaitze moduen eta agentearen hedapen-kokapenaren arabera. Horrela, agentearen eta aktiboaren arteko elkarrekintza *tightly coupled* (zerbitzuetara bideratutako komunikazio sinkrono bat ezartzen da, lotura iraunkor eta zuzen baten bidez) edo *loosely coupled* (datuetara bideratutako komunikazio asinkrono bat ezartzen da, bitartekaritza-kanal baten bidez) izan daiteke. Hedapenari dagokionez, *hybrid* (agentea aktiboa kontrolatzeko erabiltzen ez den beste gailu batean hedatzen da) edo *on device* (agentea aktiboa kontrolatzeko erabiltzen den gailu berean hedatzen da) izan daiteke.

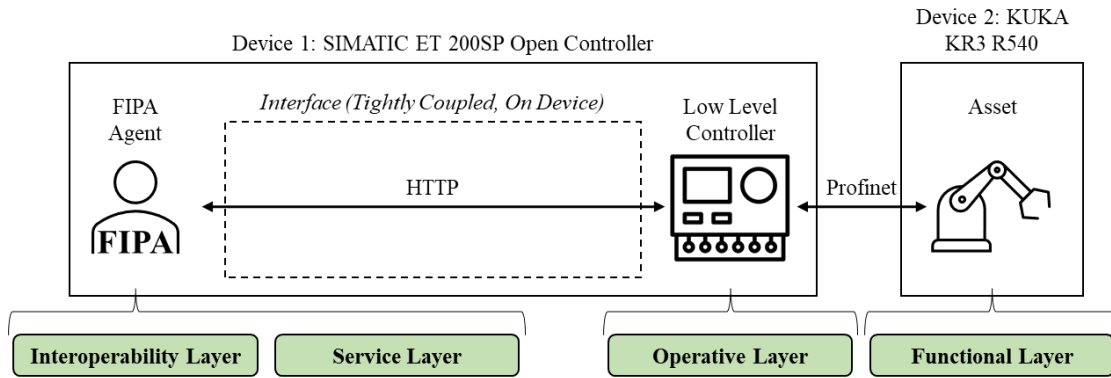
Testuinguru honetan, lehen proposatutako soluzioa IEEE 2660.1-2020 estandarrean identifikatutako lau integrazio-praktiketako edozeinetan aplikatu daiteke:

- Bi elkarrekintza moduentzako euskarria eskaintzen du, aktiboarekin elkarrengaitze erabiltzen den komunikazio-protokoloa GA agentean abstraituta baitago.
- Era berean, bi hedapen moduentzako euskarria eskaintzen du, aktiboarekiko komunikazioak GA agentean abstraitzeak AASaren hedapen gailua aukeratzeko malgutasuna ematen baitu.

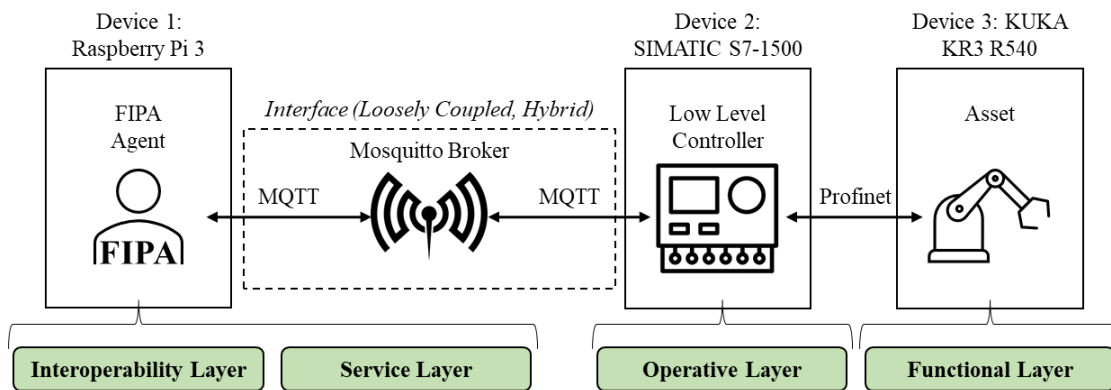
4-5 eta 4-8 bitarteko irudiek adibide sinpleen bidez soluzio hau IEEE 2660.1-2020 estandarrean aintzat hartutako lau praktiketan nola aplikatu daitezkeen erakusten dute.



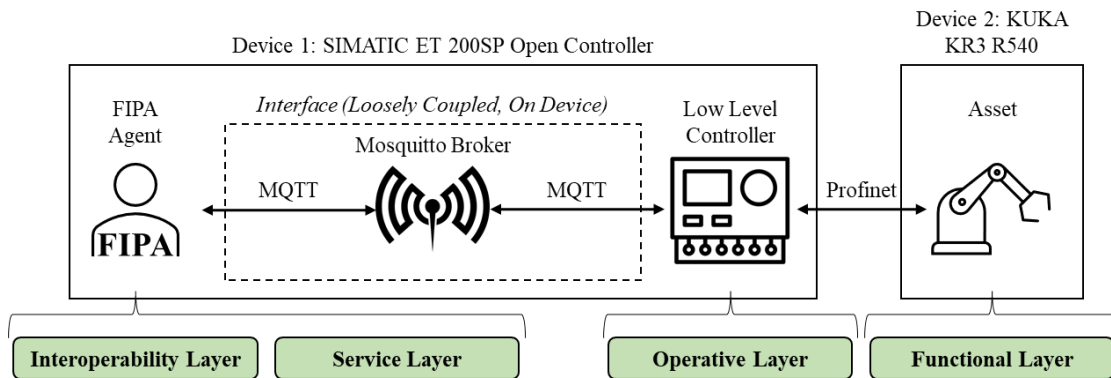
Irudia 4-5 Geruza anitzeko ikuspegiaren aplikazioa *tightly coupled, hybrid* praktikatara. Egokitzapena (Leitão et al., 2018, Figure 5).



Irudia 4-6 Geruza anitzeko ikuspegiaren aplikazioa *tightly coupled, on device* praktikarako. Egokitzapena (Leitão et al., 2018, Figure 6).



Irudia 4-7 Geruza anitzeko ikuspegiaren aplikazioa *loosely coupled, hybrid* praktikarako. Egokitzapena (Leitão et al., 2018, Figure 8).



Irudia 4-8 Geruza anitzeko ikuspegiaren aplikazioa *loosely coupled, on device* praktikarako. Egokitzapena (Leitão et al., 2018, Figure 9).

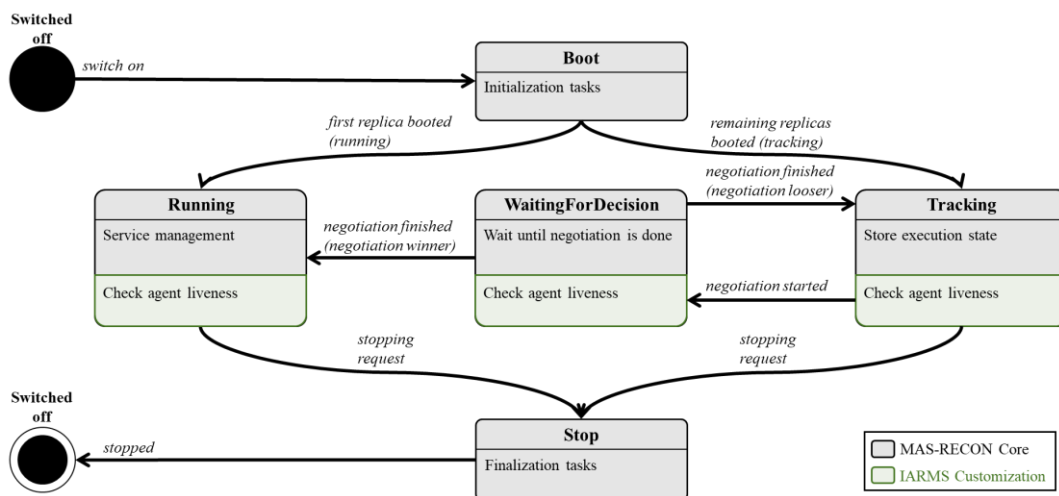
4.3.2 AppA skeleton pattern txantiloak

Azpiatal honetan *AppA skeleton pattern* txantiloien diseinua eta garapena zehazten da, egiturazko ikuspuntu batetik zein ikuspuntu funtzional batetik aztertzen da.

4.3.2.1 Egiturazko garapena

MAS-RECONek *AppA* agenteentzat emandako txantiloiairen egoera finituko makinak *RA skeleton pattern* txantiloian ikusitako hiru egoerak aurkezten ditu (*hots*, *boot*, *running* eta *stop*, ikus 4.3.1 azpiatala). Ezberdintasun bakarra *running* egoerak negoziazio-gaitasunak barne hartzen ez dituela da (*AppA* agenteek ez dute negoziatu beharrik, parte hartzen duten aplikazioen barruan lan zehatzak egiten dituztelako; *RA* agenteek, berriz, hainbat aplikaziotarako zerbitzuak eskaini ditzakete). Hain zuzen ere, *AppA* agenteek aplikazio jakin baten barruan eragiketa zehatzak egiten dituztenez, huts eginez gero ezin dira beste *AppA* agente batzuekin ordezkatu (*RA* agenteekin gerta daitekeen bezala). Horregatik, MAS-RECONek (eta hedaduraz *IARMS* I4.0 Plataformak) *AppA* agenteen egoera mantentzen duten erreplikak kudeatzeko euskarria eskaintzen du, sistemaren erabilgarritasuna bermatuz. Horretarako, *AppA* agenteen egoera finituen makina honek beste bi egoera ditu: *tracking* eta *waiting for decision*.

AppA skeleton pattern txantiloiairen kasuan, ez zen beharrezkoa izan MAS-RECONen txantiloia eskaintzen duen egoera finituen makinarekiko egoera gehigarririk gehitzea. Hala ere, *AppA skeleton pattern* txantiloia aktibo dagoen hiru egoerak (*running*, *tracking* eta *waiting for decision*) agentearen bizitasuna eta mezuen jasotzea egiaztatzeko gaitasunarekin zabaldu dira. Ondoriozko egoera finituen makina 4-9 irudian agertzen da.



Irudia 4-9 *IARMS* I4.0 Plataformaren *AppA skeleton pattern* txantiloietarako egoera finituen makina.

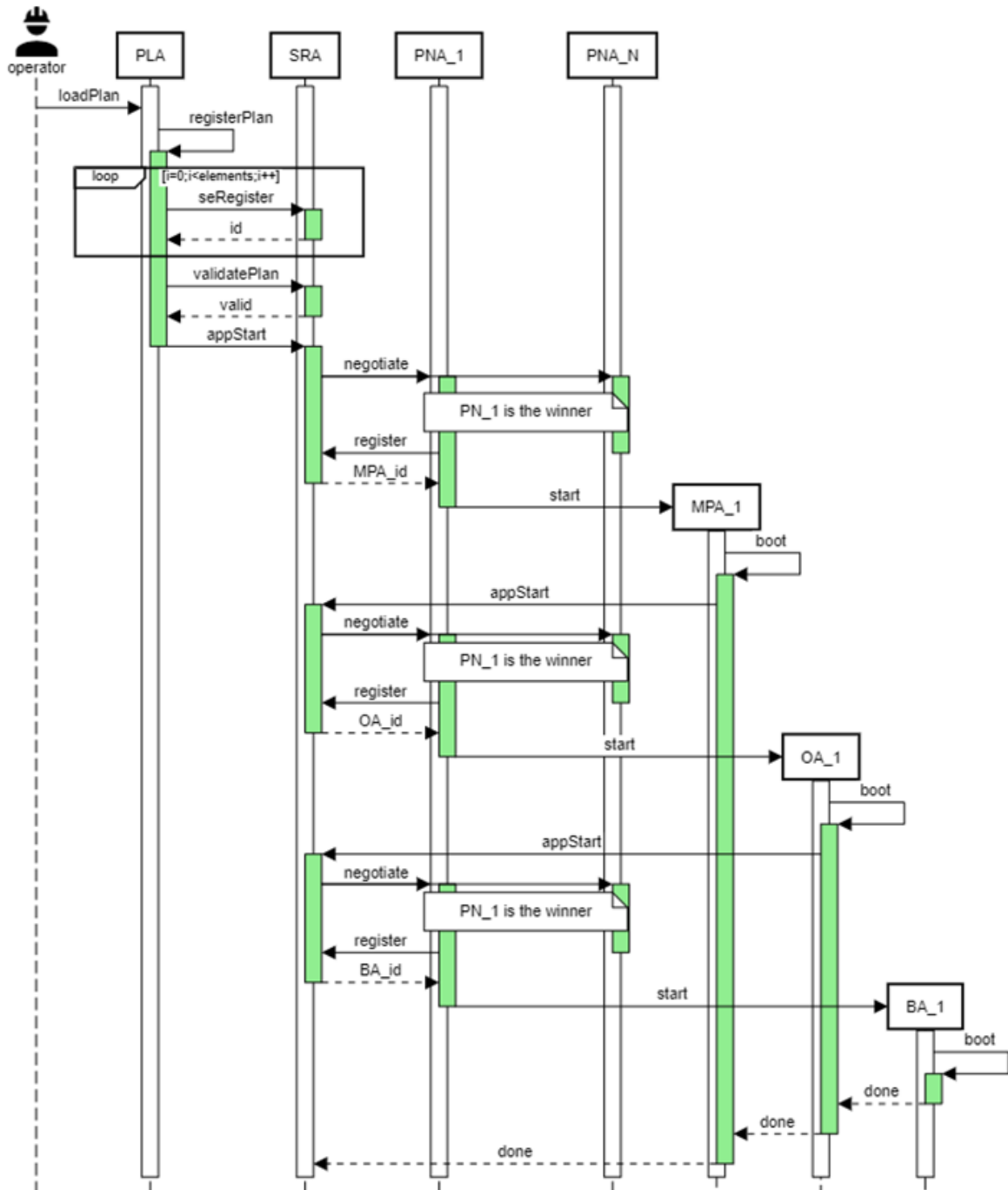
4.3.2.2 Garapen funtzionala

AppA skeleton pattern txantiloien funtzionamendua *RA skeleton pattern* txantiloien *AAS Services* zerbitzu beretan oinarritzen da. Hala ere, *boot* eta *running* egoeretan burutzen diren eragiketetan desberdintasunak daude.

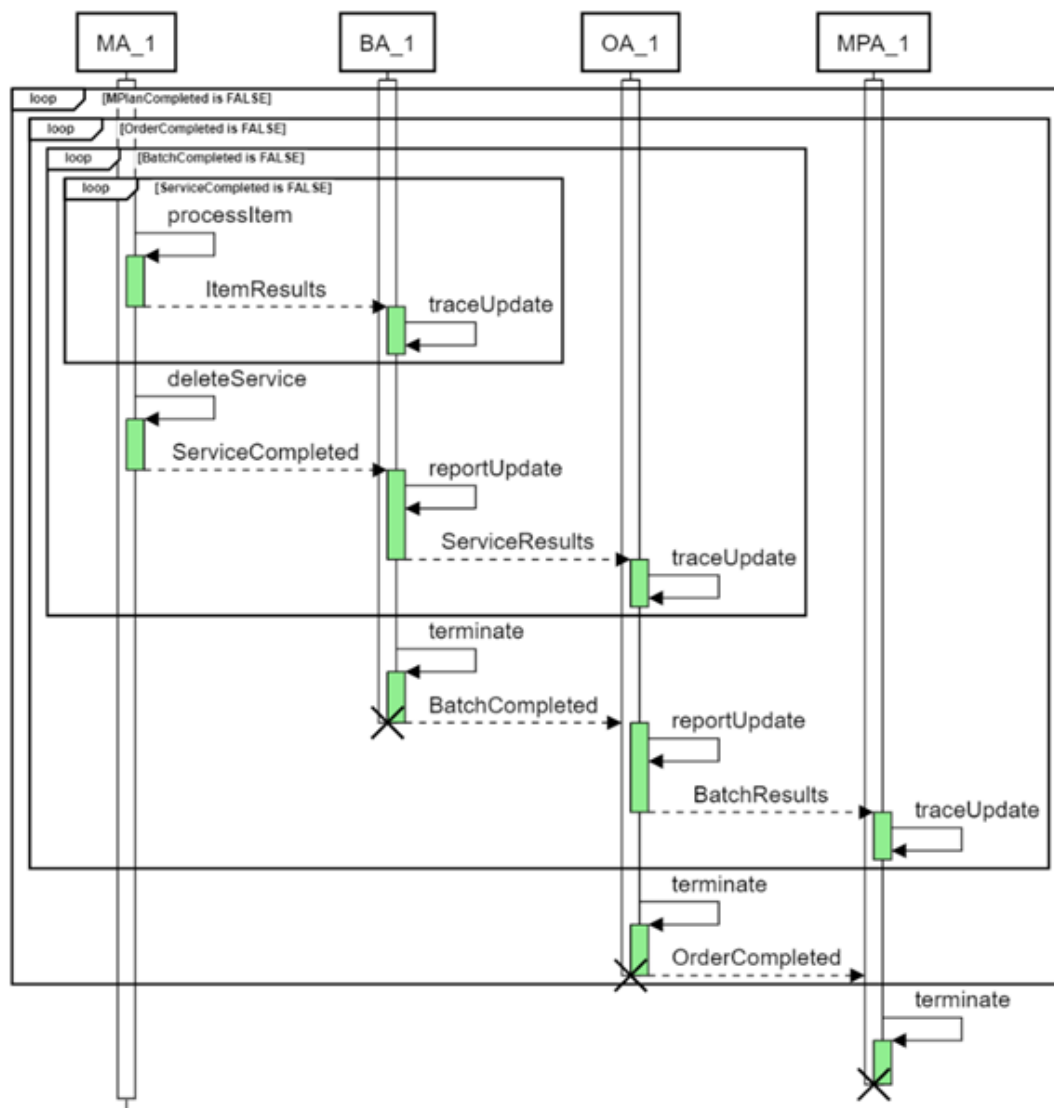
AppA skeleton pattern txantiloiek hasierako egoeraren informazioa *PLA agentea* erabiltzen duen operadore batek sisteman kargatutako fabrikazio-plan batetik jasotzen dute (ikus *loadPlan* 4-10 irudian). *AASa* sortu eta erregistratu aurretik, fabrikazio-plana dagokion domeinuaren meta-ereduarekin bat datorrela egiaztatzen da (ikus 3.2 atala). Egiaztapen hau bi faseetan egiten da. Lehen fasean, planeko entitate bakoitza banaka aztertzen da (ikus *seRegister* 4-10 irudian). Bigarren faseak bere hierarkia zuzena dela egiaztatzen du (ikus *iValidate* 4-10 irudian). Prozesu hau xehetasun handiagoz azaltzen da hemen (Casquero et al., 2020). Egiaztapena zuzena bada, lehenengo elementu hierarkikoa sortuko da, 3.2 atalean proposatutako eredu-egite ikuspegiaren arabera *MPA* dena (ikus *appStart* 4-10 irudian). *MPA* agentea prozesatze-nodo egokienean hedatzen da, sistema funtzionamenduari dagokionez (adibidez, erabilgarri dagoen memoria, prozesadorearen karga, etab.). Hori *PNA* agenteen arteko negoziazio batek zehazten du (ikus *negotiate* 4-10 irudian). Fabrikazio plan osoa martxan jartzeko, *MPA* agenteak bere menpe dituen *OA* agenteak abiarazteaz arduratzen da. Era berean, *OA* agenteak beraien menpe dauden *BA* agenteak abiarazteaz arduratzen dira. Horrez gain, *MPA* eta *OA* agenteek beren seme guztien abiaraztea amaitu dela baieztatu arte itxarongo dute *running* egoerara igaro aurretik. 4-10 irudiaren behealdean ikus daitekeenez, *OA* agente batek bere seme diren *BA* agenteak *running* egoerara igaro direla baieztatu arte *boot* egoeran jarraitzen du. Gauza bera gertatzen zaio *MPA* agenteari bere seme diren *OA* agenteen baieztapena jasotzen ez duen bitartean.

