André Dionísio Bettencourt da Silva Parreira Rocha

Mestre em Engenharia Eletrotécnica e Computadores

Increase the adoption of Agent-based Cyber-Physical Production Systems through the Design of Minimally Invasive Solutions

Dissertação para obtenção do Grau de Doutor em
Engenharia Electrotécnica e Computadores

Orientador: José António Barata de Oliveira,
Professor, Faculdade de Ciências e Tecnologia,
Universidade Nova de Lisboa

Co-orientador: Luis Domingos Ferreira Ribeiro,
Professor, Linköpings Universitet

Júri:

Presidente: Prof. Doutor Luis Manuel Camarinha de Matos

Arguentes: Prof. Doutor Paulo Jorge Pinto Leitão
Prof. Doutor Niels Lohse

Vogais: Prof. Doutor Luis Manuel Camarinha de Matos
Prof. Doutor Ricardo Jardim Gonçalves
Prof. Doutor José António Barata de Oliveira
Prof. Doutor Mauro Onori

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Julho de 2018

Increase the Adoption of Agent-based Cyber-Physical Production Systems through the Design of Minimally Invasive Solutions

For my wife Mafalda,

as an acknowledgement of the fantastic contribution you are

Acknowledgements

In this section, I would like to thank all those who contributed to the accomplishment of this dissertation.

I would firstly like to thank my supervisor, Professor José Barata, for the opportunity of working with him closely enough, which enabled to have professional experiences that I will never forget. Those experiences allow to enrich me not only from an academic point of view but also personally.

To my co-supervisor, Professor Luís Ribeiro, a special thanks. For all the availability, support and dedication along the entire thesis. Throughout this process, and during the preparation and writing of the document, his support was tireless.

A word of appreciation to the entire Thesis Accompanying Committee. Professor Paulo Leitão (IPB) and Professor Ricardo Gonçalves (FCT-UNL) were always available to contribute and help throughout the process with relevant and precious inputs for the completion of this project.

I want to thank Tiago Santos, Ricardo Peres, Pedro Monteiro and Mafalda Rocha. All of you have played a fundamental role in the development of this work. All of you have contributed exceptionally for this work to achieve the results. Without your contribution, this work could not be the same.

I would like to thank all my colleagues who worked with me at UNINOVA during these years, and to all the partners that worked with us in all the European Projects. All of them have somehow contributed, with their expertise, advises and numerus conversation on these topics, to the success of this work.

I do not want to forget all the friends I created during my journey. I want to thank all of them for the contact that was maintained even after some of them followed different directions. Especially to Fábio Oliveira, José Gonçalves, Catarina Lucena and Marcelo Rodrigues, thank you very much.

Now, a special thank you to all those who, even out of college, have contributed to making this journey end successfully. Starting with my *mano,* Gonçalo Rocha, for all the moments we spent together. Thank you Zeca!

Many thanks to Anabela, Nuno and Eduardo for their availability and the moments we spent together during these years.

My thanks to all those who, far away, in the land where I was born and that I love, continue to support me in all the stages of my life. To all of them my thanks for being always available no matter what happens. Manuela Rocha, Dionisio Rocha, Inês Silveira and Luís Silveira, thank you very much!

I would now like to thank the person who day after day contributes infinitely so that I can choose and create more for our life. With her by my side, I know I will go further, and this work is an example of this. However much I may write here, it will always be insufficient to express my gratitude and admiration for all these years we are together. Thank you so much Mafalda!

Resumo

Durante os últimos anos muitas foram as abordagens propostas para oferecer às empresas a capacidade de terem sistemas dinâmicos e flexíveis de produção. Uma das abordagens mais propostas é a utilização de sistemas distribuídos e baseados em sistemas multi-agente para a criação de sistemas ciber-físicos de produção. Embora estes sistemas tenham a capacidade de lidar com os desafios verificados pelas empresas, estes não têm sido aceites e utilizados em casos reais. Desta forma, o trabalho proposto tem como principal objetivo perceber os desafios que são verificados aquando da adoção destas soluções e desenvolver uma estratégia para aumentar a adoção dos mesmos.

Assim, é proposto o desenho e desenvolvimento de sistemas ciber-físicos de produção baseados em agentes que requeiram alterações mínimas nos sistemas de produção existentes, diminuindo o impacto e as mudanças requeridos para a adoção dos mesmos. Primeiramente é apresentada uma definição de sistema ciber-físico de produção baseado em agentes e é feito o levantamento de requisitos funcionais que devem ser respeitados. Destes requisitos funcionais derivou uma lista de princípios de desenho que devem ser cumpridos de forma a desenhar e desenvolver um sistema com estas caraterísticas.

Posteriormente e de forma a avaliar soluções que tenham como objetivo serem minimamente invasivas, é proposto um modelo de avaliação baseado num sistema de inferência difuso, que avalia cada uma das soluções segundo cada um dos princípios de desenho e globalmente.

Desta forma, o trabalho proposto apresenta os requisitos funcionais, princípios de desenho e modelo de avaliação de sistemas ciber-físicos de produção minimamente invasivos, para que os mesmos sejam desenhados e propostos de forma a aumentar a adoção de sistemas deste género.

Palavras-chave: Sistema Ciber-Físico de Produção, Sistema Multi-Agente, Sistema de Manufatura Inteligente, Minimamente Invasivo, Modelo de Avaliação.

Abstract

During the last few years, many approaches were proposed to offer companies the ability to have dynamic and flexible production systems. One of the conventional approaches to solving this problem is the implementation of cyber-physical production systems using multi-agent distributed systems. Although these systems can deal with several challenges faced by companies in this area, they have not been accepted and used in real cases. In this way, the primary objective of the proposed work is to understand the challenges usually found in the adoption of these solutions and to develop a strategy to increase their acceptance and implementation.