Behin *running* egoeran daudenean, *AppA skeleton pattern* txantiloiek modu berean funtzionatzen dute: jasotzen duten informazioaren arabera, beren aplikazio-entitatearen trazabilitatea erregistratzeko azpi-eredua eguneratzen dute (ikus *traceUpdate* 4-11 irudian). Gainera, informazioa bildu ahala, hurrengo maila

hierarkikoari jakinarazten diote (ikus *reportUpdate* 4-11 irudian). Azkenik, monitorizatzen duten entitateak fabrikazio-prozesua amaitzen duenean, hurrengo mailara bidaltzen dute beren trazabilitatea erregistratzeko azpi-eredu osoa, eta *stop* egoerara igarotzen dira, sistematik ezabatzen direlarik (ikus *terminate* 4-11 irudian).



Irudia 4-10 MPA, OA eta BA skeleton pattern txantiloien interakzioak boot egoeran adierazten dituen sekuentzia-diagrama.



Irudia 4-11 MPA, OA eta BA skeleton pattern txantiloien interakzioak *running* egoeran adierazten dituen sekuentzia-diagrama.

4.4 I4.0 Components osagaien kudeaketa erresilientea

Atal honetan *QoSA* eta *D&DA* agenteek *IARMS I4.0* Plataformari fabrikazio-prozesuan gertaerak detektatzeko eta konpontzeko mekanismoak nola ematen dizkioten zehazten da. Proposamen honek bi agente independenteetan (*QoSA* eta *D&DA*, hurrenez hurren) banatzen ditu fabrikazio-prozesuan gertaeren detektatzea eta ebaztea. Honen arrazoia, bi alderdi hauek (hots, gertaeren detekzioa eta ebazpena) aplikazioekiko menpekotasun maila desberdina azaltzen dutelako gertatzen da:

- Lan honen ikuspuntutik, fabrikazio-prozesuko gertaeren detekzioa aplikazioarekiko independentea da, kontuan hartzen diren hiru gertaera motak (denbora-muga gainditzeak, mezu-galerak edo agente-akatsak) sistemako agenteen arteko interakzioetatik abiatuta detektatu eta egiaztatu baitaitezke. Horrela, *QoSA* agentea modu generikoan, hots, aplikazioarekiko modu independentean, ezarri da.
- Fabrikazio-prozesuaren gertaerak ebaztea, aldiz, aplikazioaren menpe dago, gertaera mota batentzako egin beharreko ekintzak aldatu egin baitaitezke aplikazioaren ezaugarrien eta erabiltzailearen eskakizunen arabera. Horregatik, soluzio zehatz bat inposatu beharrean, *D&DA* agenteak gertaera mota bakoitzarekin erlazionatutako metodoak dituen interfaze pertsonalizagarri bat eskaintzen du.

Hurrengo azpiataletan, *QoSA* eta *D&DA* agenteen egitura eta ezaugarriak sakonago lantzen dira.

4.4.1 *QoSA* agentea: gertaeren detekzioa

Azpiatal honetan, *QoSA* agenteak egiaztatu ditzakeen hiru gertaera motak aztertzen dira (denbora-muga gainditzeak, mezu-galerak edo agente-akatsak). Jarraian, *QoSA* agenteak fabrikazio-prozesuan gertaera horietako bakoitza nola detektatzen duen eta hauek baieztatzeko edo baztertzeke zein urrats jarraitzen dituen azaltzen da.

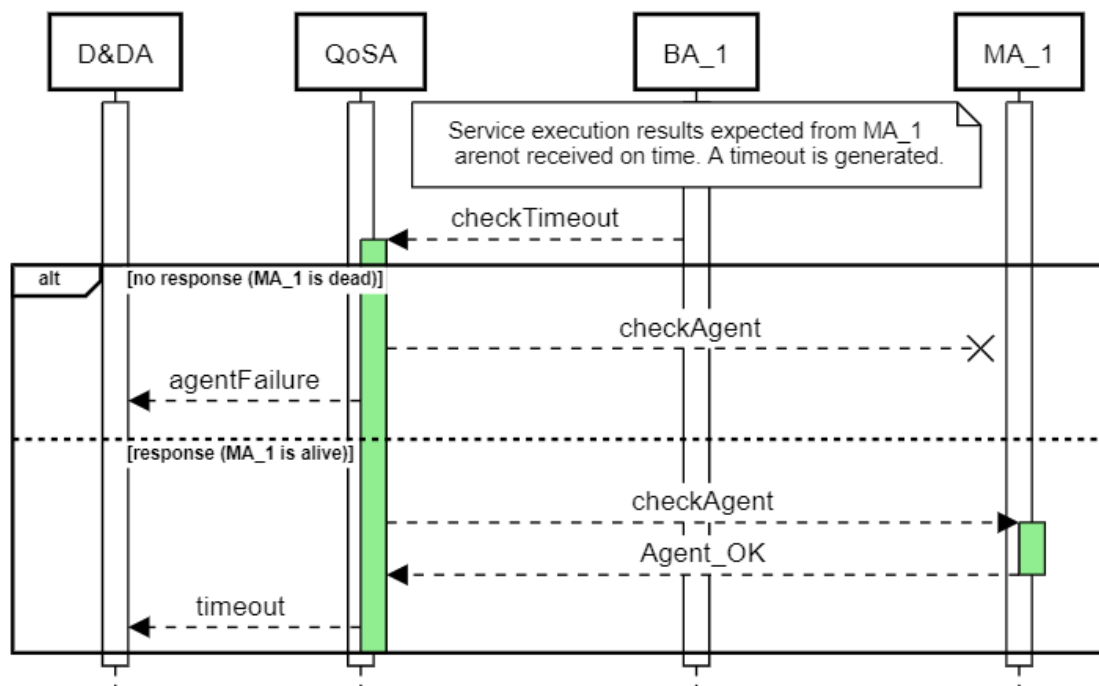
4.4.1.1 Denbora-muga gainditzeak

Fabrikazio-prozesuari lotutako denborak fabrikaren maila guztietan ebaluatu daitezke. *RA* eta *AppA* agenteek denbora-mugak gainditu diren egiaztatzen dute, prozesuaren bilakaera azpi-ereduetan aurreikusitako denborekin alderatuz. Horrela, adibidez, *BA* agente batek denbora-muga gainditze bat iragarriko du dagokion azpi-ereduan erregistratuta dagoen unean fabrikazio-zerbitzu bati lotutako emaitzak jasotzen ez baditu. Era berean, *MA* agente batek denbora-muga gainditze bat iragarriko

du kontsumigarriak berritzeko garraio baten zain badago eta aurreikusitako denboran dagokion *TA* agentearengandik informaziorik jaso ez badu.

4-12 irudiak erakusten du *QoSA* agenteak nola kudeatzen duen denbora-muga gainditze bat *BA_1* en kasuan. Zerbitzu bati buruzko informazioa jasotzeko denbora-muga gainditzen denean, *BA_1* agenteak mezu bat bidaltzen dio *QoSA* agenteari (ikus *checkTimeout* 4-12 irudian). Mezu horrek dakarren informazioak, *QoSA* agenteari *BA_1* agentearen denbora-muga gainditzearen iragarpena zein agentek eragin duen (kasu honetan, *MA_1* agentea) identifikatzeko aukera ematen dio. Horrela, *QoSA* agenteak mezu bat bidaltzen dio *MA_1* agenteari (ikus *checkAgent* 4-12 irudian) eta jasotako erantzunaren arabera *D&DA* agenteari bi jakinarazpen desberdin bidal diezazkioke:

- Erantzunik jaso ezean, agente-akats motako gertaera baten berri ematen da.
- Erantzuna jasoaz gero, denbora-muga gainditze motako gertaera baten berri ematen da.



Irudia 4-12 *QoSA* agenteak denbora-muga gainditze motako gertaera nola egiaztatzen duen adierazten duen sekuentzia-diagrama.

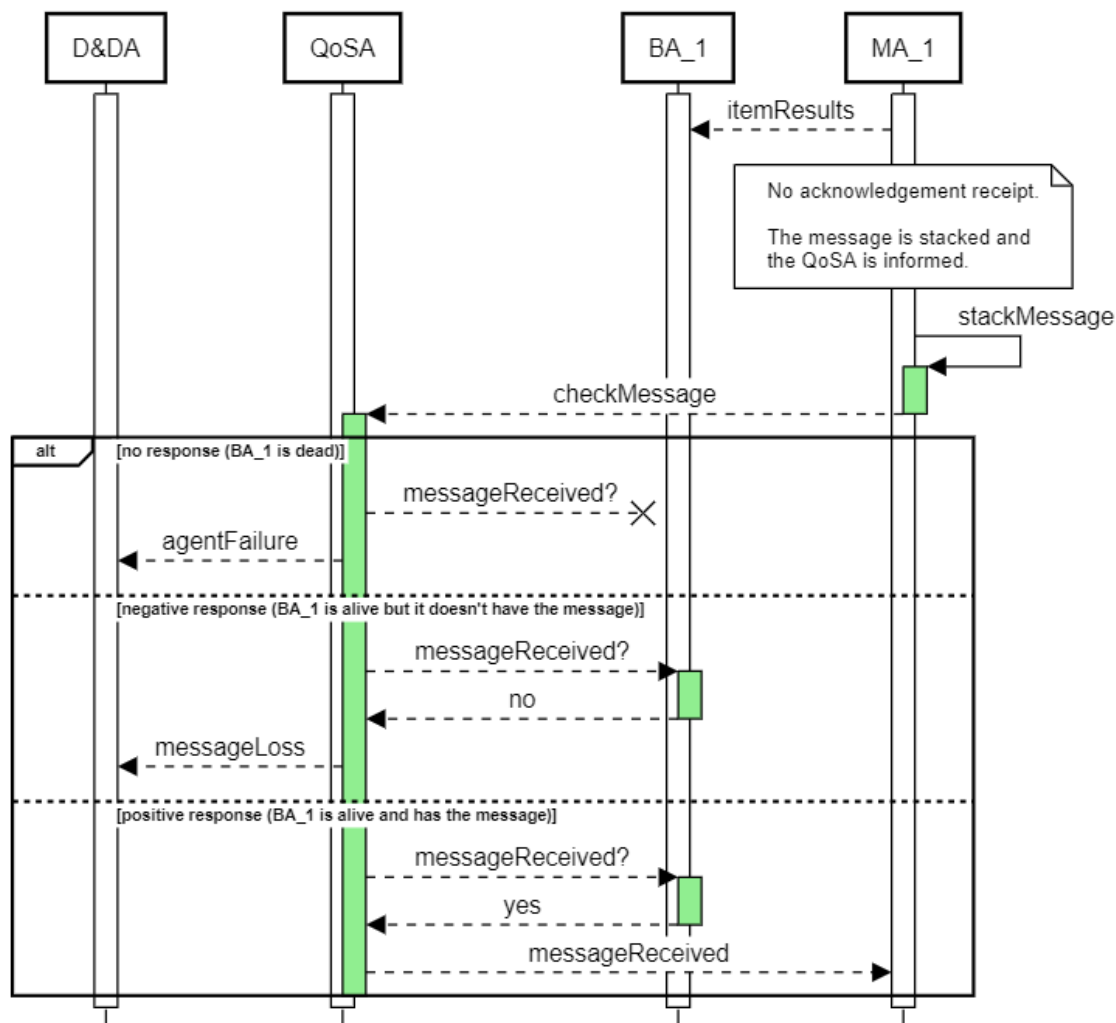
4.4.1.2 Mezu-galerak

Honelako gertaerak agente batek hartu-agiririk jasotzen ez duenean gertatzen dira. *RA* eta *AppA* agenteek gertaera horri lotutako mezuen bi buffer erregistro kudeatzen dituzte. Lehenengoak aldi baterako gordetzen ditu agenteak jasotako mezuak. Beraz, informazio hori eskuragarri dago *QoSA* agenteak mezuaren galera baieztatu edo ezeztatu dezan. Buffer erregistro horren tamaina erabiltzailearen beharretara egokitu daiteke. Bigarren buffer erregistroak beste agenteei bidalitako mezuak gordetzen ditu hartu-agiria jaso arte. Horrela, gertaera baieztatzen bada, mezuaren edukia gordetzen da.

Agente batek mezu baten galera antzematen duenean, *QoSA* agentea ohartarazten du eta mezuaren edukia eta hartzaileari buruzko informazioa ematen dio. Ondoren, *QoSA* agentea hartzailearekin harremanetan jartzen saiatzen da mezua jaso ote duen egiaztatzeko. Hiru aukera daude (ikus 4-13 irudia):

- *QoSA* agenteak ez du erantzunik jasotzen. Ondorioz, *D&DA* agenteari agente-akats motako gertaera bat jakinarazten zaio.
- *QoSA* agenteak hartzaileak mezua jaso ez duela adierazten duen erantzuna jasotzen du. Beraz, *D&DA* agenteari mezu-galera motako gertaera bat jakinarazten zaio.
- *QoSA* agenteak hartzaileak mezua jaso duela adierazten duen erantzuna jasotzen du. Kasu honetan, *QoSA* agenteak igorleari mezua jaso dela jakinarazten zaio eta *D&DA* agenteari ez zaio inolako gertaerarik jakinarazten.

Mekanismo honek *RA* eta *AppA* agenteei beraiei sistematik isolatuta gelditu diren diagnostikatzeko aukera ematen die. Lehenik eta behin, igorleak (kasu honetan, *MA_1* agenteak) hartzailearekiko (*BA_1* agentea) komunikazio-akatsa detektatzen du eta *QoSA* agentearekin harremanetan jartzen saiatzen da, gertaera jakinarazteko. *QoSA* agentearengandik erantzunik jaso ezean, *MA_1* agenteak isolatuta gelditu dela ondorioztatzen du. Erabiltzaileak funtzionalitateak gehitzeko aukera du arazo hau kudeatzeko. Adibidez, *MA_1* agentea isolatzen bada, bere egoera finituen makinaren *unavailable* egoerara pasa daiteke. Horrela, aurreikusitako zerbitzuen exekuzioa geldiarazten du, sistematik isolatuta dagoenez, arazo-iturri bat izan baitaiteke.



Irudia 4-13 QoSA agenteak mezu-galera motako gertaera nola egiaztatzen duen adierazten duen sekuentzia-diagrama.

4.4.1.3 Agente-akatsak

Izenak adierazten duen bezala, gertaera mota hau sistemako agente batek huts egin duela detektatzen denean jakinarazten da. Gertaera hori ez dute *RA* eta *AppA* agenteek zuzenean detektatzen, *QoSA* agenteak detektatzen du denbora-muga gainditze edo mezu-galera motako gertaerak egiaztatzen dituenean, ordea. Hala ere, aparteko gertaeratzat hartzen da, berari erantzuteko ekintzak desberdinak izan daitezkeelako. Aurreko kasuetan bezala, gertaera baieztatu ondoren, *D&DA* agenteari jakinarazten zaio.

4.4.2 D&DA agentea: gertaeren ebazpena

Azpiatal honetan *QoS*A agenteak jakinarazi diezazkiokeen gertaerei erantzuteko *D&DA* agentea nola egituratuta dagoen erakusten da. Kapitulu honen hasieran adierazi den bezala, gertaerak konpontzeko eta erresilientzia bermatzeko behar diren ekintzak aplikazioaren menpe daude. Hori kontuan hartuta, *D&DA* agenteak interfaze bat du, gertaera motei lotutako metodo pertsonalizagarriak eskaintzen dituena: *actions_timeouts*, *actions_messageLoss* eta *actions_agentFailure*, hurrenez hurren.

Agente-akatsa motako gertaerak kudeatzeko ekintzei dagokienez, agente motaren araberakoak dira:

- *RA* agenteen kasuan, aplikazioaren egoera berrezartzeko modua huts egin duen agentearen zerbitzuak beste *RA* agente batzuei berresleitzean datza. Berresleitze hau zerbitzu hauek egiteko gaitasuna duten *RA* agenteen arteko negoziazio baten bidez konpon daiteke, erabiltzaileak esleitutako irizpideen arabera.
- *AppA* agenteen kasuan, aldiz, erresilientzia *running* egoeran dauden *AppA* agenteen egoera mantentzen duten errepliken bidez bermatzen da. *AppA* agenteek beren aktiboak logikoak direlako erreplikak erabil ditzakete, eta, beraz, beren zerbitzuen exekuzioa kokapenarekiko independentea da. Horregatik, *AppA* agente aktibo batek (hau da, *running* egoeran dagoen agentea) huts egiten badu, bere erreplikek elkarrekin negoziatzen dute nork ordezkatzeko duen erabakitzeke. Bestalde, erreplika gisa funtzionatzen ari den *AppA* agente batek huts egiten badu, ahal den neurrian erredundantzia-maila bera mantentzeko beste erreplika bat sortuko da.

Ekintza hauen garapena eta pertsonalizazioa errazteko, *D&DA* agenteak agente mota bakoitzari lotutako beste hiru metodo ditu: *RA_failure*, *runningAppA_failure* eta *trackingAppA_failure*. Hiru metodo hauek *actions_agentFailure* metodoaren barruan erabil daitezke.

4.5 Laburpena eta ondorioak

Kapitulu honetan *IARMS I4.0* Plataformaren diseinua eta garapena landu dira. Sarreran, plataforma honek bete beharreko helburuak zerrendatzen dira, horiek nola lortuko diren azalduz. Soluzio bat proposatzen da 3. kapituluan aurkeztutako euskarri metodologikoan eta *Plattform Industrie 4.0* delakoak aintzat hartutako *Infrastructure Services* zerbitzuetan oinarrituta. Soluzio hau *FIPA* estandarra betetzen duten industria-agenteentzat diseinatu da, *ACL* protokoloaren potentziala agenteen berezko ezaugarriekin konbinatuz (negoziatzeko eta erabaki autonomoak hartzeko gaitasunak). Konbinazio hau guztiz aproposa da bai *I4.0 Component* osagaien arteko elkarreragintasunerako, bai *I4.0* Plataformak eskaintzen duen azpiegiturarako.

Ondorioz, *IARMS I4.0* Plataformak hainbat agente eskaintzen ditu, bi taldetan bana daitezkeenak: *skeleton patterns* txantioiak eta *I4.0 Platform core* nukleoa. *Skeleton patterns* txantiloiek 3.4 atalean aurkeztutako ikuspegia aplikatzen dute. *AAS* garatzeko oinarri bat ematen dute, *AAS Services* zerbitzuak hedatuz. Zerbitzu hauek barne-kudeaketan laguntzen dute osagai-mailan. *Skeleton patterns* txantiloiak *RA* eta *AppA* agenteetan banatzen dira, aktibo fisikoak eta logikoak integratzeko, hurrenez hurren, 3.2 atalean aurkeztutako eredu-egite ikuspegia jarraituz. Era berean, *I4.0 Platform core* nukleoa agente batzuk ditu *AAS Infrastructure Services* zerbitzuak (*SRA* eta *PLA* agenteak) eta *Plattform Industrie 4.0*k aurreikusitako konputazio gaitasuna (*PNA* agenteak) hornitzeko.