Thus, the document focuses on the design and development of cyber-physical production systems based on agent approaches, requiring minimal changes in the existing production systems. This approach aims of reducing the impact and the alterations needed to adopt those new cyber-physical production systems. Clarifying the subject, the author presents a definition of a minimal invasive agent-based cyber-physical production system and, the functional requirements that the designers and developers must respect to implement the new software. From these functional requirements derived a list of design principles that must be fulfilled to design and develop a system with these characteristics.

Subsequently, to evaluate solutions that aim to be minimally invasive, an evaluation model based on a fuzzy inference system is proposed, which rank the approaches according to each of the design principles and globally.

In this way, the proposed work presents the functional requirements, design principles and evaluation model of minimally invasive cyber-physical production systems, to increase the adoption of such systems.

Keywords: Cyber-Physical Production Systems, Multi-Agent Systems, Intelligent Manufacturing Systems, Minimally Invasive, Evaluation Model

Acronyms

AI	Artificial Intelligence
AIS	Artificial Immune Systems
BD	Big Data
CC	Cloud Computing
CPPS	Cyber-Physical Production System
CPS	Cyber-Physical System
CS	Computer Science
CSk	Complex Skill
DP	Design Property
DPApp	DP Application
ERP	Enterprise Resource Planning
FR	Functional Requirement
H_BCA	B Cell Agent (Self-healing experiment)
H_CPA	Cure Provider Agent (Self-healing experiment)
H_DA	Grouped Diagnosis Agent (Self-healing experiment)
H_GDA	Diagnosis Agent (Self-healing experiment)
I4.0	Industry 4.0
ICT	Information and Communication Technologies
II	Industrial Internet
IM	Inference Model
InvD	Invasiveness Deegre
IoT	Internet of Things
M_CMA	Component Monitoring Agent (Self-Monitoring Experiment)
M_DA	Deployment Agent (Self-Monitoring Experiment)
M_HLCMA	Higher-Level Component Monitoring Agent (Self-Monitoring Experiment)
M_OCA	Output Coordinator Agent (Self-Monitoring Experiment)
MAS	Multi-Agent System
MES	Manufacturing Execution System
MF	Membership Function
ML	Machine Learning
MST	Manufacturing Science and Technology
PLC	Programable Logic Controller
R_CA	Component Agent (Reconfigurable Experiment)
R_DA	Deployment Agent (Reconfigurable Experiment)
R_HMIA	Human Machine Interface Agent (Reconfigurable Experiment)
R_PA	Product Agent (Reconfigurable Experiment)
R_PMA	Production Management Agent (Reconfigurable Experiment)
R_PSA	Prime System Agent (Reconfigurable Experiment)
R_SMA	Skill Management Agent (Reconfigurable Experiment)
SCADA	Supervisory Control and Data Acquisition

SF	Smart Factory
SME	Small and Medium Enterprises
SoS	Systems of Systems
SSk	Simple Skill

Content

Table of Figures

Table of Tables

1. Introduction

In this section, a brief overview of the work developed is presented. Firstly, it is presented and explored the current situation and a contextualisation to the problematic. Afterwards, this section introduces an overview of how the research proposed aims to address and contribute to the topic.

1.1. Background

Nowadays, the world is facing a challenging era due to the developments and evolution in the last decades. This new world created a very different and complex social-economic reality. Through the development of Information and Communication Technologies (ICT), a new interconnected world has been created. This new interconnected world increased the availability of products and information. Hence, it increased the demand for highly customised and personalised products. Furthermore, this also generated a global market and increased the delivery of products with different quality requisites. Hence, the companies need to deliver customised products as well as to guarantee that the highly customised products fulfil the quality required.

Additionally, the interconnectivity has created a market where all the manufacturers can deliver products for all the possible clients. This allows the companies to increase the number of possible clients, but at the same time increases the number of companies which can deliver the same product. This creates a higher and very competitive market. To overcome this situation, the companies need to reduce as much as possible the time necessary to produce new products and to start new projects in order to take advantage of possible business opportunities. The companies with the most reduced time to market will take advantage of it.

Another topic that is forcing the companies to change their view is the necessity to implement approaches that reduces environmental impact. In this case, the companies are being forced to reduce the waste during production (raw materials, water, energy consumption, and so forth). Likewise to develop strategies to reduce the impact of the product after the end-of-life or, if possible, to reuse the old product in order to produce and deliver new ones. This approach of using the old product, after the end-of-life, to produce new products, through the usage of the materials or the refurbishment is called the circular economy (Prieto-Sandoval et al., 2017). The circular economy also allows the companies to reduce the costs and increase their productivity. However, the systems required to handle these functionalities are highly complex.

The new requirements and characteristics of the market force the companies to find and adopt new solutions. The existent production lines, based on the third industrial revolution and prepared to deal with mass production, are not able to deal with the new requirements. The production lines running these days are highly rigid and prepared to deliver as many products as possible in a reduced period of time. However, they are only ready to produce a reduced and limited number of variants of products. Hence, the existent lines are not able to deal with the disturbances presented in the previous paragraphs. Moreover, the adaptation of the existent production lines requires changes that imply long stoppages and costs. That adaptation requires the change of the entire control logic and, in some cases, the hardware also. These changes require the design of new hardware and afterwards long periods of integration. The integration of the new hardware requires an enormous effort regarding the reprogramming of the control logic. Whenever a new change is required (to produce new products, change parameters regarding quality, and so forth) the entire process needs to be performed again.

Furthermore, new functionalities can be developed to help companies to optimise their systems. Production lines, due to their complexity and the number of tasks performed in all the stages, are systems that generate a massive amount of data. Usually, this data is not collected, and in other cases, where this data is collected, minimal useful information and knowledge are generated based on it. Based on this data it is possible to optimise the production line in so many aspects, such as in the reduction of waste, better allocation of resources, among others.