AAS Services (skeleton patterns txantiloietan ezarritakoak) eta *AAS Infrastructure Services (I4.0 Platform core nukleoan ezarritakoak)* zerbitzuen zeharkako garapenak, *AAS*en eta *I4.0 Platform core* nukleoaren arteko elkarreragingarritasunean laguntzen du, eta sistemaren kudeaketa baldintza normaletan bermatzen du. Hala ere, ez diete fabrikazio prozesuei erresilientziarik ematen. Hori dela eta, *I4.0 Platform core* nukleoa *QoSA* eta *D&DA* agenteak ere eskaintzen ditu fabrikazio-prozesuan ustekabeko gertaerak kudeatzeko. Horrela, *QoSA* agenteak hiru gertaera mota baieztatu edo baztertu ditzake: denbora-muga gainditzeak, mezu-galerak eta agente-

akatsak. *D&DA* agenteari dagokionez, gertaera hauei erantzuteko ekintzak abiarazten ditu.

Horretaz gain, *IARMS* I4.0 Plataformak *customization mechanisms* mekanismoak ere eskaintzen ditu, erabiltzaileak aplikazioaren menpe dauden alderdien garapena errazteko. Horri dagokionez, 3.3 atalean aurkeztutako aktiboak integratzeko ikuspegian oinarritutako soluzio bat eskaintzen da. *AAS*ak aktiboen komunikazio-gaitasunetatik abstraitzeko, soluzio honek agente laguntzaileak erabiltzea proposatzen du. Horrela, aktibo mota bati lotutako *AAS* guztiak *design pattern* txantiloia bakar batetik abiatuz ezar daitezke (*design pattern* txantiloia, *skeleton pattern* txantiloia aplikazioaren menpeko funtzionaltasunekin zabaltzen du). Gainera, proposamen hau IEEE 2660.1-2020 estandarrean proposatutako integrazio mota guztietan aplikagarria dela egiaztatu da. Bestetik, *D&DA* agenteak *customization mechanisms* mekanismoak ere eskaintzen ditu, erabiltzaileek beren aplikazioen eskakizunetara ondoen egokitzen diren ebazpen-ekintzak definitu ahal izateko. Honetarako, *D&DA* agenteak interfaze pertsonalizagarri bat eskaintzen du *QoSA* agenteak baieztatu ditzakeen gertaera motei lotutako metodoekin.

5 KONZEPTU-PROBA

5.1 Sarrera

Kapitulu honek *IARMS* I4.0 Plataformaren aplikagarritasuna egiaztatzea du helburu. Honetarako, errendimendua ebaluatzeko probak egin dira, bai baldintza normaletan, bai fabrikazio-prozesuan gertaerarik izanez gero. Proba horietarako 5.2 atalean aurkeztutako saiakuntza-plataforma erabili da. Atal honetan ere zehazten da nola hedatu zen I4.0 Plataforma saiakuntza-plataforman, fabrikazio-aktiboen integrazioa barne. Ondoren, 5.3 atalean, plataforma ebaluatzeko erabilitako proba-agertokiak zehazten dira, baita agertoki hauetako bakoitzari lotutako adierazleak ere. 5.4 atalean proba-agertokietan lortutako emaitzak aurkezten dira. 5.5 atalean emaitzak aztertzen dira plataformak diseinu-baldintzak betetzen ote dituen egiaztatzeko.

5.2 Saiakuntza-plataforma

Atal hau bi zatitan banatu da: lehenengoak kontzeptu-proba honetan erabilitako saiakuntza-plataforma azaltzen du, bere aktibo eta ezaugarri nagusiak deskribatuz eta *IARMS* I4.0 Plataforma nola hedatu den erakutsiz. Jarraian, 3.3.2 atalean aurkeztutako integrazio-metodologia nola aplikatu den azaltzen da, saiakuntza-plataformaren aktibo nagusia integratzeko.

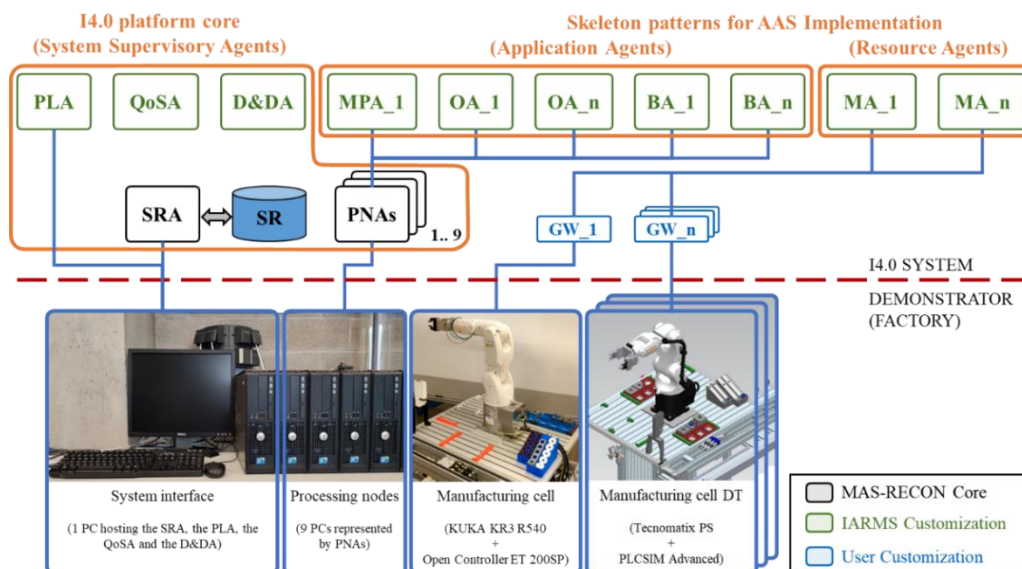
5.2.1 Saiakuntza-plataformaren ikuspegi orokorra

Kontzeptu-proba honetan erabiltzen den saiakuntza-plataformak fabrikazio-zelula bat du. Zelula honek muntaketa-eragiketak egiten ditu urratsez urratseko motor bat imitatzen duten 3Dn inprimatutako zati multzo batean. Muntaketa-eragiketa hauek tamaina aldakorreko sortetan prozesatzen dira, pieza batetik seira bitarteko multzoetan. Horretarako, fabrikazio-zelulak piezak maneiatzen dituen KUKA KR3 R540⁵ robot bat eta Siemens ET 200SP Open Controller⁶ kontrolagailu bat ditu. Kontrolagailu honek robotaren kontrola ahalbidetzen duen automata programagarri

⁵ KUKA KR3 R540 datasheet

⁶ Siemens ET 200SP Open Controller

(aurrerantzean, PLC, ingelesezko siglen arabera) ingurune bat, eta hirugarrenen softwarea hedatzeko aukera ematen duen Windows 10 ingurune bat ditu. Bi inguruneak isolatuta daude, baina datuak truka ditzakete memoria partekatuaren edo sareko interfaze birtualizatuen bidez. Probak malgutasun handiagoz egiteko, biki-digitalak (aurrerantzean, *DT*, ingelesezko siglen arabera) erabiltzen dira fabrikazio-zelula gehigarriak simulatzeko. *DT* hauek Tecnomatix Process Simulate⁷ eta PLCSIM Advanced⁸ software tresnen erabileran oinarritzen dira. Tecnomatix Process Simulate softwareak *DT*ak garatzea ahalbidetzen du, roboten ibilbideak simulatzea eta kontrol-kodea sortzea barne. Bestalde, PLCSIM Advanced softwareak TIA Portal⁹ garapen-ingurune integratua garatutako automatizazio-proiektu baten kodea simulazioan exekutatzeko aukera ematen du. Software hauen konbinazioak programazio eta interfaze berdinak dituzten edozein fabrikazio-zelula fisikoren *DT*ak sortzea ahalbidetzen du, beraz, *IARMS* I4.0 Plataformarentzat bereizezinak dira. Fabrikazio-zelulez gain, saiakuntza-plataformak hamar Dell Optiplex 760 ordenagailuz osotutako kluster bat ere badu, Ubuntu 20.04 sistema eragilearekin. Ordenagailu hauek prozesatze-nodo gisa jokatzeko dute, eta fabrikazio-prozesua kudeatzeko beharrezkoak diren *I4.0 Platform core* nukleoa eta *AppA* agenteak hedatzeko konputazio gaitasuna ematen dute.



Irudia 5-1 IARMS I4.0 Plataformaren hedapenaren xehetasuna saiakuntza-plataforman.

⁷ Tecnomatix Process Simulate

⁸ S7-PLCSIM Advanced

⁹ Totally Integrated Automation (TIA) Portal

Kontzeptu-proba honetan, *IARMS* I4.0 plataforma ezarri da *Java Agent DEvelopment framework (JADE¹⁰)* agente-plataforma erabiliz. *JADE* agente-plataformak *FIPA* estandarra betetzen duten agenteak kudeatzeko klaseak eta metodoak ematen ditu. 5-1 irudiak *IARMS* I4.0 Plataformaren hedapena erakusten du saiakuntza-plataforman. Alde batetik, klusterraren ordenagailuetako bat *SRA*, *PLA*, *QoSA* eta *D&DA* agenteak hedatzeko gorde da. Ordenagailu honek sistemarekiko interfaze gisa balio du. Klusterreko gainerako ordenagailuek prozesatze-nodo gisa jokatzen dute, bakoitza *PNA* agente batek ordezkatuta. Nodo hauek *BA*, *OA* eta *MPA* agenteak hedatzeko erabiltzen dira. Bestalde, fabrikazio-zelula fisikoren ordezkarria den *MA* agentea *ET 200SP Open Controller* kontrolagailuaren *Windows* ingurunean hedatu da. Simulatutako fabrikazio-zelulei dagokienez, *MA* agenteak *DTak* simulatzen diren ordenagailuetan hedatu dira. Kasu honetan, *GA* agenteak beren *MA* agenteen gailu beretan hedatu dira (*ET 200SP*-ak edo ordenagailuak, alegia), eta informazioa trukutzen dute beren *PLC* inguruneekin (benetakoak edo simulatuak) memoria partekatuaren bidez.

5.2.2 Fabrikazio-zelulen integrazioa

Azpiatal honetan zelula fabrikatzaileak (hala zelula fisikoa nola *DTak*) *IARMS* 4.0 plataforman nola integratu diren azaltzen da, 3.3.2 atalean aurkeztutako integrazio-metodologia aplikatuz.

5.2.2.1 Banaketa: aktibo erabilgarrien analisia

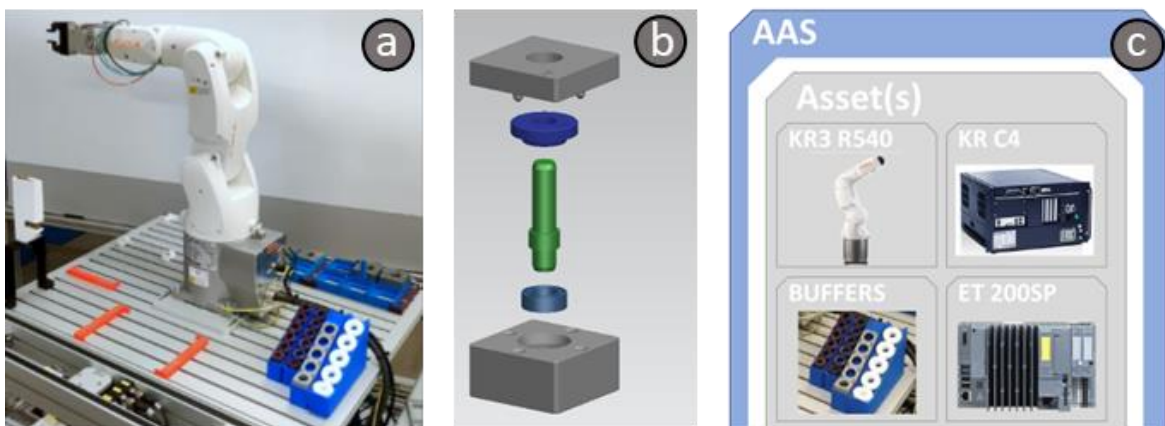
Lehenengo urratsa fabrikazio-zelulan dauden aktiboak identifikatzea da (ikus 5-2.a irudia). Aktibo nagusia *KUKA KR3 R540* robota da. Robot hau *KR C4 compact controller* kontrolagailu batek kontrolatzen du, eta azken hau, era berean, *ET 200 SP Open Controller* kontrolagailuak kudeatzen du. Lehen esan bezala, fabrikazio-zelula honek muntaketa-eragiketak egiten ditu bost zatiz osotutako multzo batean: oinarria, errodamendua, ardatza, barruko estalkia eta kanpoko estalkia (ikus 5-2.b irudia). Eragiketa hauek paleten gainean garraiatzen diren pieza bat eta sei arteko tamaina

¹⁰ <https://jade.tilab.com/>

aldakorreko sorteei aplikatzen zaizkie. Horretarako, robota hainbat bufferrez inguratutako lan-mahai batean kokatzen da, muntaia-eragiketak egiteko beharrezkoak diren material kontsumigarriekin. Horrez gain, lan-mahai honek sarrera-posizio bat du, lehengaiak dituzten paletak jasotzeko; lanpostu bat, muntaketa-eragiketak egiteko; eta irteera-posizio bat, muntaketa-eragiketak amaitu ondoren produktu bukatuak dituzten paletak kokatzeko. Material kontsumigarriak eta produktuak garraiatzeko erabiltzen diren paletak manipulatzeko, robotak sarrera/irteera digitalak dituen EtherCAT nodo baten bidez kontrolatzen diren bi matxarda ditu. Aktibo hauek *I4.0 Component* osagaiaren parte dira, 5-2.c irudian adierazten den bezala.

5.2.2.2 Zerbitzuen definizioa: geruzen arteko interfazeen zehaztapena

Bigarren urratsean, aktibo nagusiak eskaintzen dituen zerbitzuak eta geruzen arteko interfazeek eskaintzen dituzten metodoak definitu dira. Fabrikazio-zelulak produktuaren muntaketa osoa edo partziala egin dezake, lehengaietatik edo amaitu gabeko produktuetatik abiatuta. Honela, robotak hamar muntaketa-zerbitzu egin ditzake, *assembly_X(nOfItems)* deitzen direnak, non X letrak muntaketa mota adierazten du (*assembly_1(nOfItems)* zerbitzutik *assembly_10(nOfItems)* zerbitzura, ikus 5-1 taula).



Irudia 5-2 Xehetasuna: a) fabrikazio-zelula; b) produktuaren zatiak; c) ondoriozko *I4.0 Component* osagaia.

Taula 5-1 Fabrikazio-zelulak ematen dituen muntaketa-zerbitzuen xehetasuna.

| Muntaketa-zerbitzu mota | Materiala sarrera-posizioan | Eskatutako muntaketa |
|-------------------------|---|---|
| assembly_1(nOfItems) | oinarria | errodamendu |
| assembly_2(nOfItems) | | errodamendu + ardatza |
| assembly_3(nOfItems) | | errodamendu + ardatza + barruko estalkia |
| assembly_4(nOfItems) | | errodamendu + ardatza + barruko estalkia + kanpoko estalkia |
| assembly_5(nOfItems) | oinarria + errodamendu | ardatza |
| assembly_6(nOfItems) | | ardatza + barruko estalkia |
| assembly_7(nOfItems) | | ardatza + barruko estalkia + kanpoko estalkia |
| assembly_8(nOfItems) | oinarria + errodamendu + ardatza | barruko estalkia |
| assembly_9(nOfItems) | | barruko estalkia + kanpoko estalkia |
| assembly_10(nOfItems) | oinarria + errodamendu + ardatza + barruko estalkia | kanpoko estalkia |

Zerbitzu hauek pieza kopuru aldakor bat sortzeko muntaketa-eragiketak egiten dituzte. Beraz, sarrera-parametro bat behar da fabrikatu beharreko pieza-kopurua zehazteko. Zerbitzu-eskaeretan beharrezkoa den *core information* informazioa muntaketa-zerbitzuaren izenez eta pieza-kopurua parametroaren balioaz osotuta dago (adibidez, *assembly_4(6)*). Hauek *Asset Related Services* zerbitzuak direnez, geruzen arteko interakzioak behar dira. Horrela, *Functional Layer* geruzak *Operative Layer* geruzari sarrera-parametro bezala muntatu beharreko pieza kopurua duten metodoak eskaini behar dizkio. Trukean, bi seinale aktibatzen dira pieza bakoitzaren muntaketa hasi eta amaitzen denean. Bestalde, *Operative Layer* geruzak *Service Layer* geruzatik fabrikatu beharreko pieza kopurua sarrera-parametro gisa jasotzen du. Irteera gisa, zerbitzu osoaren eta pieza bakoitzaren hasiera- eta amaiera-denborak itzultzen ditu (*Functional Layer* geruzetik jasotako seinaleen oinarrituta), zerbitzua egin duen makinaren identifikadore batekin batera. Azkenik, *Service Layer* geruzak pieza baten amaiera-denbora jasotzen duen bakoitzean, pieza horri buruzko informazio guztia *Interoperability Layer* geruzara bideratzen du, geruza honek zerbitzu-eskatzaileari informazioa bidal diezaion.

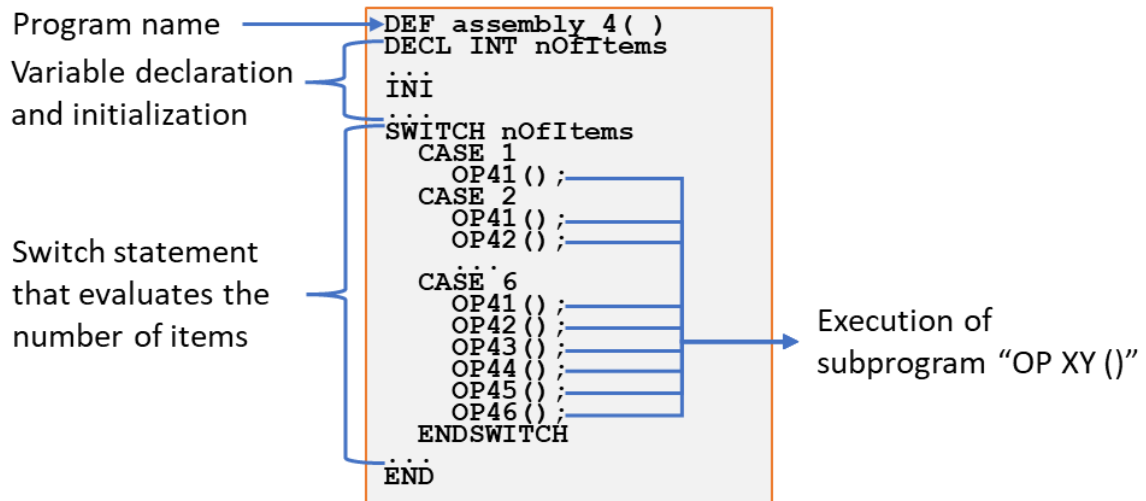
5.2.2.3 Hedapena: gailua eta teknologia aukeratzea

Hirugarren urratsa geruza bakoitzak ezartzeko behar dituen eskakizunen araberrako aktibo eta teknologia egokiak aukeratzea da. *Functional Layer* geruzak aktibo nagusiaren eragiketak kontrolatzeko atzipen zuzena izan behar du. Horregatik, KR C4 compact controller kontrolagailuan robot programa gisa ezarri da. *Operative Layer*

geruzari dagokionez, zerbitzu-eskaerak iragaztea, interpretatzea eta zerbitzua burutzeko beharrezkoak diren eragiketak egitea du helburu. Eragiketa guzti hauek datuen truke intentsiboa eta komunikazio-gaitasun onak eskatzen dituzte *Functional Layer* eta *Service Layer* geruzetarantz. Horregaitik, *Operative Layer* geruza ET 200SP Open Controller kontrolagailuaren PLC ingurunean automatizazio-proiektu gisa ezarri da. Gailu honek KR C4 compact controller kontrolagailurako (eta, beraz, *Functional Layer* geruza muntaketa-eragiketetarako) atzipena du, Profinet konexio baten bidez. Azkenik, *Service Layer* geruzak *Component Manager*-a ezarri behar du zerbitzu-eskaerak kudeatzeko, eta *Interoperability Layer* geruzak, berriz, horrelako eskaerak jaso, erantzun eta prozesatu egiten ditu. Horregaitik, bi geruzak ET 200SP Open Controller kontrolagailuaren Windows ingurunean hedatu dira, industria-agente baten bidez (kasu honetan, MA agente baten bidez). ET 200 SP Open Controller kontrolagailuaren bi inguruneen arteko komunikazioa memoria partekatuaren bidez ezarri da. Honek konputazio gaitasuna handia bermatzen du, PLC ingurunearen denbora errealeko eskakizunak arriskuan jarri gabe. Honi esker, *Operative Layer* eta *Functional Layer* geruzak sisteman AASen artean ezarritako komunikazio-saretik isolatu daitezke (Windows ingurunea sare horretara konektatuta dago, baina PLC ingurunea ez).

5.2.2.4 Garapena: behetik gorako ezarpena

Azken urratsean, geruza bakoitzerako zehaztutako metodoak ezarri dira. *Functional Layer* geruza robot programa nagusi gisa garatu da, 5-1 taulan adierazitako zerbitzuak ezartzen dituzten hainbat azpiprogramatan egituratuta dagoena. 5-3 irudiak azpiprograma hauen egitura erakusten du *assembly_4(nOfItems)* kasu konkreturako. Aldagaiak adierazteko eta abiarazteko hasierako zatia aurkezten du, eta ondoren, eskatutako item kopurua ebaluatzen duen baldintzapeko-egitura (ikus *SWITCH nOfItems* 5-3 irudian). Ondorioz, azpiprograma bat edo gehiago exekutatzeko dira. Azpiprograma horiek "OP XY ()" gisa identifikatzen dira, non "X" karaktereak zerbitzu-zenbakia adierazten du (kasu honetan, 4), eta "Y" karaktereak, berriz, piezak paletan duen posizioa.



Irudia 5-3 *Functional Layer* geruza ezartzen duen robotaren azpiprogrametako baten sasikodea.

Adibidez, eskatutako pieza kopurua seikoa bada (ikus *CASE 6* 5-3 irudian), honek `OP41()` azpiprogramatik `OP46()` azpiprogramara bitarteko azpiprogramak exekutatu direla, biak barne. Azpiprograma bakoitzean, robotak paletaren sei posizioetako bakoitzean muntaketa egingo du. Azpiprograma bakoitzaren hasieran eta amaieran, seinaleak aktibatzen dira muntaketa hasi edo amaitu dela adierazteko. Gainera muntatu den elementuen kopurua adierazteko zenbaki osoa ere itzultzen da.

Operative Layer gezurara igaroz, *Functional Layer* geruzaren metodoak deitzeko zein informazio trukatu behar den definitu da. ET 200SP Open Controller kontrolagailuaren (*Operative Layer* geruza) eta KR C4 compact controller kontrolagailuaren (*Functional Layer* geruza) artean trukaturako parametroak 5-2 taulan zerrendatzen dira. Ondoren, funtzio bat garatu da *Functional Layer* geruzatik jasotako emaitzak prozesatzeko. Funtzio honek elementuak hasiera- eta amaiera- denborak sortzen ditu, *itemStarted* eta *itemCompleted* seinaleak erabiliz (ikus 5-2 taulan) eta *Service Layer* geruzarekin elkarrengaitzeko erabiltzen diren datu-egituretan gordetzen ditu. Gainera, *itemCompleted* seinalea jasotzen denean, makinaren erreferentzia eta artikulu zenbakia datu-egitura horietan ere gordetzen dira.

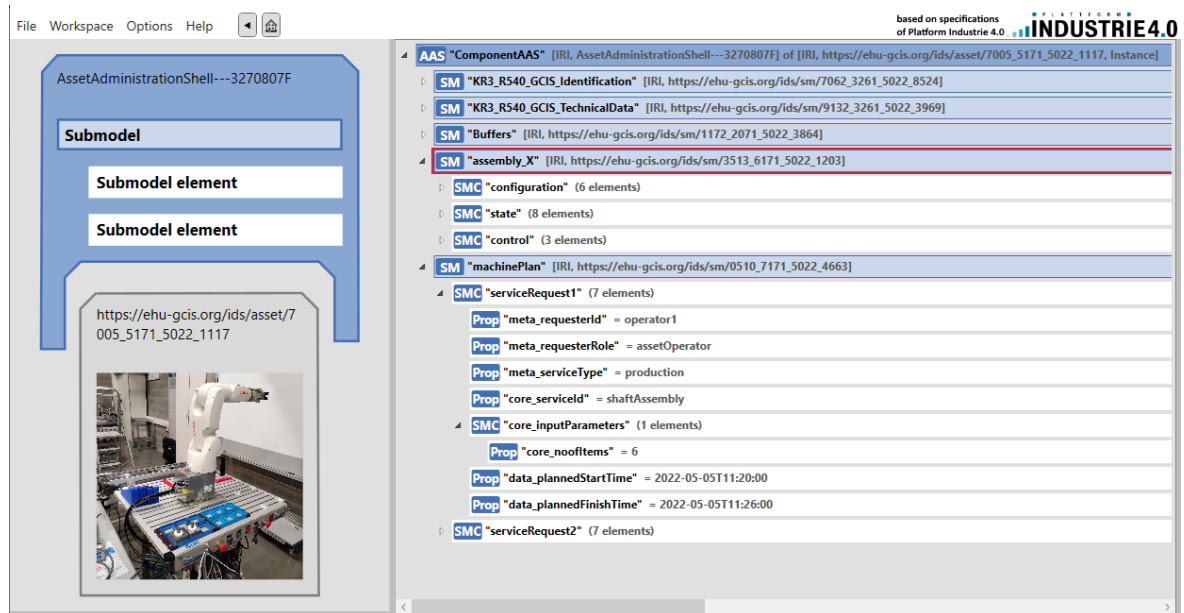
Taula 5-2 *Operative Layer* eta *Functional layer* geruzen artean trukaturako aldagaien zerrenda.

| Fabrikazioarekin lotutako seinaleak | | | |
|--|------|--------------|--|
| Izena | Mota | PLC helbidea | Deskribapena |
| numberOfItems | Int | %QB2247 | Muntatu beharreko piezen kopurua adierazten duen sarrera-parametroa. |
| executeOperation | Bool | %Q2248.0 | Eragiketa bat hasteko erabilitako aldagai osagarria, sarrera-parametroak baliozkotu ondoren. |
| itemStarted | Bool | %I2246.0 | Robotak pieza berri bat muntatzen hasi duela adierazten duen irteera-parametroa. |
| itemCompleted | Bool | %I2246.1 | Robotak pieza berri bat muntatzen amaitu duela adierazten duen irteera-parametroa. |
| operationStarted | Bool | %I2247.0 | Robotak muntaketa-eragiketa hasi duela adierazten duen irteera-parametroa. |
| operationCompleted | Bool | %I2247.1 | Robotak muntaketa-eragiketa amaitu duela adierazten duen irteera-parametroa. |
| itemcompletedNumber | Int | %IB2248 | Sortan osatutako pieza kopurua adierazten duen irteera-parametroa. |
| Aktiboaren kudeaketarekin lotutako seinaleak | | | |
| Izena | Mota | PLC helbidea | Deskribapena |
| \$EXT_START | Bool | %Q2001.2 | Robot-programa bat kanpotik abiarazteko aukera ematen duen seinalea. |
| \$MOVE_ENABLE | Bool | %Q2001.3 | Robotaren mugimenduak kanpotik gaitzeko aukera ematen duen seinalea. |
| \$CONF_MESS | Bool | %Q2001.4 | Mezu bat jaso dela baieztatzeko aukera ematen duen seinalea. |
| \$DRIVES_OFF | Bool | %Q2001.5 | Robotaren ardatzak aktibatzeke aukera ematen duen seinalea. |
| \$DRIVES_ON | Bool | %Q2001.6 | Robotaren ardatzak desaktibatzeke aukera ematen duen seinalea. |

Azkenik, *Service Layer* eta *Interoperability Layer* geruzak batera ezarri dira, AAS osatuz. Berriro ere, lehen urratsa beheko geruzarekin (*Operative Layer* geruza, alegia) datuak nola trukatu zehaztea izan zen. 5-3 taulan ikus daitekeen bezala, informazioa hiru datu-egituratan banatu da: *configuration*, zerbitzua eskatzeko behar den informazioa barne hartzen duena; *state*, emaitza gisa itzuliko den informazioa definitzen duena; eta *control*, *Service Layer* eta *Operative Layer* geruzen arteko interakzioak arautzeko erabilitako barne-adierazleak zehazten dituena. *Service Layer* eta *Operative Layer* geruzen arteko integrazioa ziurtatu ondoren, hurrengo urratsa AASaren azpi-ereduak definitzea izan zen. Kontzeptu-proba honi dagokionez, atal honetan deskribatutako zerbitzuak erakusten dituen azpi-eredu bat dago (*assembly_X submodel* azpi-eredua), 5-3 taulan aurkeztutako datu-egiturez osatua. Integrazio-prozesuan parte hartzen duten aktiboak erakusten dituzten azpi-ereduak ere badaude (robotak bera, bufferrak, etab.). Azkenik, *machinePlan submodel* azpi-eredu bat sartu da *Asset Related Services* zerbitzuei buruzko eskaerak erregistratzeko. Azpi-eredu honek bi funtzio betetzen ditu: batetik, *Service Layer* geruzari aukera ematen dio modu ordenatuan jasotzen dituen zerbitzu-eskaerak kudeatzeko; bestetik, *I4.0 Component* osagaien planifikatutako eragiketarak kontsultatzeko aukera ematen du. 5-4 irudiak ondoriozko AASa erakusten du.

Table 5-3 Zerbitzu-geruzen eta operazio-geruzen artean trukaturako datu-egiturak.

| Configuration | | |
|-----------------------|--------|-------------------------------------|
| Izena | Mota | Deskribapena |
| meta_requesterId | String | Eskatzailearen identifikadorea. |
| meta_requesterRole | String | Eskatzailearen rola. |
| core_serviceId | String | Zerbitzuaren erreferentzia. |
| core_noofftems | Int | Eskatutako pieza kopurua. |
| State | | |
| Izena | Mota | Deskribapena |
| meta_requesterId | String | Eskatzailearen identifikadorea. |
| meta_requesterRole | String | Eskatzailearen rola. |
| meta_machineReference | Int | Makinaren erreferentzia. |
| meta_itemNumber | Int | Osatutako pieza kopurua. |
| data_initialTimeStamp | Long | Piezaren muntaketa hasiera-denbora. |
| data_finalTimeStamp | Long | Piezaren muntaketa amaiera-denbora. |
| data_serviceTimeStamp | Long | Zerbitzuaren egite hasiera-denbora. |
| Control | | |
| Izena | Mota | Deskribapena |
| flag_NewService | bool | Zerbitzu-eskaera berria. |
| flag_ItemCompleted | bool | Pieza osatuaren adierazlea. |
| flag_ServiceCompleted | bool | Zerbitzu burutuaren adierazlea. |



Irudia 5-4 AASaren ikuspegi orokorra (AASX Package Explorer softwarearekin sortuta).

5.3 Proba-agertokiak definizioa

Atal honetan *IARMS* I4.0 Plataforma ebaluatzeko erabili diren proba-agertokiak aurkezten dira. Lehen azpiatalean, proba-agertoki batzuk eta *Key Performance Indicators (KPI)* adierazleak modu generikoan proposatzen dira (hau da, *IARMS* I4.0 Plataforman ez ezik, edozein I4.0 Plataforma ebaluatzeko erabiliak izan daitezzen). Azpimarratzekoa da azpiatal honetan aurkeztutako lana Research Centre in Digitalization and Intelligent Robotics (CeDRI¹¹) ikerketa-zentroko ikertzaileekin egindako lankidetzaren emaitza dela, Polytechnic Institute of Bragança erakundearen hiru hilabeteko egonaldia egin ondoren. Jarraian, bigarren azpiatalak 5.2 atalean deskribatutako saiakuntza-plataforman proba-agertokiak nola aplikatu diren erakusten du.

5.3.1 Proba-agertoki eta *KPI* orokorrak

Lan honetan aurkeztutako doktorego-tesiaren garapenean zehar lankidetzaren estua izan da CeDRI ikerketa-zentroko ikertzaileekin. Egonaldi aurretik eta egonaldian zehar egindako bilera batzuetan, I4.0 Plataformen ebaluazio konparatiboa erraztuko luketean soluzioak landu ziren. Ondorioz, 5-4 taulan agertzen diren sei proba-agertokiak definitu dira. Agertoki hauek I4.0 Plataforma baten errendimendua ebaluatzeko pentsatuta daude malgutasunaren ikuspegitik. Hala ere, interesgarriak izan daitezkeen beste alderdi batzuk, hala nola eraginkortasuna edo segurtasuna, ebaluatzeko agertoki gehiago gehitu beharko liriateke. I4.0 Plataformen errendimendua neurtzeko, agertoki bakoitzak hainbat *KPI* izan ditzake (ikus 5-5 taula).

¹¹ <https://cedri.ipb.pt/>

Taula 5-4 CeDRI ikerketa-zentroko ikertzaileekin lankidetzan proposatutako proba-agertokiak.

| Agertokia | Deskribapena | Portaera nabarmena | KPI |
|-----------|--|--|----------------|
| #0 | Dena ondo dabilen erreferentziako egoera. | Sistemaren gaitasun operatibo normala. Helburua erreferentziako parametroak kalkulatzeko da, beste egoera batzuekin alderatzeko. | OEE, MLT |
| #1 | Halako batean, premiazko eskaera bat agertzen da. | Sistemak premiazko eskaerak kudeatzeko eta eragiketarako birbanatzeko duen gaitasuna. | TDT |
| #2 | Halako batean, sistemari makina berri bat gehitzen zaio, eta berehala erabilgarri dago produkzioarako. | Sistemak fabrikaren topologiaren bilakaerara egokitzeko duen gaitasuna. | TTO |
| #3 | Halako batean, makina erredundante batek huts egiten du. | Makina erredundante baten matxura kudeatzeko sistemaren gaitasuna. | OEE, MLT, TTDE |
| #4 | Halako batean, makina baten prozesatze-denbora handitzen da eragiketa guztietarako. | Sistemak makina baten prozesatze-denboraren aldaketara egokitzeko duen gaitasuna. | OEE, MLT, TTDE |
| #5 | Halako batean, entitate erabakitzzaileen arteko komunikazioari euskarri ematen dien sarea erortzen da. | Sistemak sare erabakitzzaileen galera kudeatzeko duen gaitasuna. | OEE, MLT, TTDE |

OEE - Overall Equipment Effectiveness; MLT - Manufacturing Lead Time; TDT - Task Designation Time; TTO - Time To Operation; TTDE - Time to Detect the Event.

Taula 5-5 CeDRI ikerketa-zentroko ikertzaileekin lankidetzan proposatutako KPIen deskribapena.

| KPI | Deskribapena | Formula |
|------|--|-----------------------------------|
| OEE | Baliabide baten eraginkortasuna neurtzen du, baliabidea operatiboa dagoen ala ez (<i>Availability, A</i>), aldi batean fabrikatutako produktuen kopurua (<i>Performance, P</i>) eta kalitatea (<i>Quality, Q</i>) kontuan hartuta. | $OEE = A * P * Q$ $OEE \leq 1$ |
| MLT | Agindua eskatu zenetik (<i>Order Time, OT</i>) prozesua amaitu arte (<i>Completion Time, CT</i>) igarotako denbora kalkulatzeko denbora-adierazlea. | $MLT = CT - OT$ |
| TDT | Ezusteko gertakariren batek egungo fabrikazio-plana egikaritzea eragozten duenetik (<i>Event Time, ET</i>) eragiketa horiek birbanatzen diren arte (<i>Redistribution Time, RT</i>) igarotako denbora kalkulatzeko denbora-adierazlea. | $TDT = RT - ET$ |
| TTO | Baliabide berri bat sistemara konektatzen denetik (<i>Plugging Time, PT</i>) zerbitzuak egiteko erabilgarri dagoen arte (<i>Resource Ready Time, RRT</i>) igarotako denbora kalkulatzeko denbora-adierazlea. | $TTO = RRT - PT$ |
| TTDE | Fabrikazio-prozesuari eragiten dion gertaera bat jazotzen denetik (<i>Event Time, ET</i>) sistemak gertaera detektatu arte (<i>Detection Time, DT</i>) igarotako denbora kalkulatzeko denbora-adierazlea. | $TTDE = DT - ET$ |

5.3.2 Proba-agertokiak eta KPI pertsonalizatuak

Azpiatal honetan kontzeptu-proba honen proba-agertokiak eta KPIak aurkezten dira, aurreko azpiatalean aurkeztutako agertoki eta KPI generikoetan kokatuz. Hiru proba-agertoki zehaztu dira:

- Lehenengoak plataformaren *RA* eta *AppA* agenteen hedapen-denbora (martxan jartzen direnetik erabat operazionalak diren arte igarotako denbora) baldintza normaletan ebaluatzen du.
- Bigarrenak *RA* agente baten (zehazki, *MA* agente baten) akatsak eragindako gertaera baten kudeaketa eta fabrikazio-aplikazioaren egoera berrezartzeko modua ditu ardatz.
- Hirugarrenak prozesatze-nodo batek huts egiten duen kasua aztertzen du, *AppA* agente batzuren akatsa eraginez eta aplikazioaren aurreko egoera berreskuratzeko ekintzak abiaraziz.

Hiru agertoki horiek jarraian deskribatuko ditugu.