New production paradigms have been proposed by the academia to deliver new highly flexible and adaptable production approaches capable of dealing with the production disturbances emerging already now and the challenges of the future. Production paradigms like Bionic or Holonic Manufacturing Systems aim to deliver approaches to control the production process with self-organised and optimised solutions where the decisions are taken on the fly without rigid and immutable control logic. Hence, emerging technologies have been presented and adopted to bring new possibilities for the shop-floor control and optimisation. Technologies like Multi-Agent Systems (MAS) and Service Oriented Architectures (SOA) based software allowed the development of new approaches to performing control, reconfiguration, monitoring, among other functionalities. Although these new solutions presented promising results to tackle the problems of the companies, in fact, the solutions always faced issues with both hardware and technology limitations. Usually, the complexity of the solutions forced the usage of hardware with high computational capacity, networks with the capacity to link several nodes (large bandwidth), among other requirements. These restrictions, during an extended period, made impossible the deployment of such solutions in real industrial environments.

However, with the improvement of the ICT in the last years the proposed production paradigms started to be implemented, tested and deployed in industrial environments. Hence, the concept of Cyber-Physical Production System (CPPS) emerged.

The CPPS encompasses the virtualisation of the physical world (shop-floor resources) in a one-to-one representation. This means that each component interacting in the physical world is abstracted in the logical world (cyber world) through a software entity. The software representation can interact with the abstracted physical component, in the cyber world, and communicate with other cyber representations. With this approach, it is possible to create communication channels between any physical component type, regardless of the vendors or specific communication protocols, since it is through the cyber world exclusively that the communication amongst the cyber-physical entities is performed.

Through the virtualisation of the physical world into the cyber world, the CPPS approaches allow the interoperability of the different actors. This virtualisation increases the flexibility and adaptability of the different functionalities (such as hardware or software functionalities) since the communication among the different actors is done through the cyber world exclusively. Moreover, the effort necessary to integrate and change the system's components and tools is dramatically reduced.

Due to the design proposed for the CPPS, based on the abstraction of each component with a logical representation and the capacity to optimise the entire system through the collaboration among them, some implementations were proposed using MAS approaches. The concept of MAS proposes a distributed software where all the agents are autonomous and capable of communicating with each other in order to solve problems and optimise the system collaboratively.

In an agent-based CPPS, an agent (software) abstracts each production resource (hardware), software tool and the factory itself, which it can coordinate and know everything about that component. Although the agent-based solutions proposed so far try to solve the presented challenges and problems, in fact, the solutions are not well accepted. They tend to be deployed during a small period or in most of the cases they are not deployed at all (Bussmann and Schild, 2001; Leitão et al., 2015; Mönch et al., 2003).

Although the companies see the added value of the solutions proposed so far, the changes required to adopt them make their application difficult and in most of the cases impossible. Since the factories are already established, with all the existent control logic and functionalities already running, the application of such solutions implies not only a financial effort but also stoppages, wastes regarding existent hardware and tools, and the necessity to train the personnel about these new systems and available functionalities. All these steps and effort required to adopt the new approaches force the companies to decline this possibility.

Hence, it is essential to analyse and assess which are the main difficulties and challenges found during the deployment of the solutions proposed. Moreover, it is necessary to understand which strategies can be delivered in order to close this gap and increase the adoption of these systems.

The work proposed in this document aims to assess the existent challenges as well as the definition of a strategy and path for the development of agent-based CPPS that imply fewer changes and increase the adoption of these solutions. Firstly, it is crucial to collect which requirements need to be fulfilled by the solutions in order to be considered more attractive for the companies. Furthermore, a list of Design Properties (DP) will be presented. The DPs are generic and applicable to the development of many agent-based CPPS. Hence, whenever it is necessary to design and develop an agent-based CPPS, with a reduced impact on the existent production line, the DPs presented in this document can guide that work.

With the DPs presented in this work the author expects to deliver a contribution, in the development of agent-based CPPS, that can increase the adoption of such solutions in the real industrial scenarios. Furthermore, an evaluation model is presented in order to allow the researchers, designers and developers the ability to evaluate the solutions based on the invasiveness and effectiveness.

1.2. Research Problem

The literature review presents a vast number of solutions to overcome the challenges previously presented. In some cases, it is possible to verify that those solutions develop higher-level functionalities which can be delivered without requiring major changes and adaptations within the existent production facilities. These higher-level developments mainly work in the cyber world. In the cyber world, it is possible to deliver solutions with the capacity to process big amounts of data and run advanced algorithms. These approaches allow the development of solutions capable of optimising the allocation of resources, reduce the energy consumption, reconfigure the control logic, detect maintenance and quality issues among others functionalities.

With the development of solutions, mainly focused on the cyber world, it is possible to reduce the impact and the changes required to deploy a new agent-based CPPS in an existent facility. Hence, the author believes that it is possible to collect generic functionalities that must be delivered by the cyber world in order to reduce the impact of the new solutions.

Moreover, after the assessment of the functionalities required to deliver a system with a minimal impact, it is possible to deliver a list of good practices to guide the design and development of agent-based CPPS. The author believes, that with a collection and definition of DPs to guide the development of future systems in order to reduce the changes required for them will increase the adoption of these systems within the industrial sector.

1.2.1. Research Questions and Hypothesis

It is essential to study and assess which characteristics and principles an agent-based solution should have in order to reduce as much as possible the impact on the existent infrastructures. This problematic raises the following research question:

> Q1: Which are the main design properties that should be considered when designing a minimally invasive agent based cyber-physical production system using traditional manufacturing environments with discrete control?

The previous research question, Q1, can be addressed according to the following hypothesis:

H1: If the developed agent-based cyber-physical production systems, aimed to deliver new functionalities to the existent manufacturing systems with discrete control without requiring significant changes and adaptations, share common properties and characteristics, it is possible to provide a list of design properties that must be followed in the development of such systems.