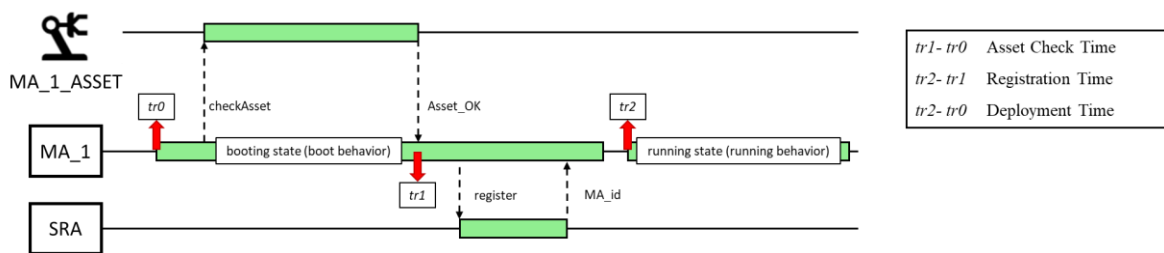
5.3.2.1 Hedapen-proba

Proba honetan, *RA* eta *AppA* agenteen hedapen-denbora ebaluatzen da. Zehazki, proba honen interesa agenteen hedapen-denboren arteko desberdintasuna arrazoitzean datza, zenbatekoei, zentzuzkoak diruditen bitartean, arreta berezirik jarri gabe. Hedapen prozesu honetan gertatzen denari buruzko xehetasun gehiago izateko, neurketa bereiziak lortzea erabaki da, 4.3 atalean azaldu zen bezala, *RA* eta *AppA* agenteek martxan jartzeko prozesu desberdinak jarraitzen baitituzte:

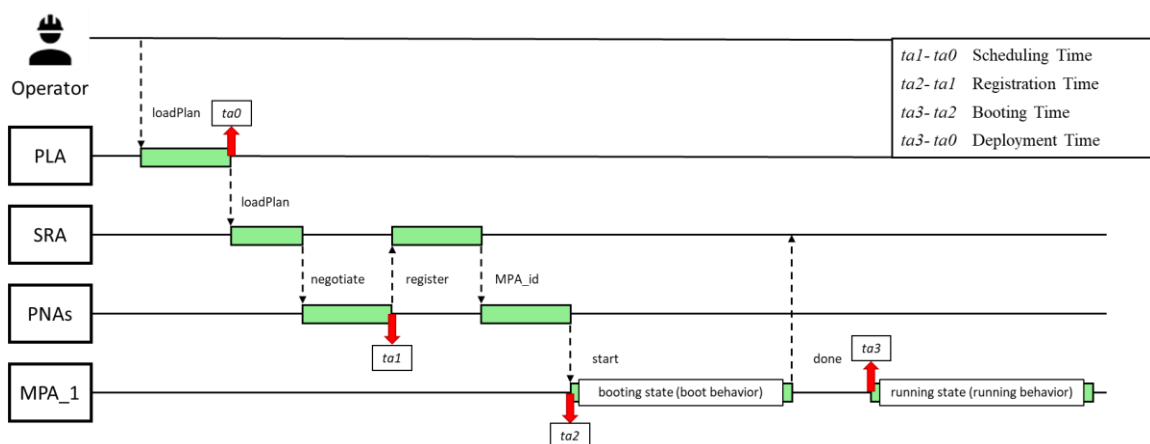
- *RA* agenteen kasuan (*MA_1* agentearen kasurako 5-5 irudian adierazita), *tr0* denbora-marka *AASa* sortzen den unean neurtzen da. Jarraian, *tr1* denbora-marka aktiboarekiko komunikazioa zuzena dela baieztatzen denean neurtzen da. Azkenik, *tr2* denbora-markak *AASa running* egoeran sartzten den unea adierazten du. Denbora-marka hauekin, hiru denbora-tarte zehaztu ziren: *asset check time* ($tr1-tr0$), *registration time* ($tr2-tr1$) eta *deployment time* ($tr2-tr0$).
- *AppA* agenteen kasuan (*MPA_1* agentearen kasurako 5-6 irudian adierazita), *ta0* denbora-markak erabiltzaileak *PLA* agentearen bidez fabrikazio-plan bat hedatzea eskatu duen unea adierazten du. Eskaera honek negoziazio bat hasten du *PNA* agenteen artean, *AAS* bakoitza zein prozesatze-nodotan hedatzen den zehazteko. *ta1* denbora-marka negoziazioa amaitzen denean neurtzen da (hau

da, AASa zein prozesatze-nodotan hedatuko den erabakitzen denean). Ondoren, ta_2 denbora-marka *boot* egoeraren hasieran neurtzen da, AASa jada sortu dela jakiteko. Azkenik, ta_3 AASa *running* egoeran sartzen denean neurtzen da. Denbora-marka hauekin, lau denbora-tarte zehaztu ziren: *scheduling time* (ta_1-ta_0), *registration time* (ta_2-ta_1), *booting time* (ta_3-ta_2) eta *deployment time* (ta_3-ta_0).

Hedapen-proba hau fabrikazio-aktiboek (fisikoak zein logikoak) beren AASen bidez (RA eta AppA agenteetan oinarrituta, hurrenez hurren) erabilgarri eta sistemara konektatuta egoteko behar duten denboran zentratzen da. Beraz, proba-agertoki 5-4 taulan #2 bezela izendatu den agertoki generikoari dagokio. Agertoki honi lotutako KPIa *Time To Operation (TTO)* da, eta haren definizioa proba honetan kalkulaturako hedapen-denboretara nahiko ondo egokitzen da (ikus 5-5 taula).



Irudia 5-5 Hedapen-proban MA agenteetan oinarritutako AASetarako bildutako denbora-markak adierazten dituen denbora-diagrama.

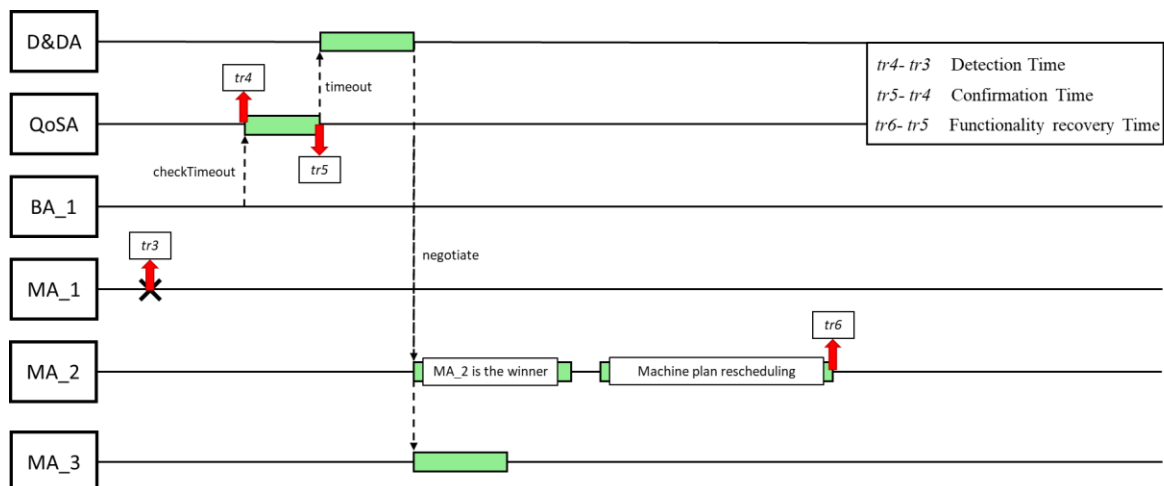


Irudia 5-6 Hedapen-proban MPA agenteentzat bildutako denbora-markak adierazten dituen denbora-diagrama.

5.3.2.2 Makinaren akats-proba

Proba honen helburua zerbitzuak burutzeke dauzkan RA agente baten galeraren aurrean IARMS I4.0 Plataformak duen erantzuna egiaztatzea da. Kasu honetan, makina baten akatsa simulatzeko, fabrikazio zelula simulatu bati (hau da, DT bat) lotutako MA agente bat modu kontrolatuan ezabatuko da. 4.4.2 azpiatalean azaltzen den bezala, RA agente baten akatsak berekin dakar burutzeke dauden zerbitzuak egiteko gaitasuna duten gainerako RA agenteen arteko negoziazioa, zerbitzu horiek negoziazioaren irabazleari berresleitzen zaizkiolarik.

Aurreko kasuan bezala, hainbat neurketa bereizi gehitu dira gertaera detektatzeko, berresteko eta ebazteko prozesu honetan, 5-7 irudian ikus daitekeen bezala. Lehenik eta behin, *tr3* denbora-markak MA agentearen ezabatze-unea adierazten du. Ondoren, *tr4* denbora-marka QoSA agenteari gertaera baten berri ematen zaion uneari dagokio, eta *tr5* denbora-marka, gertaera hau egiaztatzen den uneari. Azkenik, *tr6* denbora-marka aplikazioaren egoera operatiboa berreskuratu denean neurtzen da. Une hau MA agenteek beren artean negoziatu eta MA agente irabazleak bere makina-planean esleitutako zerbitzuak sartu ondoren gertatzen da.



Irudia 5-7 Makinaren akats-proban jasotako denbora-markak adierazten dituen denbora-diagrama.

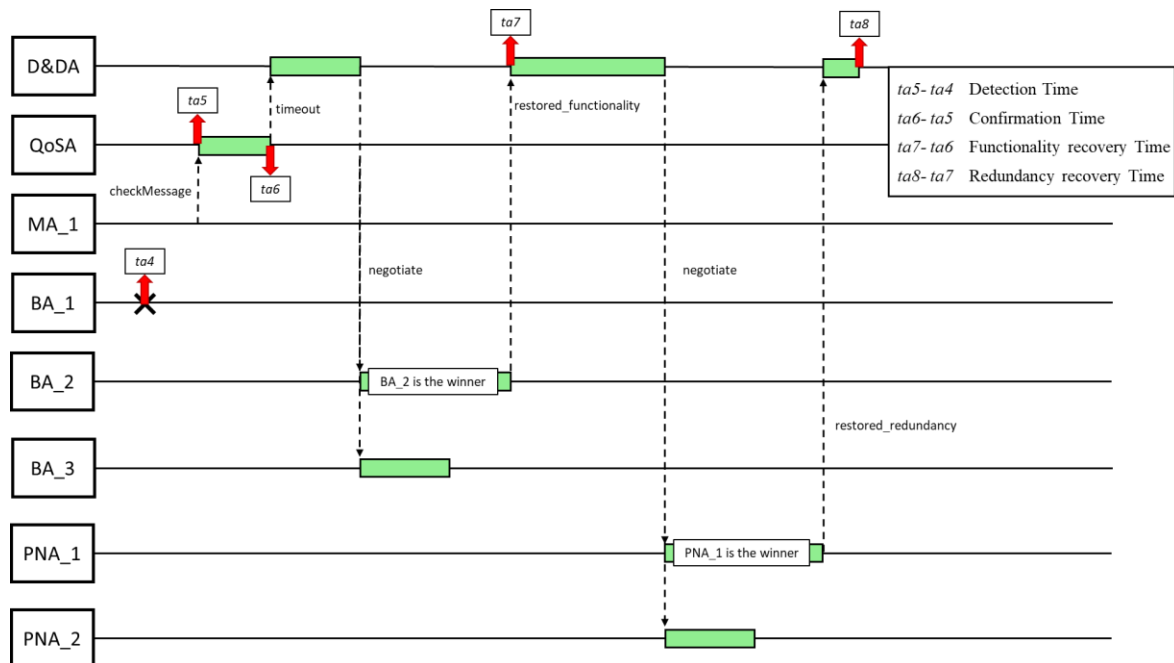
Proba-agertoki hau 5-4 taulan #3 izenez izendatu den agertoki generikoari dagokio. Hala ere, ez dator guztiz bat agertoki ebaluatzeko hautatutako *KPI*ekin. Agertoki hau ez da produkzio *KPI*etan oinarritzen, hala nola *Overall Equipment Effectiveness (OEE)* eta *Manufacturing Lead Time (MLT)*, *KPI* hauek *IAMS I4.0* Plataformaren errendimenduaren menpe baino gehiago saiakuntza-plataformaren menpe baitaudelako. Aldiz, agertoki hau nagusiki plataformaren menpe dauden alderdietan zentratzen da, hala nola *QoSA* eta *D&DA* agenteen funtzionamenduan edo negoziazio-mekanismoan. Honek gertaeraren detekzioa barne hartzen du, agertoki horretako hirugarren *KPI*ari dagokiona, hots, *Time To Detect the Event (TTDE)*.

5.3.2.3 Prozesatze-nodoaren akats-proba

Proba honek aurrekoa osatzen du, plataformak prozesatze-nodo bateko akats baten aurrean nola erantzuten duen egiaztatuz, eta *AppA* agente baten edo batzuren akatsa eraginez. Honek negoziazio bat edo bi eragingo ditu huts egindako agente bakoitzeko, beraien egoeraren arabera:

- *Running* egoeran zeuden *AppA* agenteek bi negoziazio behar dituzte: lehenengoa, bere erreplikaren bat *running* egoerara pasatzeko, eta bigarrena, ahal bada, erredundantzia-mailari eusten dion erreplika berri bat sortzeko. Aplikazioaren egoera ahalik eta azkarren berrezartzeko, negoziazio hauek ordena honetan ebatzi behar dira.
- *Tracking* egoeran zeuden *AppA* agenteek bigarren negoziazioa baino ez dute egiten (erredundantzia-mailari eusten diona).

Bigarren kasua lehenengoan jasota dagoenez, agertoki honek *running* egoeran dauden *AppA* agenteak ditu ardatz. 5-8 irudiak prozesuan zehar egiten diren neurri bereziak zehazten ditu. *ta4* denbora-markak probaren hasiera adierazten du, eta prozesatze-nodoan akatsa gertatzen den unean neurtzen da. Jarraian, *QoSA* agenteak *ta5* eta *ta6* denbora-markak erregistratzen ditu, gertaera berrikusteko eskaera jasotzean eta gertaera baieztatzean, hurrenez hurren. Azken bi denbora-markak, *ta7* eta *ta8*, funtzionaltasuna (erreplika bat *running* egoerara pasatu da) eta erredundantzia (erreplika berri bat sortu da) berrezartzen direnean erregistratzen dira.



Irudia 5-8 Prozesatze-nodoaren akats-proban jasotako denbora-markak adierazten dituen denbora-diagrama.

Proba-agertoki hau 5-4 taulako #5 bezala izendatu den agertoki generikoari dagokio, nodo baten akatsak erreferentziazko agertoki horretan aintzat hartutako sare-akatsaren antzekoa baita. Makinaren akats-probarekin gertatzen zen bezala, agertoki honetan egindako neurriak *TTDE KPI*arekin lotuta daude (ikus 5-5 taula), baina ez da *OEE* edo *MLT KPI*ekin lotutako informazioirik neurtzen.

5.4 Emaitzak eta eztabaida

Hurrengo azpiataletan 5.3.2 azpiatalean deskribatutako hiru proba-agertokiak gauzatu ondoren lortutako emaitzak jasotzen dira.

5.4.1 Hedapen-proba

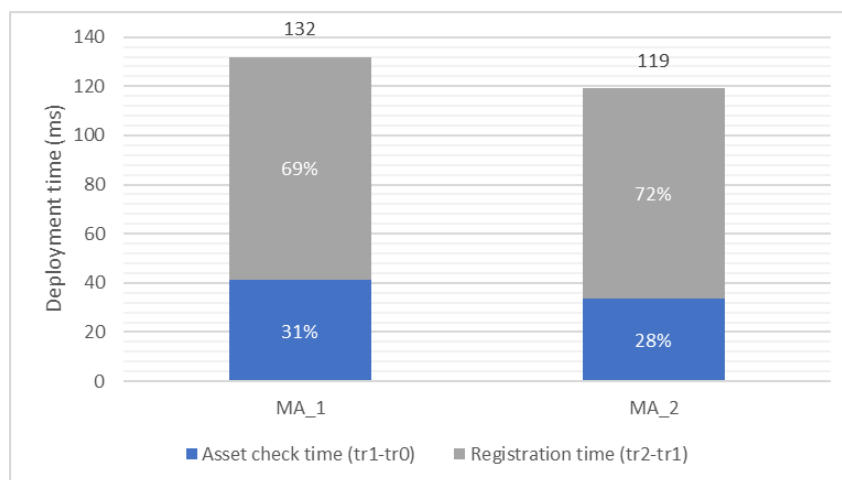
Proba-agertoki honetarako, *manufacturingPlan* batean, *order* batean eta *batch* bitan (makina banari esleitutakoak) egituratutako fabrikazio-plan bat erabili da. Proba-agertoki hau 20 aldiz errepikatu zen datuen banaketa estatistiko egokia bermatzeko. AASen hedapen-denboren emaitzak 5-6 taulan laburbiltzen dira.

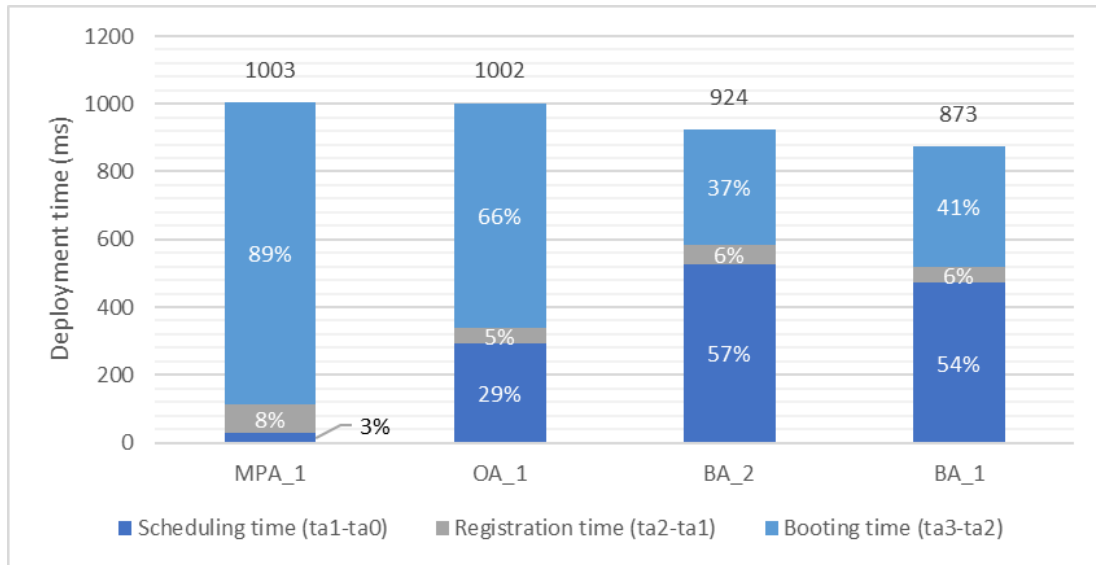
Taula 5-6 Hedapen-proba egin ondoren lortutako batezbesteko hedapen-denbora (N=20).

| AAS | Batezbesteko hedapen-denbora (ms) |
|-------|-----------------------------------|
| MA_1 | 132 |
| MA_2 | 119 |
| MPA_1 | 1003 |
| OA_1 | 1002 |
| BA_1 | 924 |
| BA_2 | 873 |

Ikus daitekeenez, *MA* eta *AppA* agenteetan oinarritutako *AAS*en hedapen-denboraren artean alde nabarmena dago: 125 ms inguru *MA* agenteentzako, eta 873 ms eta 1003 ms *BA_2* eta *MPA_1* agenteentzat, hurrenez hurren. Alde honek hurrengo azalpena du: *RA* agenteetan oinarritutako *AAS*en kasuan (hots, proba honetako *MA* agenteak), hedapen-prozesua zuzenean egiten da, aktiboa bizirik dagoela egiaztatzeaz gain, ez baita beste ekintzarik burutu behar. Aldiz, *AppA* agenteetan oinarritutako *AAS*en kasuan, hedapen-prozesua baldintzatuta dago, *AAS* bakoitzaren hedapen-kokapena erabakitzeke egiten den *PNA* agenteen arteko negoziazio-prozesuak baldintzatzen baitu. Gainera, *MPA*, *OA* eta *BA* agenteen arteko harreman hierarkikoak haien hedapen-denborak ere baldintzatzen ditu.

Desberdintasun horiek hobeto ulertzeko, 5-9 eta 5-10 irudiek *AAS* mota bakoitzaren hedapen-prozesua zehaztatzen dute. 5-9 irudiak *MA* agenteetan oinarritutako *AAS*en hedapen-denbora erakusten du, tartetean banatuta. Bi kasuek antzeko erantzuna erakusten dute: denboraren % 30 inguru ematen dute aktiboa eskuragarri dagoela baieztatzen, eta gainerakoa *SR* biltegian *AAS*a erregistratzen.