The previous hypothesis, H1, raises an additional research question, related to the evaluation of the proposed minimally invasive agent-based CPPS:

Q2: How it is possible to evaluate the design and development of a minimally invasive agent-based cyber-physical production system?

The previous research question, Q2, can be addressed according to the following hypothesis:

H2: It is possible to evaluate the design and development of minimally invasive agent-based cyber-physical production system through the definition of an evaluation model capable of evaluating the implementation of each design property separately, and posteriorly evaluate the implementation based on the usage of all design properties. This model can be achieved through the development of a fuzzy inference system-based model.

1.3. Aimed Contributions

The previous research questions and the particular hypothesis will contribute to several aspects. The contributions (C) are:

- C1 - *Analysis of the application of multi-agent systems' (MAS) for the development of CPPS*: The proposed research aims to explore the current state of the art regarding the application of agent-based solutions to develop CPPS in order to improve the manufacturing processes. With this contribution the author intends to assess the possibilities and challenges verified during the development and deployment of these solutions;

- C2 – *Definition of minimally invasive agent-based CPPS*: Before the introduction of a strategy to reduce the impact of the agent-based CPPS, it is necessary to understand which are the characteristics of a minimally invasive system and, how it is possible to define a minimally invasive agent-based CPPS. Hence, during this work, it will be proposed a definition of minimally invasive agent-based CPPS. This definition will help and guide the work introduced in the document;

- C3 – *Functional Requirements (FRs) elicitation for minimally invasive agent-based CPPS*: One of the most critical and important steps during the development of the research proposed in this document is the correct and effective elicitation of a set of FRs. In this case, the functional requirements collected aim to describe which functionalities need to be delivered by a new agent-based CPPS in order to be as less invasive as possible. The collection of FRs requires conversations and meetings with industry experts, in order to generate requirements that effectively describe the functionalities and specifications required by the industry;

- C4 – *Design and development of minimally invasive experimental prototypes*: It is important, in order to assess the solutions' design and the FRs fulfilment defined in C3, to design and develop experiments to define how the solutions must satisfy the collected requirements. Three experiments are proposed in this work. The experiments aim to generate Design Properties (DPs) specific for each experiment.

- C5 - *Definition of generic DPs for the design of minimally invasive agent based CPPS*. With all the lessons learnt during the design and development of the experiments (C4), in conjunction with the three lists of specific DPs, it will be possible to create a list of generic DPs. This list of DPs is one of the main contributions of the proposed work. The objective is to formulate a list of good practices and guidelines to help during the design of minimally invasive solutions.

- C6 – *Model to evaluate minimally invasive solutions*: Even with a well-defined list of DPs to guide the development of a minimally invasive agent-based CPPS, it is complicated to evaluate and quantify the invasiveness and the application of the DPs of a specific solution. Hence, it is proposed a fuzzy inference system-based model capable of to evaluate the solutions. The evaluation model evaluates the application of each DP and the solution as a whole. Hence, it is possible to verify the solution as well as the most critical DPs which must be verified in order to improve the overall solution.

1.4. Integration with other Research Activities

The proposed work is based on previous experiences, as presented in Figure 1, namely European research projects from the already finished FP7 work program. The basis of the proposed research is the usage of the agents to abstract and optimise the shop floor functionalities. This approach has been widely explored and used in the FP7 IDEAS project (Onori et al., 2012).

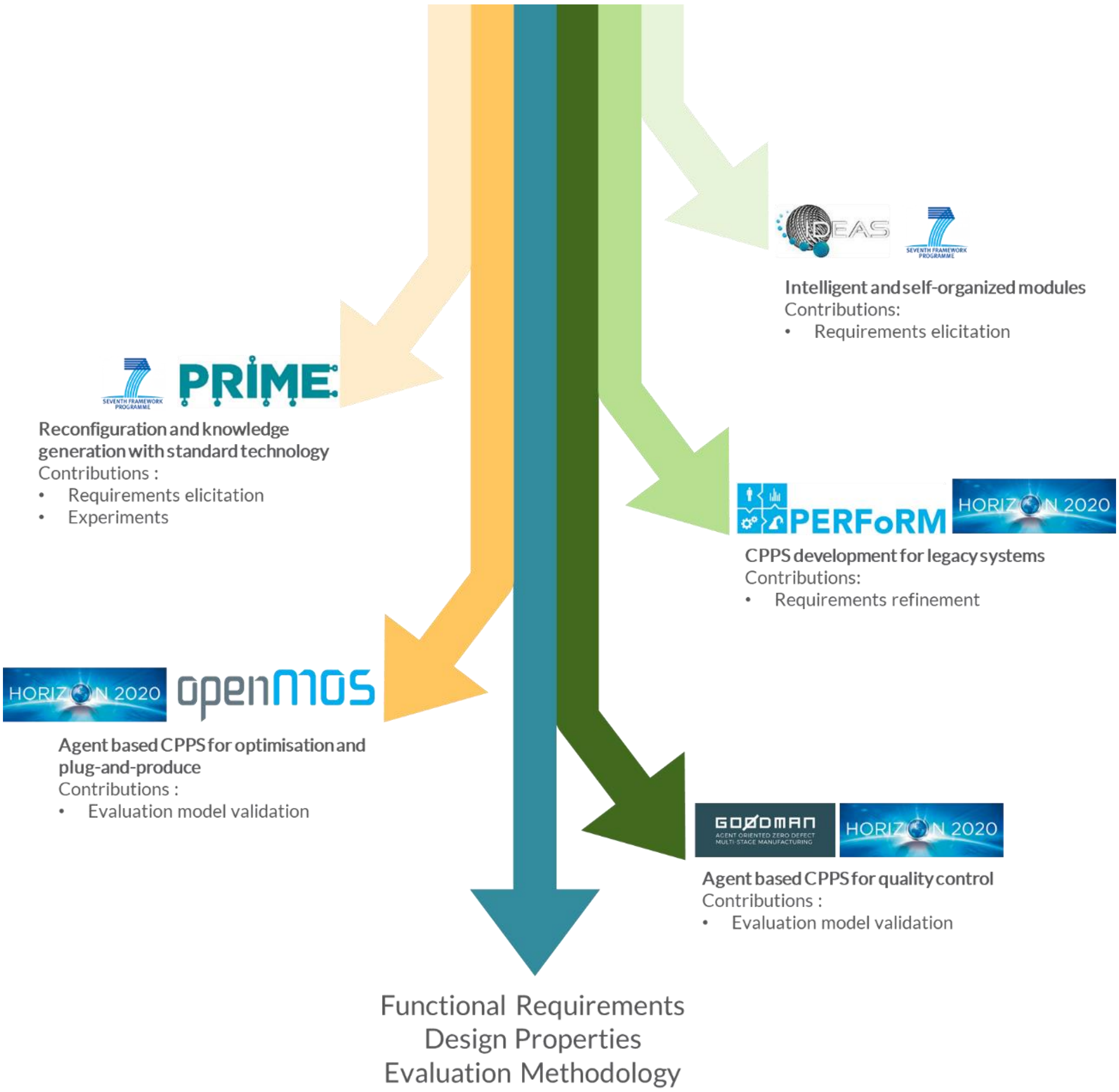


Figure 1 - Integration with other research activities

The IDEAS project developed under the FP7 work programme used the concept of Systems of Systems (SoS) to abstract the shop floor in different levels and functionalities, and coordinate the execution at different granularities. This concept is particularly important to define the agent-based systems at different levels of complexity and comprehension. Based on the complexity of the solution proposed in IDEAS and the possibility to discuss with the industrial partners, the IDEAS project was essential to collect FRs.

Also, under the FP7 work programme, where most of the work presented in this document was developed, the PRIME project (Rocha et al., 2014) used the previously presented concept of SoS to design multi-agent environments. Those environments were capable of reconfiguring the control logic, without directly interfering with the process control neither the execution. Another relevant functionality developed under the PRIME project was the capability to generate knowledge based on raw data dynamically. These two agent-based CPPSs, developed under the PRIME project are the basis for two of the experiments presented in this research. The PRIME project also contributed to the FRs elicitation and in the development of two out of three experiments designed during this work.

The H2020 PERFoRM project (Leitão et al., 2016a; Peres et al., 2016) primary focus is not to deliver agent-based solutions. However, this project is an interesting environment to discuss with the partners in order to assess if the FRs defined in the previous projects must be refined or improved.

The H2020 openMOS (Danny et al., 2017) and H2020 GO0D MAN (Utz and Lee, 2017) projects from the Horizon 2020 work programme, are relevant to the research. Both projects aim to develop agent-based CPPS requiring as less as possible changes in the existent facilities. The openMOS aims to develop an agent-based CPPS to optimise the production and reduce the energy consumption. On the other hand, the GO0D MAN project focuses on the development of an agent-based CPPS capable of dealing with quality issues in a multi-stage production system. These two cases will be evaluated using the purposed evaluation model, and they will also allow the possibility to validate the evaluation model itself.

1.5. Research Methodology

The proposed research follows the traditional research methodology. However, the traditional approach diagram was slightly adapted. The adapted methodology aims to merge the traditional approach with an inductive reasoning approach, in step five (Goddard and Melville, 2004). In step five it is expected to study the patterns and generate new knowledge based on the assessment of the three experiments.

The research methodology proposed encompasses eight steps, as depicted in Figure 2.

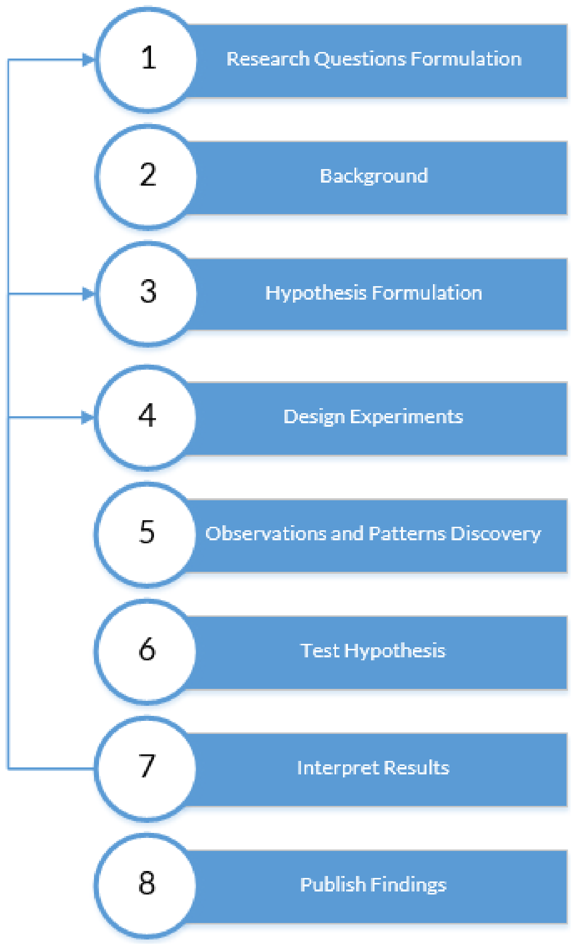


Figure 2 – Adopted research methodology

The process starts with the research questions formulation. In this stage, the author defines the focuses of the research, formulating the research questions that present the problem. In this particular case, stage one was already presented, in the previous section. Q1 and Q2 are the results of this first stage.