**Irudia 5-9 MA agenteetan oinarritutako AASetarako hedapen-denbora bereziak.**



Irudia 5-10 AppA agenteetan oinarritutako AASentzako hedapen-denbora bereziak.

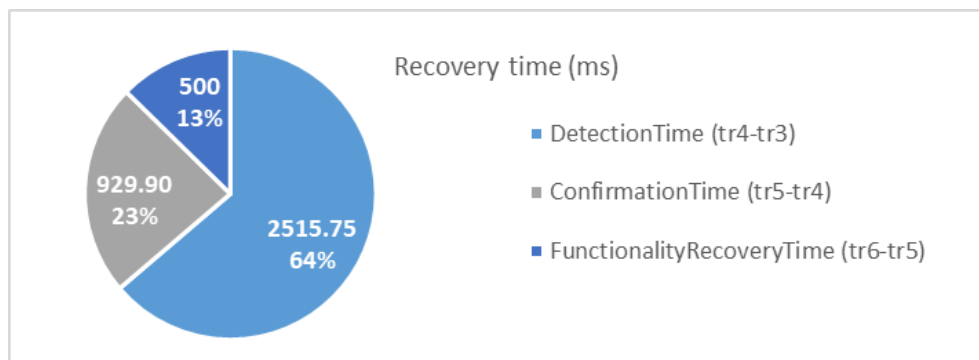
5-10 irudiari dagokionez, AppA agenteetan oinarritutako AASentzat lortutako emaitzak erakusten ditu, 4-10 irudian erakutsitako hedapen-sekuentzia egiaztatzeko aukera ematen dutenak. Lehenik eta behin, hedapen-denbora handiena MPA_1 agenteari dagokio (1003 ms), ondoren OA_1 agenteari (1002 ms) eta, azkenik, BA_2 (924 ms) eta BA_1 (873 ms) agentei. Honek zentzua du, MPA_1 agenteak ez baitu bere boot egoera amaitzen OA_1 agentearen baieztapena jaso arte, eta gauza bera gertatzen zaio OA_1 agenteari BA_2 eta BA_1 agentei dagokienez. Abiarazte kate-prozesu honek AAS hauetako denbora-tarteen arteko aldeak ere azaltzen ditu. MPA_1 agentearen kasuan, scheduling time denbora-tartea minimoa da (hedapen-denboraren % 3 baino ez), egiten den lehen negoziazioa baita. Registering time denbora-tartea ere laburra da (guztizkoaren % 8); izan ere, MPA_1 agentea da SRA agentearekin elkarrengaitan duen lehena, erantzuteko denbora minimizatuz. Bestalde, booting time denbora-tartea hedapen-denboraren % 89 da, ez denbora hau bere abiarazte-lan guztiak egiteko behar duelako, baizik eta OA_1 agentearen baieztapena jasotzeko zain dagoelako. Honen froga da MPA_1 agentearen batezbesteko hedapen-denbora OA_1 agentearena baino 1 ms luzeagoa ez dela: baieztapen-mezua jaso bezain laster, MPA_1 agentea running egoerara pasatzen da. Hierarkian behera aurrera goazela, scheduling time denbora-tarteak gora egiten du, izan ere, AAS bakoitzak hartu behar duen nodoa zehazteko negoziazioak sekuentzialki ebazten dira. Beraz, ta0 denbora-markaz geroztik igarotako denbora gero eta handiagoa da (kontuan izan behar da OA_1, BA_1 eta BA_2 agenteen ta0 denbora-marka MPA_1 agentearen denbora-marka bera dela, baina ta1

denbora-markak kasu bakoitzerako desberdinak direla). Aldi berean, *booting time* denbora-tarteak gero eta txikiagoak dira; izan ere, *OA_1* agenteak *BA_1* eta *BA_2* agenteen baieztapenaren zain egon behar du, eta agente hauek abiarazte-lanak zuzenean egiten dituzte.

5.4.2 Makinaren akats-proba

Proba-agertoki hau egiteko, bi makina baino gehiago erabili behar dira. Beraz, makina batek huts eginez gero, sisteman negoziatzeko bi makina geratzen dira. Horregatik, proba-agertoki honetarako, hedapen-probako agertokian erabilitako konfigurazio bera mantendu zen, eta sistemari esleitutako zereginik gabeko hirugarren makina bat gehitu zitzaion. Une jakin batean, makina hauetako bati lotutako *MA* agentea urrunetik ezabatu zen (ikus 5-1 irudia) sistemaren interfazearen ordenagailutik horretarako sortutako Java klase bat erabiliz. Gertaera *BA* agente batek detektatu zuen, denbora-muga gainditze bat iragarritz, matxuratutako makinatik espero zuen informazioa garaiz jaso ez zuelako. Berrito, proba-agertoki hau 20 aldiz egin zen datuen banaketa estatistiko egokia bermatzeko. Proba-agertoki honen emaitzak 5-11 irudian agertzen dira.

Irudian ikusten den lehenengo gauza denbora gehiena *detection time* denbora-tarteari dagokiola da, hau da, gertaera gertatu zenetik (*tr3*) *QoSA* agenteari jakinarazi zitzaion arte (*tr4*) igarotako denborari. Izan ere, kasu honetan gertaera detektatzen duen lehena *BA* agentea da, huts egindako *MA* agentearen informazioa jasotzea espero baitzuen. Ordura arte, gertaera ezkutuan zegoen, gaur egun ez baita *heartbeat* seinaleen erabilera bezalako egiaztapen dinamikoko sistemarik ezarri. Gertaera egiaztatze eskaera jaso ondoren, *QoSA* agenteak akatsa baieztatze eta dagokion abisua *D&DA* agenteari bidaltzeko (*confirmation time* denbora-tartea) batezbeste 1 s baino gutxiago behar du. Aldiz, gertaeraren ebazpena (*functionality recovery time* denbora-tartea) abiadura handian gertatzen da, negoziazioak eta ondorengo *MA* agente irabazlearen makina-planaren birplangintzak 0.5 s inguru irauten dute.



Irudia 5-11 Makinaren akatsa berreskuratzeko denbora bereiziak.

5.4.3 Prozesatze-nodoaren akats-proba

Proba-agertoki hau atal honetan aurkeztu diren hiru proba-agertokietako luzeena eta zehatzena da. Alde batetik, prozesatze-nodo baten akatsaren (azken batean, *AppA* agente-akatsa) aurrean emandako erantzuna baliozkotzeko erabili da. Bestalde, nodo-eta erreplika-kopuruak fabrikazio-aplikazioen hedapenean eta kudeaketan duten eragina aztertzeko erabili da.

Oraingo honetan, kargatutako fabrikazio-planaren egitura konplexuagoa da: *manufacturingPlan* bat, bi *order* eta hiru *batchez* osotuta, azken hauek makina banari esleituta. Fabrikazio-plan hau hainbat nodo eta erredundantzia-mailarekin hedatu da. Beraz, konbinazioak kasu mugatzaileenetik, hiru prozesatze-nodorekin eta *AppA* agente bakoitzaren erreplika bakarrarekin (*running* egoeran dagoen instantzia), hain mugatzailea ez den kasuraino, prozesamenduko bederatzi nodorekin eta bederatzi erreplikarekin (bat *running* egoeran eta gainerako zortziak *tracking* egoeran), doaz.

Proben agertokia honela gauzatu zen: fabrikazio-plana behar bezala eta arazorik gabe kargatu eta hasi zen. Halako batean, prozesatze-nodo bat sistematik ezabatu zen, nodo horretan zeuden agente guztiekin batera. Gertaera hau *MA* agente batek detektatu zuen *BA* agente bati zerbitzu baten emaitzak bidaltzen saiatu zenean, baina ez zuen erantzunik jaso, eta une horretan *QoSA* agenteari arazoaren berri eman zion. Beste behin ere, proba 20 aldiz errepikatu zen, eta exekuzio horien batezbesteko emaitzak 5-7 taulan agertzen dira.

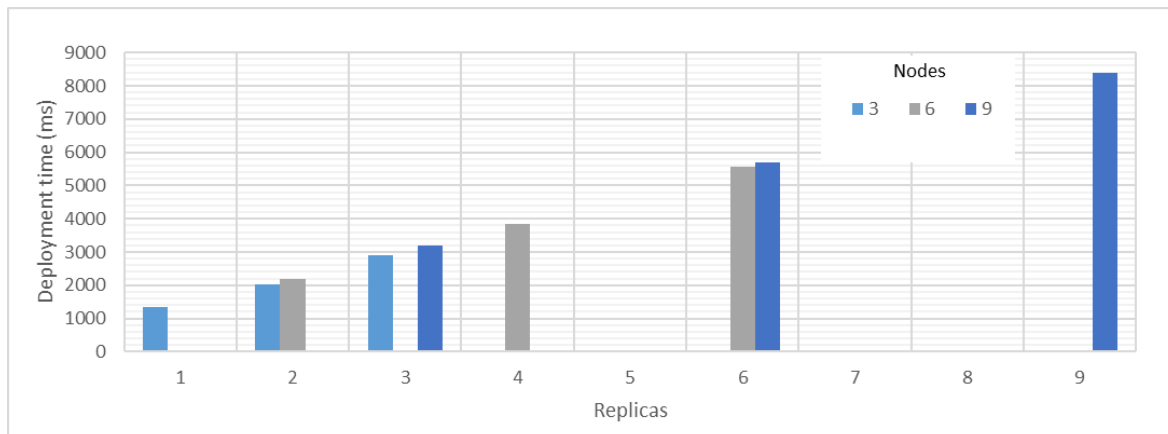
Taula 5-7 Prozesatze-nodoaren akats-proba egin ondoren lortutako batez besteko denborak (N=20).

| Nodoak | Erreplikak | Hedapen-denbora | Detekzio-denbora | Baiezkatze-denbora | Funtzionaltasunaren berreskuratze-denbora | Erredundantziaren berreskuratze-denbora |
|--------|------------|-----------------|------------------|--------------------|---|---|
| 3 | 1 | 1354.25 | 24197 | 534.3 | N/A | N/A |
| | 2 | 2030.55 | 24334 | 534.8 | 1125.9 | 1332.6 |
| | 3 | 2898.3 | 24438 | 534.65 | 3042.85 | N/A |
| 6 | 2 | 2195.15 | 24466 | 541.35 | 1214.2 | 1655.65 |
| | 4 | 3840.3 | 24949 | 538.9 | 3234.65 | 3415.6 |
| | 6 | 5552.75 | 24492 | 539.75 | 3292.9 | N/A |
| 9 | 3 | 3187 | 24433 | 535.8 | 3136.05 | 3350.8 |
| | 6 | 5699.7 | 24428 | 537.15 | 3256.75 | 3418.9 |
| | 9 | 8392.35 | 24464 | 535 | 3352 | N/A |

Emaitza horiek zehatz-mehatz aztertu aurretik, 5-7 taulan aurkeztutako emaitzen alderdi pare bat aipatzekoak dira:

- Aurreko proba-agertokian gertatu zen bezala, detekzio-denborak aztertutako gainerako neurketekin alderatuta magnitude-ordena desberdinean daude. Berriz ere, detekzioa gertaera eragiten duen agentearekin harremanetan jartzeko saiakera ematen den unean soilik jazotzen delako gertatzen da hori.
- *Funtzionaltasunaren berreskuratze-denbora* eta *Erredundantziaren berreskuratze-denbora* zutabeek ez dute baliorik {3 nodo, 1 erreplika} kasurako. Hau erreplika *running* egoeran bakarrik hedatzen denean aplikazioaren egoera mantendu ezinari egotz dakiok.
- Era berean, {3 nodo, 3 erreplika}, {6 nodo, 6 erreplika} eta {9 nodo, 9 erreplika} kasuek ez dute inolako baliorik *Erredundantziaren berreskuratze-denbora* zutabeen. Izan ere, ez dago prozesatze-nodorik eskuragarri erreplika berriak sortzeko (nodo guztietan agente bakoitzaren erreplika bat dago).

Lortutako emaitzak hobeto aztertzeko, 5-7 taulan agertzen diren denborak irudikatu dira. Adibidez, 5-12 irudian erakusten diren hedapen-denboren kasuan, nabarmenki egiaztatzen da erreplika-kopuruaren araberakoak direla. Aitzitik, nodo kopuruak hedapen-denbora modu esangurasuan iragartzen ez duela antzematen da (eragina izan dezake, baina erreplika kopuruak eragindakoa baino askoz ere ahulagoa).



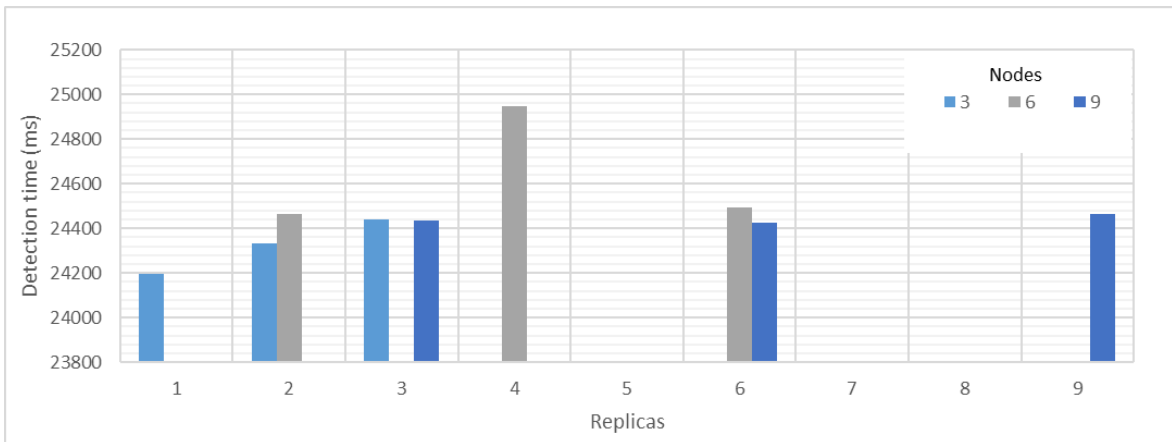
Irudia 5-12 Prozesatze-nodoaren akats-proba egin ondoren lortutako hedapen-denborak.

Detekzio- eta baieztatze-denborei dagokienez (5-13 eta 5-14 irudiak), emaitzek nodo- eta erreplika-kopuruaren menpe ez daudela erakusten dute. Ondorio horretatik harago, detekzio-denborak altuak dira (24 s ingurukoak), eta baieztatze-denborak 0.5 s inguruan daude.

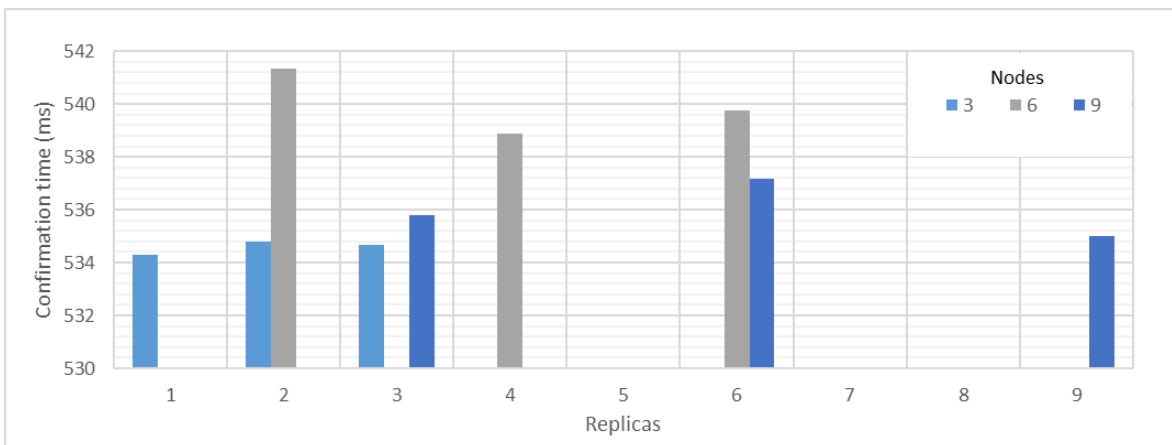
Funtzionaltasunaren berreskuratze-denborari dagokionez (ikus 5-15 irudia), emaitzek 2 erreplikatarako txikia dela erakusten dute. Hau espero izatekoa zen, kasu honetan sistemak ez baitu negoziatu behar. 3 erreplikatik aurrera, *funtzionaltasunaren berreskuratze-denboran* 100 ms inguruko igoera dagoela ikusten da 3 erreplikako gehikuntza bakoitzeko. Badirudi nodo kopuruak ez diola *funtzionaltasunaren berreskuratze-denborari* eragiten.

Azkenik, *erredundantziaren berreskuratze-denbora* (ikus Irudia 5-16 irudia) *funtzionaltasunaren berreskuratze-denborarekin* estuki lotuta dago, erredundantzia berreskuratzeke, lehenik funtzionaltasuna berreskuratu beharra baitago beti. Kasu honek erreplika berri bat sortzeko negoziazio bat eskatzen du, beraz, eskuragarri dauden nodo kopuruarekiko menpekotasuna espero zen (adibidez, {9 nodo, 3 erreplika} kasuan 6 nodo aske geldituko dira akatsa jazo ondoren; {6 nodo, 3 erreplika} kasuan, berriz, 3 nodo aske geldituko dira). Hala ere, lortutako emaitzetan ez da menpetasun hau nabaritzen. Izan ere, egindako probek erredundantzia 6 nodo askerekin 3 nodo askerekin baino azkarrago berreskuratu zela erakusten dute. Emaitza zehatz hau negoziazioan zein nodok parte har dezaketen egiaztatzeko behar

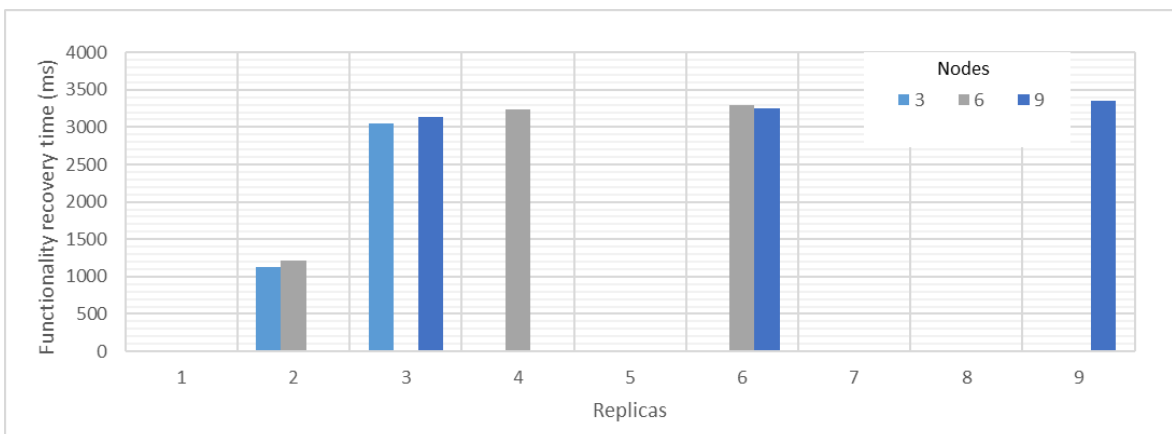
den denbora negoziazioak berak behar duen denbora baino luzeago izateak eragin dezake.



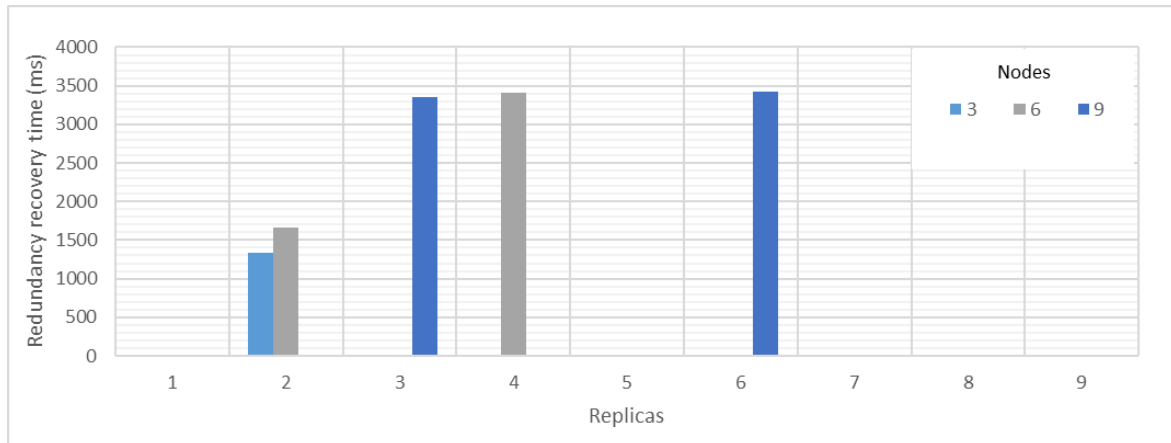
Irudia 5-13 Prozesatze-nodoaren akats-proba egin ondoren lortutako detekzio-denborak.



Irudia 5-14 Prozesatze-nodoaren akats-proba egin ondoren lortutako baieztatze-denborak.



Irudia 5-15 Prozesatze-nodoaren akats-proba egin ondoren lortutako Funtzionaltasunaren berreskuratze-denborak.



Irudia 5-16 Prozesatze-nodoaren akats-proba egin ondoren lortutako Erreduantziaren berreskuratze-denborak.

5.5 Laburpena eta ondorioak

Kapitulu honek *IARMS* I4.0 Plataformak industria-inguruneetan izan dezakeen aplikagarritasuna erakusten duen kontzeptu-proba bat aurkezten du.

Lehenik eta behin, kontzeptu-proba honetan erabilitako saiakuntza-plataformaren deskribapena egin da. Eskura dauden baliabideak eta I4.0 Plataforma saiakuntza-plataforman hedatzeko erabilitako tresnak eta teknologiak azpimarratu dira. Gainera, fabrikazio-zelula bat 4.0 Industrian nola integratu daitekeen aztertu da, 3.3.2 azpiatalean aurkeztutako metodologia praktikan jarritz.

Jarraian, *IARMS* I4.0. Plataforma ebaluatzeko proba-agertokiak eta hauei lotutako *KPI*ak definitu dira. Gai hau orokorretik partikularrera joratu da. CeDRI ikerketa-zentroko ikertzaileekin lankidetzan egindako lanaren zati gisa, agertoki eta *KPI* generikoak definitu dira Industria 4.0ren soluzioak malgutasunari dagokionez ebaluatzeko. Ondoren, saiakuntza-plataforma honetarako proba-agertoki pertsonalizatuak aurkeztu dira, eta emaitza onak lortu dira hauek proba-agertoki generikoekin bat datozela egiaztatzerakoan: proposatutako hiru agertokiak bat datoz proposamenarekin, nahiz eta proposatutako *KPI* guztiak kalkulatu ez diren.

Azkenik, hiru probak egin ondoren lortutako emaitzak aurkeztu dira. Lortutako emaitzen eztabaida 5.4 atalean azaltzen da. Iruzkina orokor gisa, gertaeren detektatzen denbora, oro har, oso luzea dela antzeman da. Hau baldintza normaletan (hots, *running* egoeran) lan egiten duten agenteen arteko elkarrekintzan oinarritutako gertaeren detekzio moduari egotz dakioke. Horregatik, alderdi hori hobetzeko, etorkizunean komenigarria dirudi *heartbeat* bezalako mekanismoak txertatzeko aukera aztertzea.

6 CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

As a closure to the PhD dissertation, this chapter summarizes the main conclusions obtained at the end of this work. To this end, the objectives set out in Section 1.2 are reviewed, and it is checked to what extent the questions posed in Section 1.1 have been answered.

The first objective was to carry out a comparative analysis of the reference architectures for Industry 4.0, in search of synergies or at least information to be able to decide which one to follow. In this regard, the conducted analysis of the state of the art shows a clear and unequivocal answer: *RAMI 4.0* has emerged as the undisputed reference architecture for Industry 4.0, followed by *IIRA*. It so happens that *Plattform Industrie 4.0* and *IIC*, the institutions behind *RAMI 4.0* and *IIRA*, have the closest collaboration, or at least that is what can be deduced from the number of reports and documents they have jointly published. Precisely, the support it receives from the *Plattform Industrie 4.0* and the interest it arouses in the scientific community mean that the volume of information on the subject is incomparable to that of other reference architectures. However, there is still a long way to go: there is a lot of material on the formalization of submodels and how to export and exchange them, but aspects related to the integration of assets or proactivity in communications present gaps that it is essential for researchers to fill with our work.

In this regard, the next objective was to design a methodology that would allow the integration of assets in Industry 4.0 regardless of their functional and communication characteristics. The analysis of the state of the art reveals a lack of works that bring together a generic character and a methodological proposal. This problem has been addressed by proposing an approach that divides the integration problem into simple and manageable fragments. In addition, the proposal has been accompanied by an integration methodology. In this way, the user is guided in the development of solutions based on the approach. The novelty of this proposal lies in not treating integration as just another aspect in the development of *AAS*, but giving it a specific weight, since it is one of the main brakes in the adoption of Industry 4.0 by companies.

In addition to asset integration, it has been detected that other aspects such as asset characterization or AAS development also lack methodological support for the development of solutions for Industry 4.0. These problems have also been addressed in a generic way, starting from the standards and concepts on which *RAMI 4.0* is based. Concerning asset characterization, the standards that define the *RAMI 4.0* factory hierarchy have been analyzed, and a modeling has been proposed to represent the dynamism of manufacturing processes in Industry 4.0. Regarding the development of AASs, its characteristics have been analyzed and it is proposed to carry it out in two stages: the skeleton pattern and the design pattern. This makes it possible to develop the functionalities required for interoperability and service management in a generic way (*skeleton pattern*), independently of the implementation of these services for a specific application (*design pattern*). As a result, Chapter 3 encompasses a methodological approach to Industry 4.0 based on *RAMI 4.0*, dealing with all these three issues (asset characterization and integration and AAS development) independently. This approach contributes to facilitate users' access to Industry 4.0 without the need to master in depth the concepts or technologies related to this paradigm.

This approach has been the basis for addressing the following objective: the design and development an I4.0 Platform that provides the necessary infrastructure to support the resilient management of manufacturing processes. The IARMS I4.0 Platform, presented in Chapter 4, proposes two blocks to provide the necessary infrastructure: on the one hand, *skeleton patterns* support the interoperability and internal management of the I4.0 components; on the other hand, an *I4.0 Platform core* enables management at the application level and ensures resilient manufacturing. This core implements the infrastructure services contained in *Plattform Industrie 4.0*. However, these only guarantee the operation of the system under normal operating conditions. For this reason, they have been complemented with additional services that allow the detection and resolution of incidents in the manufacturing process. It has been concluded that industrial agents aligned with the *FIPA* standard represent a valid alternative to meet this objective, thanks to the combination of *ACL* communication and the inherent capabilities of industrial agents (autonomous decision making, asset

integration, etc.). In addition, there are different agent development frameworks that offer basic *FIPA*-compliant functionalities, providing a foundation on which to build.

The last objective relates to the definition of test scenarios to evaluate the IARMS I4.0 Platform. In this sense, the work done has not been limited to a specific proposal, but tests and metrics have been defined that can be generically applicable to any I4.0 Platform, allowing benchmarking. In this part it is essential to remember that this work has been carried out in collaboration with CeDRI researchers. This proposal has served as a basis for the definition of specific scenarios applied to the demonstrator available in the research laboratory.

The contributions described in this PhD dissertation have been published in the form of 2 papers in Journal Citation Report index international Journals, and 7 communications in the most relevant international congresses in the field of research.

First, as a result of the methodological proposals to bring Industry 4.0 closer to companies and professionals, the following papers have been published:

1. López, A., Estévez, E., Casquero, O., Marcos, M., 2020. Using industrial standards for modeling flexible manufacturing systems, in: 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS). Presented at the 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS), pp. 41–46. <https://doi.org/10.1109/ICPS48405.2020.9274785>
2. López, A., Casquero, O., Estévez, E., Leitão, P., Marcos, M., 2021. Towards the generic integration of agent-based AASs and Physical Assets: a four-layered architecture approach, in: 2021 IEEE 19th International Conference on Industrial Informatics (INDIN). Presented at the 2021 IEEE 19th International Conference on Industrial Informatics (INDIN), pp. 1–6. <https://doi.org/10.1109/INDIN45523.2021.9557568>
3. López, A., Estévez, E., Casquero, O., Marcos, M., 2022. A Methodological Approach for integrating Physical Assets in Industry 4.0. IEEE Transactions on Industrial Informatics 1–9. <https://doi.org/10.1109/TII.2022.3230714> IF

JCR@2021: 11.648, 4/112 in Computer Science, Interdisciplinary Applications (Q1).

4. López, A., Casquero, O., Marcos, M., 2021b. Design patterns for the implementation of Industrial Agent-based AASs, in: 2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS). Presented at the 2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS), pp. 213–218. <https://doi.org/10.1109/ICPS49255.2021.9468129>

With respect to the design and development of the IARMS I4.0 Platform, the following works can be highlighted:

1. López, A., Estévez, E., Marcos, M., 2021c. Trazabilidad de la producción basada en agentes industriales, in: XLII Jornadas de automática libro de actas (Castellón, 1-3 de septiembre de 2021). Presented at the XLII Jornadas de Automatica, p. 7. <https://doi.org/10.17979/spudc.9788497498043.717>
2. López, A., Casquero, O., Estévez, E., Armentia, A., Orive, D., Marcos, M., 2023. An industrial agent-based customizable platform for I4.0 manufacturing systems. *Computers in Industry* 146, 103859. <https://doi.org/10.1016/j.compind.2023.103859> IF JCR@2021: 11.245, 5/112 in Computer Science, Interdisciplinary Applications (Q1).

Finally, the following works are related to the proof of concept, addressing either the definition of test scenarios or the development of the demonstrator used:

1. Lopez, A., Sakurada, L., Leitao, P., Casquero, O., Estevez, E., De la Prieta, F., Marcos, M., 2022. Technology-Independent Demonstrator for Testing Industry 4.0 Solutions, in: 2022 IEEE 20th International Conference on Industrial Informatics (INDIN). Presented at the 2022 IEEE 20th International Conference on Industrial Informatics (INDIN), pp. 21–26. <https://doi.org/10.1109/INDIN51773.2022.9976144>
2. Orive, D., López, A., Estévez, E., Orive, A., Marcos, M., 2021. Desarrollo de gemelos digitales para la simulación e integración de activos de fabricación en la industria 4.0, in: XLII Jornadas de automática libro de actas (Castellón, 1-3 de

septiembre de 2021). Presented at the XLII Jornadas de Automática, p. 8.
<https://doi.org/10.17979/spudc.9788497498043.709>

3. Aragon, A., López García, A., Sarachaga, I., Casquero, O., Orive, D., Armentia, A., Pérez, F., 2022. Componente I4.0 de una estación de ensamblado, in: XLIII Jornadas de Automática: libro de actas: 7, 8 y 9 de septiembre de 2022, Logroño (La Rioja). Servizo de Publicacións da UDC, pp. 877–884.
<https://doi.org/10.17979/spudc.9788497498418.0877>

6.2 Future works

During the development of this work, different lines of interest have been detected for the development of future research work. These lines respond to limitations found in the development of this work to aspects treated superficially but require an in-depth analysis.

Nowadays, there are several authors who defend that people have to be a fundamental part of Industry 4.0. For this reason, one line of action for the future could be the extension of the modeling proposal presented in this work, incorporating the rest of the resources contemplated in the IEC 62264 and IEC 61512 standards, with special emphasis on the "personnel" resource.

Another interesting line of work would be related to the use of different agent development frameworks. Although JADE remains in force (the current version, V4.6.0, was released on 07/12/2022), it is true that alternative platforms, such as Python Agent DEvelopment framework (PADE¹²) or Smart Python Agent Development Environment (SPADE¹³) have recently gained popularity. The reason is that they are based on Python. This gives access to libraries with Machine Learning functionalities that could be used to enhance the agents' capabilities.

¹² <https://pade.readthedocs.io/en/latest/#>

¹³ <https://spade-mas.readthedocs.io/en/latest/index.html#>

Finally, it is considered essential to continue with the effort to define generic scenarios and metrics for the comparative analysis of solutions for Industry 4.0. In this sense, there is room to increase the number of scenarios to be evaluated, both in terms of flexibility and other aspects (safety, energy consumption, etc.).

REFERENCES

Adolphs, P., Epple, U., 2015. Status Report: Reference Architecture Model Industrie 4.0 (RAMI4.0).

Alexopoulos, K., Anagiannis, I., Nikolakis, N., Chryssolouris, G., 2022. A quantitative approach to resilience in manufacturing systems. *International Journal of Production Research* 60, 7178–7193. <https://doi.org/10.1080/00207543.2021.2018519>

Alignment Report for Reference Architectural Model for Industrie 4.0/ Intelligent Manufacturing System Architecture, 2018.

Arm, J., Benesl, T., Marcon, P., Bradac, Z., Schröder, T., Belyaev, A., Werner, T., Braun, V., Kamensky, P., Zezulka, F., Diedrich, C., Dohnal, P., 2021. Automated Design and Integration of Asset Administration Shells in Components of Industry 4.0. *Sensors* 21, 2004. <https://doi.org/10.3390/s21062004>

Ashtari Talkhestani, B., Jung, T., Lindemann, B., Sahlab, N., Jazdi, N., Schloegl, W., Weyrich, M., 2019. An architecture of an Intelligent Digital Twin in a Cyber-Physical Production System. *at - Automatisierungstechnik* 67, 762–782. <https://doi.org/10.1515/auto-2019-0039>

Bader, S.R., Maleshkova, M., Lohmann, S., 2019. Structuring Reference Architectures for the Industrial Internet of Things. *Future Internet* 23.

Bakliwal, K., Dhada, M.H., Palau, A.S., Parlikad, A.K., Lad, B.K., 2018. A Multi Agent System architecture to implement Collaborative Learning for social industrial assets. *IFAC-PapersOnLine* 51, 1237–1242. <https://doi.org/10.1016/j.ifacol.2018.08.421>

Barbosa, J., Leitão, P., Adam, E., Trentesaux, D., 2015. Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution. *Computers in Industry* 66, 99–111. <https://doi.org/10.1016/j.compind.2014.10.011>

Baumgärtel, H., Verbeet, R., 2020. Service and Agent based System Architectures for Industrie 4.0 Systems, in: *NOMS 2020 - 2020 IEEE/IFIP Network Operations and*

Management Symposium. Presented at the NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium, pp. 1–6.
<https://doi.org/10.1109/NOMS47738.2020.9110406>

Bennulf, M., Danielsson, F., Svensson, B., Lennartson, B., 2020. Goal-oriented process plans in a multi-agent system for Plug & Produce. IEEE Trans. Ind. Inf. 1–1.
<https://doi.org/10.1109/TII.2020.2994032>

Boss, B., Malakuti, S., Lin, S.-W., Usländer, T., Clauer, E., Hoffmeister, M., Stojmenovic, L., 2020. Digital Twin and Asset Administration Shell Concepts and Application in the Industrial Internet and Industrie 4.0.

Braune, A., Diedrich, C., Grüner, S., Hüttemann, G., Klein, M., Legat, C., Lieske, M., Löwen, U., Thron, M., Usländer, T., 2019. Usage View of Asset Administration Shell.

Braune, A., Hüttemann, G., Klein, M., Löwen, U., Thron, M., 2018. Discussion Paper – Usage Viewpoint of Application Scenario Value-Based Service.

Cañas, H., Mula, J., Campuzano-Bolarín, F., Poler, R., 2022. A conceptual framework for smart production planning and control in Industry 4.0. Computers & Industrial Engineering 173, 108659. <https://doi.org/10.1016/j.cie.2022.108659>

Casquero, O., Armentia, A., Estevez, E., López, A., Marcos, M., 2020. Customization of agent-based manufacturing applications based on domain modelling, in: 21st IFAC World Congress.

Cavaliere, S., Salafia, M.G., 2020. Asset Administration Shell for PLC Representation Based on IEC 61131–3. IEEE Access 8, 142606–142621.
<https://doi.org/10.1109/ACCESS.2020.3013890>

China Manufacturing 2025, 2017.

Colombo, A.W., Karnouskos, S., Kaynak, O., Shi, Y., Yin, S., 2017. Industrial Cyberphysical Systems: A Backbone of the Fourth Industrial Revolution. *IEEE Industrial Electronics Magazine* 11, 6–16. <https://doi.org/10.1109/MIE.2017.2648857>

Contreras, J.D., Garcia, J.I., Diaz, J.D., 2017. Developing of Industry 4.0 Applications. *Int. J. Onl. Eng.* 13, 30. <https://doi.org/10.3991/ijoe.v13i10.7331>

Cruz Salazar, L.A., Ryashentseva, D., Lüder, A., Vogel-Heuser, B., 2019. Cyber-physical production systems architecture based on multi-agent's design pattern—comparison of selected approaches mapping four agent patterns. *Int J Adv Manuf Technol.* <https://doi.org/10.1007/s00170-019-03800-4>

Cruz Salazar, L.A., Vogel-Heuser, B., 2022. A CPPS-architecture and workflow for bringing agent-based technologies as a form of artificial intelligence into practice. *at - Automatisierungstechnik* 70, 580–598. <https://doi.org/10.1515/auto-2022-0008>

Dai, W., Dubinin, V.N., Christensen, J.H., Vyatkin, V., Guan, X., 2017. Toward Self-Manageable and Adaptive Industrial Cyber-Physical Systems With Knowledge-Driven Autonomic Service Management. *IEEE Transactions on Industrial Informatics* 13, 725–736. <https://doi.org/10.1109/TII.2016.2595401>

de las Morenas, J., Garcia-Higuera, A., Garcia-Ansola, P., 2017. Shop Floor Control: A Physical Agents Approach for PLC-Controlled Systems. *IEEE Trans. Ind. Inf.* 13, 2417–2427. <https://doi.org/10.1109/TII.2017.2720696>

Digital Transformation Monitor Germany: Industrie 4.0 [WWW Document], 2017. URL https://ati.ec.europa.eu/sites/default/files/2020-06/DTM_Industrie%204.0_DE.pdf (accessed 11.7.22).

DIN SPEC 91345, 2016.

do Livramento Gonçalves, G., Leal Filho, W., da Silva Neiva, S., Borchardt Deggau, A., de Oliveira Veras, M., Ceci, F., Andrade de Lima, M., Salgueirinho Osório de Andrade

Guerra, J.B., 2021. The Impacts of the Fourth Industrial Revolution on Smart and Sustainable Cities. *Sustainability* 13, 7165. <https://doi.org/10.3390/su13137165>

Dorofeev, K., Cheng, C.-H., Guedes, M., Ferreira, P., Profanter, S., Zoitl, A., 2017. Device adapter concept towards enabling plug&produce production environments, in: 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). Presented at the 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Limassol, pp. 1–8. <https://doi.org/10.1109/ETFA.2017.8247570>

Evans, P., Annunziata, M., 2012. Industrial Internet: Pushing the boundaries of minds and machines.