After the research question formulation, in stage two, it is defined the background. In this stage, it is important to define all the information relevant to the research and the current state of the art regarding the topic and the problematic. The work developed in this stage is presented with an extended literature review. This literature review focuses on all the topics relevant to the work, such as industrial agents or CPPS.

With the background finished, a set of hypotheses was proposed. Usually, the number of the hypothesis is equal to the number of research questions. The result of this stage was already presented in section 1.2.

The fourth stage encompasses the definition of the experiments. For this specific work, there are three experiments which will be presented posteriorly. The objective, with these three experiments, is to design and implement different minimally invasive agent-based CPPS. The result of each experiment is a list of design properties. Each list describes the design properties used for each specific experiment.

Stage five is an addiction to the classical research methodology due to the specificity of the research developed, which aims to assess and observe the three experiments. Through the assessment of the three experiments, the author aims at finding similarities and partners among them. Afterwards, it will be proposed a list of general characteristics of the experiments, as presented in Figure 3.

Figure 3 - Research methodology step five detailed

In stage six, the generic characteristics generated from the previous step will be validated and tested. During this stage, two additional test cases will be used to assess the results obtained in the previous stage.

In the subsequent stage, the results will be analysed and verified according to the hypothesis defined in stage three. If the results do not fulfil the defined hypothesis, the process returns to previous stages. Depending on the results, the process can return to the research questions or hypothesis formulation or can return to the experiments definition.

When the hypotheses are verified, it is essential to publish the results obtained. The last stage focuses on this topic. In order to effectively disseminate and increase the impact of the results, the following list of international conferences (Figure 4) and journals was created (Figure 5).

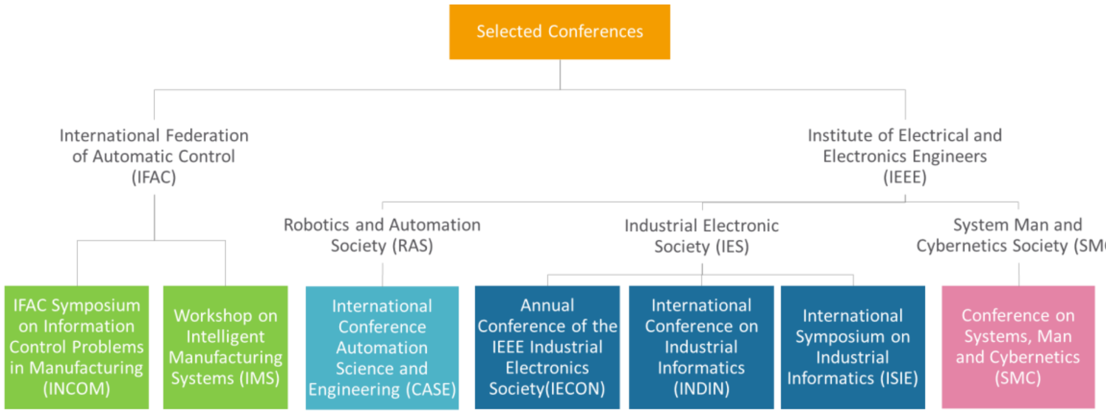


Figure 4 - International conferences relevant to the work

The selected conferences and journals are relevant for the area and indexed in databases that will increase the impact.

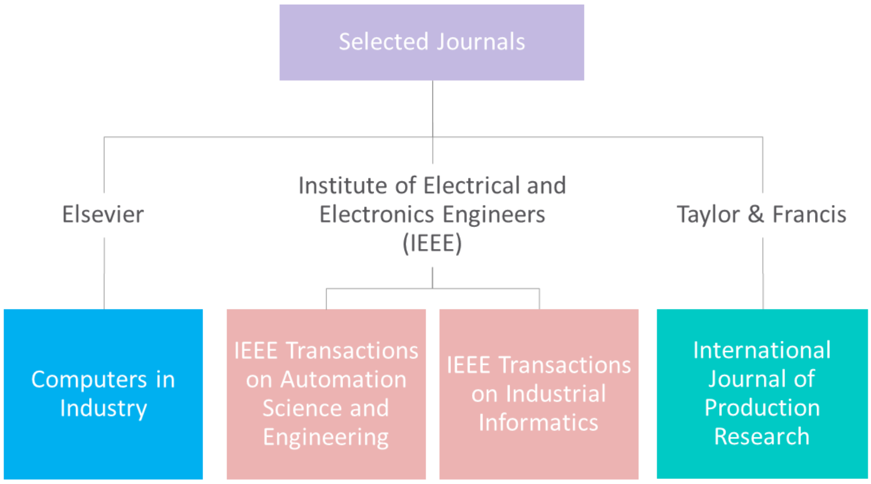


Figure 5 - Journals relevant to the work

1.6. Document Organisation

This document is divided into seven sections. This first Chapter presented the problematic, the objectives of the work and a document overview. This chapter is sprightly related and presents the results accomplished during stage one and three of the research methodology.

Chapter two presents an extended literature review of the relevant topics for the work developed during this research. The literature review chapter presents the last developments in relevant topics like the fourth industrial revolution, CPPS, industrial agents, among others. The assessment of the current state of the art will create the foundations for this work, and it presents the results of stage two of the research methodology.

Chapter three presents all the information related to the elicitation of the FRs. Firstly, the method to discuss and collect the FRs is presented. Posteriorly, a list of FRs for the cyber system and a list of FRs for the existent native system is presented. Also, at the beginning of this chapter the definition of cyber and native system is presented.

In chapter four the design and development of the three experiments are presented. One of the experiments focuses on reconfiguration, another in self-monitoring and the other one focuses on self-healing. For each experiment, a list of design properties is presented. At the end of this chapter, a list of generic design properties is proposed. This list of generic design properties aims to guide the design and development of any minimally invasive agent-based CPPS.