Faheem, M., Shah, S.B.H., Butt, R.A., Raza, B., Anwar, M., Ashraf, M.W., Ngadi, Md.A., Gungor, V.C., 2018. Smart grid communication and information technologies in the perspective of Industry 4.0: Opportunities and challenges. *Computer Science Review* 30, 1–30. <https://doi.org/10.1016/j.cosrev.2018.08.001>

Fay, A., Gehlhoff, F., Seitz, M., Vogel-Heuser, B., Baumgaertel, H., Diedrich, C., Lüder, A., Schöler, T., Sutschet, G., Verbeet, R., 2019. Agents for the realisation of Industrie 4.0 - VDI Status Report. <https://doi.org/10.13140/RG.2.2.23998.84805>

Fraile, Sanchis, Poler, Ortiz, 2019. Reference Models for Digital Manufacturing Platforms. *Applied Sciences* 9, 4433. <https://doi.org/10.3390/app9204433>

Fuchs, J., Schmidt, J., Franke, J., Rehman, K., Sauer, M., Karnouskos, S., 2019. I4.0-compliant integration of assets utilizing the Asset Administration Shell, in: 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). Presented at the 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1243–1247. <https://doi.org/10.1109/ETFA.2019.8869255>

Gangoiti, U., López, A., Armentia, A., Estévez, E., Casquero, O., Marcos, M., 2022. A Customizable Architecture for Application-Centric Management of Context-Aware Applications. IEEE Access 10, 1603–1625. <https://doi.org/10.1109/ACCESS.2021.3138586>

Han, S., 2020. A review of smart manufacturing reference models based on the skeleton meta-model. Journal of Computational Design and Engineering 7, 323–336. <https://doi.org/10.1093/jcde/qwaa027>

IEC 61512-1, 1997.

IEC 62264-1, 2013.

IEC 62264-2, 2013.

IEC 62890, 2020.

IEC PAS 63088, 2017.

IEEE Recommended Practice for Industrial Agents: Integration of Software Agents and Low-Level Automation Functions, 2021. . IEEE Std 2660.1-2020 1–43. <https://doi.org/10.1109/IEEESTD.2021.9340089>

IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries, 1991. . IEEE Std 610 1–217. <https://doi.org/10.1109/IEEESTD.1991.106963>

Industrial Value Chain Reference Architecture (IVRA), 2016.

Industrial Valuechain Initiative [WWW Document], n.d. URL <https://iv-i.org/wp/en/> (accessed 8.17.20).

Industry IoT Consortium [WWW Document], n.d. . Industry IoT Consortium. URL <https://www.iiconsortium.org/> (accessed 11.8.22).

Kagermann, H., Lukas, W.-D., Wahlster, W., 2011. Industrie 4.0: Mit dem Internet der Dinge auf dem Weg zur 4. industriellen Revolution.

Karnouskos, S., Leitão, P., 2017. Key Contributing Factors to the Acceptance of Agents in Industrial Environments. *IEEE Transactions on Industrial Informatics* 13, 696–703. <https://doi.org/10.1109/TII.2016.2607148>

Karnouskos, S., Leitao, P., Ribeiro, L., Colombo, A.W., 2020. Industrial Agents as a Key Enabler for Realizing Industrial Cyber-Physical Systems: Multiagent Systems Entering Industry 4.0. *IEEE Industrial Electronics Magazine* 14, 18–32. <https://doi.org/10.1109/MIE.2019.2962225>

Kovalenko, I., Balta, E.C., Tilbury, D.M., Barton, K., 2022. Cooperative Product Agents to Improve Manufacturing System Flexibility: A Model-Based Decision Framework. *IEEE Transactions on Automation Science and Engineering* 1–18. <https://doi.org/10.1109/TASE.2022.3156384>

Leitão, P., Barbosa, J., 2016. Building a Robotic Cyber-Physical Production Component, in: Borangiu, T., Trentesaux, D., Thomas, A., McFarlane, D. (Eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing, Studies in Computational Intelligence*. Springer International Publishing, Cham, pp. 295–305. https://doi.org/10.1007/978-3-319-30337-6_27

Leitão, P., Barbosa, J., Pereira, A., Barata, J., Colombo, A.W., 2016. Specification of the PERFoRM architecture for the seamless production system reconfiguration, in: *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*. Presented at the *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pp. 5729–5734. <https://doi.org/10.1109/IECON.2016.7793007>

Leitão, P., Karnouskos, S., Ribeiro, L., Moutis, P., Barbosa, J., Strasser, T.I., 2018. Integration Patterns for Interfacing Software Agents with Industrial Automation Systems, in: IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society. Presented at the IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society, pp. 2908–2913. <https://doi.org/10.1109/IECON.2018.8591641>

Leitão, P., Queiroz, J., Sakurada, L., 2022. Collective Intelligence in Self-Organized Industrial Cyber-Physical Systems. *Electronics* 11, 3213. <https://doi.org/10.3390/electronics11193213>

Leitao, P., Restivo, F., 2006. ADACOR: A holonic architecture for agile and adaptive manufacturing control. *Computers in Industry*. <https://doi.org/10.1016/j.compind.2005.05.005>

Li, Q., Tang, Q., Chan, I., Wei, H., Pu, Y., Jiang, H., Li, J., Zhou, J., 2018. Smart manufacturing standardization: Architectures, reference models and standards framework. *Computers in Industry* 101, 91–106. <https://doi.org/10.1016/j.compind.2018.06.005>

Lin, S.-W., Miller, B., Durand, J., Bleakley, G., Chigani, A., Martin, R., Murphy, B., Crawford, M., 2019. The Industrial Internet of Things Volume G1: Reference Architecture.

Lin, S.-W., Murphy, B., Clauer, E., Loewen, U., Neubert, R., Bachmann, G., Pai, M., Hankel, M., 2017. Architecture Alignment and Interoperability.

Lopez, A., Sakurada, L., Leitao, P., Casquero, O., Estevez, E., De la Prieta, F., Marcos, M., 2022. Technology-Independent Demonstrator for Testing Industry 4.0 Solutions, in: 2022 IEEE 20th International Conference on Industrial Informatics (INDIN). Presented at the 2022 IEEE 20th International Conference on Industrial Informatics (INDIN), pp. 21–26. <https://doi.org/10.1109/INDIN51773.2022.9976144>

Marcon, P., Diedrich, C., Zezulka, F., Schröder, T., Belyaev, A., Arm, J., Benesl, T., Bradac, Z., Vesely, I., 2018. The Asset Administration Shell of Operator in the Platform of Industry 4.0, in: 2018 18th International Conference on Mechatronics - Mechatronika

(ME). Presented at the 2018 18th International Conference on Mechatronics - Mechatronika (ME), pp. 1–5.

Martin, J., Casquero, O., Fortes, B., Marcos, M., 2019. A Generic Multi-Layer Architecture Based on ROS-JADE Integration for Autonomous Transport Vehicles. *Sensors* 19, 69. <https://doi.org/10.3390/s19010069>

Miny, T., Stephan, G., Usländer, T., Vialkowitsch, J., 2021. Functional View of the Asset Administration Shell in an Industrie 4.0 System Environment.

Moghaddam, M., Cadavid, M.N., Kenley, C.R., Deshmukh, A.V., 2018. Reference architectures for smart manufacturing: A critical review. *Journal of Manufacturing Systems* 49, 215–225. <https://doi.org/10.1016/j.jmsy.2018.10.006>

Monostori, L., 2018. Cyber-Physical Systems, in: *The International Academy for Production*, Chatti, S., Tolio, T. (Eds.), *CIRP Encyclopedia of Production Engineering*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–8. https://doi.org/10.1007/978-3-642-35950-7_16790-1

Munkelt, T., Krockert, M., 2018. AGENT-BASED SELF-ORGANIZATION VERSUS CENTRAL PRODUCTION PLANNING, in: *2018 Winter Simulation Conference (WSC)*. Presented at the 2018 Winter Simulation Conference (WSC), IEEE, Gothenburg, Sweden, pp. 3241–3251. <https://doi.org/10.1109/WSC.2018.8632305>

Nakagawa, E.Y., Antonino, P.O., Schnicke, F., Capilla, R., Kuhn, T., Liggesmeyer, P., 2021. Industry 4.0 reference architectures: State of the art and future trends. *Computers & Industrial Engineering* 156, 107241. <https://doi.org/10.1016/j.cie.2021.107241>

Neal, A.D., Sharpe, R.G., van Lopik, K., Tribe, J., Goodall, P., Lugo, H., Segura-Velandia, D., Conway, P., Jackson, L.M., Jackson, T.W., West, A.A., 2021. The potential of industry 4.0 Cyber Physical System to improve quality assurance: An automotive case study for wash monitoring of returnable transit items. *CIRP Journal of Manufacturing Science and Technology* 32, 461–475. <https://doi.org/10.1016/j.cirpj.2020.07.002>

Nishioka, Y., 2017. Industrial Value Chain Reference Architecture.

Orio, G. di, Maló, P., Barata, J., 2019. NOVAAS: A Reference Implementation of Industrie4.0 Asset Administration Shell with best-of-breed practices from IT engineering, in: IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society. Presented at the IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society, pp. 5505–5512. <https://doi.org/10.1109/IECON.2019.8927081>

Peres, R.S., Dionisio Rocha, A., Leitao, P., Barata, J., 2018. IDARTS – Towards intelligent data analysis and real-time supervision for industry 4.0. *Computers in Industry* 101, 138–146. <https://doi.org/10.1016/j.compind.2018.07.004>

Pethig, F., Niggemann, O., Walter, A., 2017. Towards Industrie 4.0 compliant configuration of condition monitoring services, in: 2017 IEEE 15th International Conference on Industrial Informatics (INDIN). Presented at the 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), pp. 271–276. <https://doi.org/10.1109/INDIN.2017.8104783>

Pisching, M.A., Pessoa, M.A.O., Junqueira, F., dos Santos Filho, D.J., Miyagi, P.E., 2018. An architecture based on RAMI 4.0 to discover equipment to process operations required by products. *Computers & Industrial Engineering* 125, 574–591. <https://doi.org/10.1016/j.cie.2017.12.029>

Pivoto, D.G.S., de Almeida, L.F.F., da Rosa Righi, R., Rodrigues, J.J.P.C., Lugli, A.B., Alberti, A.M., 2021. Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: A literature review. *Journal of Manufacturing Systems* 58, 176–192. <https://doi.org/10.1016/j.jmsy.2020.11.017>

Plattform Industrie 4.0 [WWW Document], n.d. URL <https://www.plattform-i40.de/IP/Navigation/EN/Home/home.html> (accessed 11.2.22).

Robot Revolution Initiative [WWW Document], n.d. URL <https://www.jmfrri.gr.jp/english/> (accessed 12.7.22).

Sakurada, L., Leitão, P., 2020. Multi-Agent Systems to Implement Industry 4.0 Components, in: 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS). Presented at the 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS), pp. 21–26. <https://doi.org/10.1109/ICPS48405.2020.9274745>

Sakurada, L., Leitao, P., la Prieta, F.D., 2022. Agent-Based Asset Administration Shell Approach for Digitizing Industrial Assets. *IFAC-PapersOnLine* 55, 193–198. <https://doi.org/10.1016/j.ifacol.2022.04.192>

Schou, C., Madsen, O., 2017. A plug and produce framework for industrial collaborative robots. *International Journal of Advanced Robotic Systems* 14, 1729881417717472. <https://doi.org/10.1177/1729881417717472>

Schuh, G., Anderl, R., Dumitrescu, R., Krüger, A., 2020a. Industrie 4.0 Maturity Index.

Schuh, G., Anderl, R., Dumitrescu, R., Krüger, A., 2020b. Using the Industrie 4.0 Maturity Index in Industry.

Schwab, K., 2016. *The Fourth Industrial Revolution*.

Schweichhart, K., 2016. Reference Architectural Model Industrie 4.0 (RAMI 4.0).

Seif, A., Toro, C., Akhtar, H., 2019. Implementing Industry 4.0 Asset Administrative Shells in Mini Factories. *Procedia Computer Science* 159, 495–504. <https://doi.org/10.1016/j.procs.2019.09.204>

Soares, N., Monteiro, P., Duarte, F.J., Machado, R.J., 2021. Extending the scope of reference models for smart factories. *Procedia Computer Science* 180, 102–111. <https://doi.org/10.1016/j.procs.2021.01.134>

Standard Status Specifications [WWW Document], n.d. URL <http://www.fipa.org/repository/standardspecs.html> (accessed 9.2.22).

Systems and software engineering — Architecture description, 2011.

Tang, D., Zheng, K., Zhang, H., Zhang, Z., Sang, Z., Zhang, T., Espinosa-Oviedo, J.-A., Vargas-Solar, G., 2018. Using autonomous intelligence to build a smart shop floor. *Int J Adv Manuf Technol* 94, 1597–1606. <https://doi.org/10.1007/s00170-017-0459-y>

Tang, H., Li, D., Wang, S., Dong, Z., 2018. CASOA: An Architecture for Agent-Based Manufacturing System in the Context of Industry 4.0. *IEEE Access* 6, 12746–12754. <https://doi.org/10.1109/ACCESS.2017.2758160>

Tantik, E., Anderl, R., 2017. Integrated Data Model and Structure for the Asset Administration Shell in Industrie 4.0. *Procedia CIRP* 60, 86–91. <https://doi.org/10.1016/j.procir.2017.01.048>

Trunzer, E., Calà, A., Leitão, P., Gepp, M., Kinghorst, J., Lüder, A., Schauerte, H., Reifferscheid, M., Vogel-Heuser, B., 2019. System architectures for Industrie 4.0 applications: Derivation of a generic architecture proposal. *Prod. Eng. Res. Dev.* 13, 247–257. <https://doi.org/10.1007/s11740-019-00902-6>

Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P., 1998. Reference architecture for holonic manufacturing systems: PROSA. *Computers in Industry* 37, 255–274. [https://doi.org/10.1016/S0166-3615\(98\)00102-X](https://doi.org/10.1016/S0166-3615(98)00102-X)

VDI/VDE. 2021. 2653 Sheet 4: Multi-agent systems in industrial automation – Selected patterns for field level control and energy systems, 2022.

Vogel-Heuser, B., Seitz, M., Cruz Salazar, L.A., Gehlhoff, F., Dogan, A., Fay, A., 2020. Multi-agent systems to enable Industry 4.0. *at - Automatisierungstechnik* 68, 445–458. <https://doi.org/10.1515/auto-2020-0004>

Wei, S., Hu, J., Cheng, Y., Ma, Y., Yu, Y., 2017. The essential elements of intelligent Manufacturing System Architecture, in: 2017 13th IEEE Conference on Automation Science and Engineering (CASE). Presented at the 2017 13th IEEE Conference on Automation Science and Engineering (CASE), pp. 1006–1011. <https://doi.org/10.1109/COASE.2017.8256234>

Wenger, M., Zoitl, A., Müller, T., 2018. Connecting PLCs With Their Asset Administration Shell For Automatic Device Configuration, in: 2018 IEEE 16th International Conference on Industrial Informatics (INDIN). Presented at the 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), pp. 74–79. <https://doi.org/10.1109/INDIN.2018.8472022>

What is Industrie 4.0? [WWW Document], n.d. URL <https://www.plattform-i40.de/PI40/Navigation/EN/Industrie40/WhatIsIndustrie40/what-is-industrie40.html> (accessed 8.16.20).

World Economic Forum, 2016. What is the fourth industrial revolution? [WWW Document]. World Economic Forum. URL <https://www.weforum.org/agenda/2016/01/what-is-the-fourth-industrial-revolution/> (accessed 8.15.20).

World Economic Forum, McKinsey & Company, 2018. The Next Economic Growth Engine.

Ye, X., Hong, S.H., 2019. Toward Industry 4.0 Components: Insights Into and Implementation of Asset Administration Shells. *IEEE Ind. Electron. Mag.* 13, 13–25. <https://doi.org/10.1109/MIE.2019.2893397>

Ye, X., Hong, S.H., Song, W.S., Kim, Y.C., Zhang, X., 2021. An Industry 4.0 Asset Administration Shell-Enabled Digital Solution for Robot-Based Manufacturing Systems. *IEEE Access* 9, 154448–154459. <https://doi.org/10.1109/ACCESS.2021.3128580>

Ye, X., Jiang, J., Lee, C., Kim, N., Yu, M., Hong, S.H., 2020. Toward the Plug-and-Produce Capability for Industry 4.0: An Asset Administration Shell Approach. *EEE Ind. Electron. Mag.* 14, 146–157. <https://doi.org/10.1109/MIE.2020.3010492>

Zanero, S., 2017. Cyber-Physical Systems. *Computer* 50, 14–16. <https://doi.org/10.1109/MC.2017.105>

Zeid, A., Sundaram, S., Moghaddam, M., Kamarthi, S., Marion, T., 2019. Interoperability in Smart Manufacturing: Research Challenges. *Machines* 7, 21. <https://doi.org/10.3390/machines7020021>

Zimmermann, P., Axmann, E., Brandenbourger, B., Dorofeev, K., Mankowski, A., Zanini, P., 2019. Skill-based Engineering and Control on Field-Device-Level with OPC UA, in: 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). Presented at the 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1101–1108. <https://doi.org/10.1109/ETFA.2019.8869473>

GLOSSARY

GLOSSARY

| | |
|-----------------|---|
| ACL | <i>Agent Communication Language</i> |
| AAS | <i>Asset Administration Shell</i> |
| API | <i>Application Programming Interface</i> |
| BA | <i>Batch Agent</i> |
| CeDRI | <i>Research Centre in Digitalization and Intelligent Robotics</i> |
| CPS | <i>Cyber-Physical Systems</i> |
| DT | <i>Digital Twin</i> |
| D&DA | <i>Diagnosis and Decision Agent</i> |
| FIPA | <i>Foundation for Intelligent Physical Agents</i> |
| GCIS | <i>Grupo de Control e Integración de Sistemas</i> |
| GFB | <i>General Function Blocks</i> |
| IARMS | <i>Industrial Agents for Resilient Manufacturing Systems</i> |
| ICT | <i>Information and Communications Technology</i> |
| IIC | <i>Industry IoT Consortium</i> |
| IIRA | <i>Industrial Internet Reference Architecture</i> |
| IoT | <i>Internet of Things</i> |

| | |
|-----------------|--|
| IMSA | <i>Intelligent Manufacturing System Architecture</i> |
| IVI | <i>Industrial Value Chain Initiative</i> |
| IVRA | <i>Industrial Value Chain Reference Architecture</i> |
| JADE | <i>Java Agent DEvelopment Framework</i> |
| KPI | <i>Key Performance Indicator</i> |
| MA | <i>Machine Agent</i> |
| MDE | <i>Model-Driven Engineering</i> |
| MPA | <i>Manufacturing Plan Agent</i> |
| OA | <i>Order Agent</i> |
| OPC UA | <i>OPC Unified Architecture</i> |
| PLA | <i>Planner Agent</i> |
| PLC | <i>Programmable Logic Controller</i> |
| PNA | <i>Processing Node Agent</i> |
| PnP | <i>Plug and Produce</i> |
| QoSA | <i>Quality of Service Agent</i> |
| RAMI 4.0 | <i>Reference Architectural Model for Industrie 4.0</i> |
| ROS | <i>Robot Operating Systems</i> |

SMU *Smart Manufacturing Unit*

SR *System Repository*

SRA *System Repository Agent*

TA *Transport Agent*