After the collection and definition of the generic design properties, in chapter five an evaluation model, based on Fuzzy Inference System (FIS), to assess each solution is presented. This model intends to evaluate any design and development of a minimally invasive approach. The evaluation model will be used to assess the experiments as well as the two external test cases.

Chapter six shows the evaluation of these cases and explores the obtained results. The idea of using the proposed solutions with the openMOS and GO0D MAN projects is to validate and assess the DPs as well as the evaluation model itself.

The document ends with the concluding chapter that summarises the developments and enumerates the open issues that must be explored in future activities.

2. Literature Review

This chapter contains a literature review on the main topics related to the proposed research in this work. In the first chapter, current trends and last developments are presented, supported by the concept of I4.0. Being one of the leading topics regarding I4.0 applications, the CPPS are subsequently explained. In order to compare the advantages of using CPPS and industrial agents, it is also summarised the current situation and technologies applied to the existent production systems. The last chapter introduces the concepts of agent and MAS and also surveys the application of such technologies for industrial purposes.

In the last decades the evolution of the technology, mainly the information technologies, created an interconnected world. This interconnected world increased the awareness of the population about new possibilities and realities. This new world created also a global market, where all the customers can acquire products from any company around the world, and vice-versa, any company can enter in any country and any market. This new global market also demands for products with high-quality standards, more personalised and cheaper. Moreover, these new customers are also aware about social threats, such as the environmental impact of the processes required to produce.

These new constraints force the companies to change how they produce and face the market dramaticaly. The existent and deployed mass production approaches are not able of dealing with this volatile and rigorous environment where the demanding changes rapidly and regularly. The current production systems, resulting from the third industrial revolution, are mainly founded in automated systems with high performance. However, these rigid systems are no capable of rapidly changing and adapting. Hence, whenever a change is required (produce a new variant of products, change quality requirements, and so forth), it is necessary to request external personnel to do it. For instance, in most of the cases, it is necessary to design new hardware and change the control logic.

Hence, the companies are facing difficult times due to the requirements and needs imposed by a global and volatile market. The companies need solutions capable of reducing the time necessary to produce a new product or variant (time-to-market) and increase the ability to produce small batches, mainly constituted by highly customised and personalised products. Furthermore, these functionalities must also reduce the price of the products and reduce the environmental impact necessary to produce them.

2.1. Fourth Industrial Revolution

These days the fourth industrial revolution is taking place (Figure 6), making use of the recently emerged ICT technologies, such as the Internet of Things (IoT) or Cloud Computing (CC). This revolution, designated in Europe by I4.0 (Deloitte, 2014; Gilchrist, 2016; Kagermann et al., 2013) or in the US as Industrial Internet (II) (Consortium, 2015), aims to develop a vastly connected industrial environment. In this interconnected ecosystem, all the entities and equipment present in the entire value chain are connected and sharing valuable information dynamically among them. In Figure 6, the evolution of the industrial sector along the four revolutions is shown.

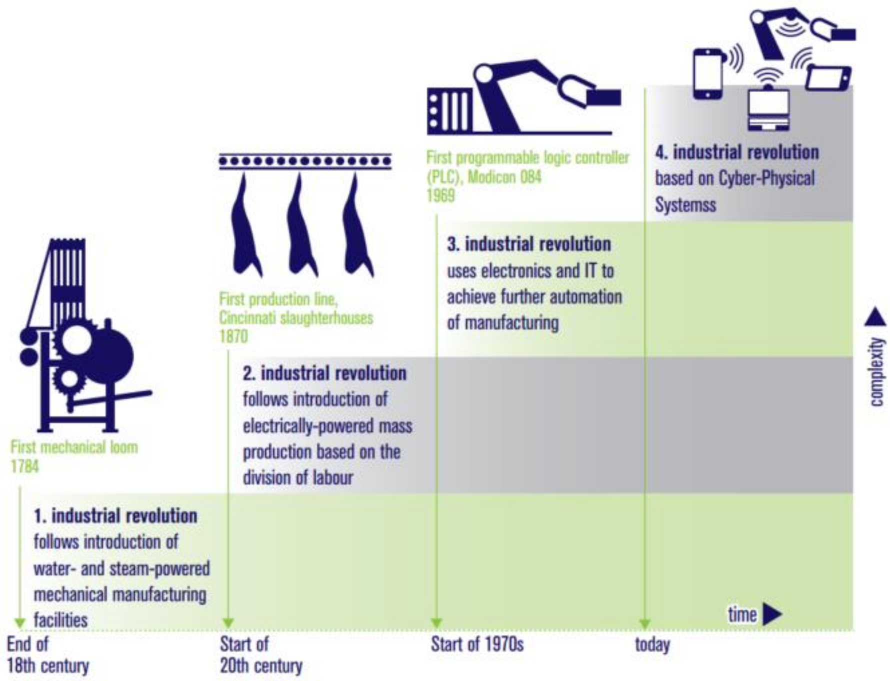


Figure 6 - The four stages of the industrial revolution (Kagermann et al., 2013)

This interconnected industrial world can be described and defined by three dimensions, as stated in (Stock and Seliger, 2016):

- *"Horizontal integration across the entire value network"*: This integration allows the communication and optimisation among the different companies. value chain's actors (see Figure 7), such as suppliers, logistic, customers, and so forth. Hence, with the information's digitalisation, the different sectors of the company can dynamically and in real-time share valuable information and adapt to the different needs along the value chain.

The horizontal integration aims to interconnect all the value chain's actors, in order to better coordinate and adapt the entire value chain. Henceforth, all the factories act like an intelligent and active node in this complex ecosystem. Whenever a factory needs to request services or materials from another entity, that request is triggered for the supplier autonomously. The selection of the supplier is done according to the factory and the ecosystem optimisation, based on the available resources.

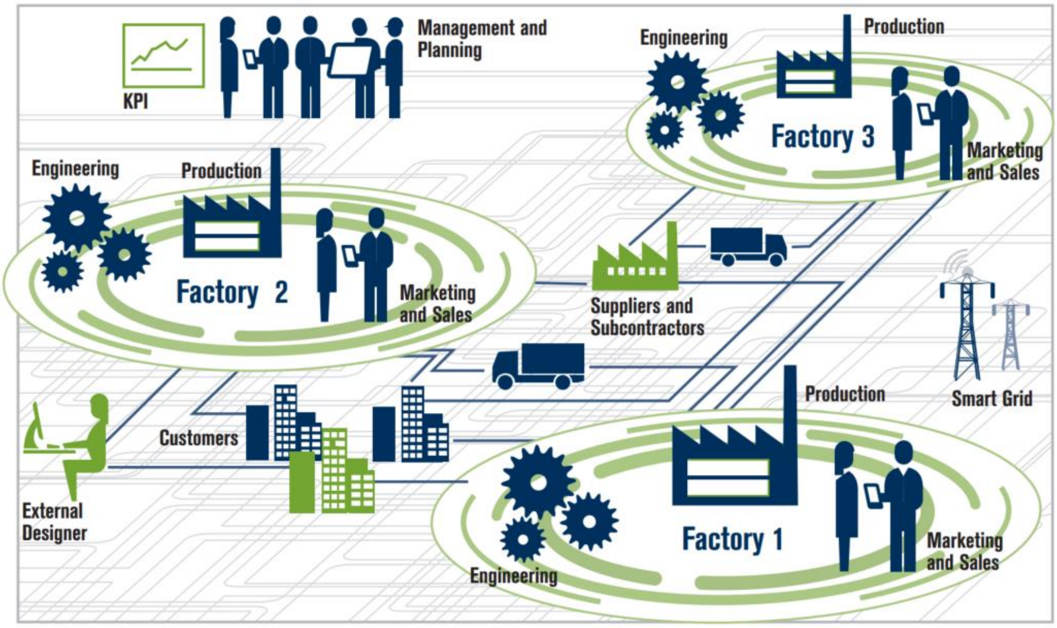


Figure 7 - Horizontal integration in an Industry 4.0 based context (Kagermann et al., 2013)

- *"End-to-end engineering across the entire product life-cycle"*: The digitalisation of product's information allows the companies to collect and use relevant data from the products during the entire product life-cycle. Hence, from the moment the company receives the raw material until the product 's life ends, all the relevant information is extracted and shared in order to improve the value chain (product design, suppliers' selection or process).

- *"Vertical integration and networked manufacturing systems"*: The vertical integration focuses on the intelligent integration and cooperation among the entities on the shop-floor. The entities' cooperation creates itself a value creation module in the entire value chain.

The vertical integration, as presented in Figure 8, aims to interconnect all the resources and software inside the factory. This interconnection allows the system to be more flexible and to deal with disturbances more efficiently and rapidly. Moreover, the integration among the different factory's areas permits the optimisation of the entire factory's ecosystem, through the coordination of resources to optimise the resources itself, raw material, among the others.

Figure 8 - Vertical integration in an Industry 4.0 based context

Merging the three characteristics presented, the concept of Smart Factory (SF) emerged. A SF constitutes a node in the entire value chain network, and it is responsible to autonomously manage and optimise the internal ecosystem (vertical integration). The role of a SF, where the SF is implemented as a CPPS for internal management of the factory and integrates a complex ecosystem with all the value chain nodes, is shown in Figure 9.

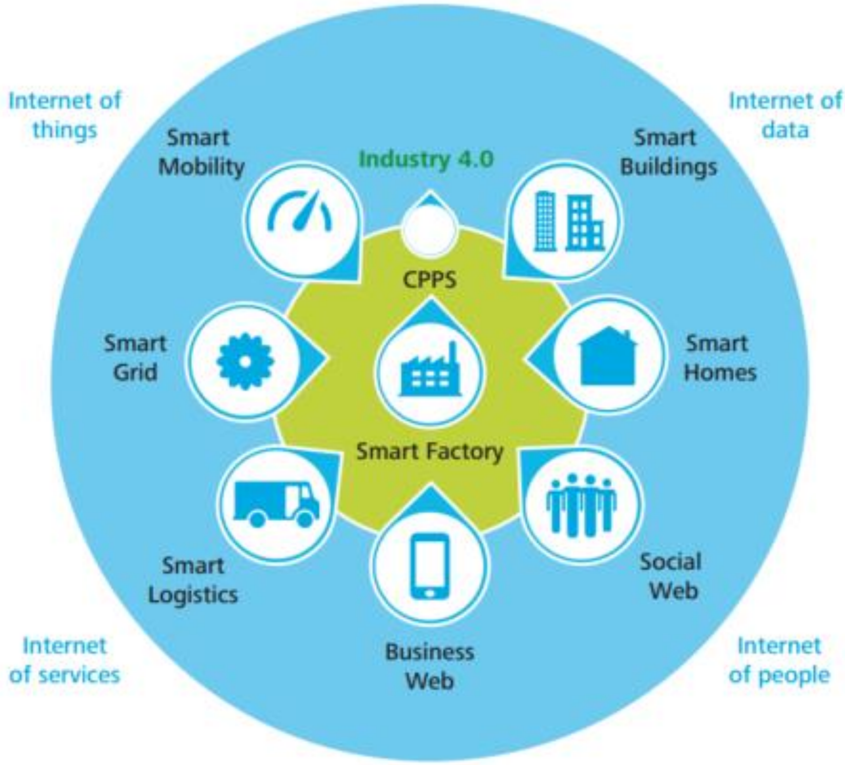


Figure 9 - Industrial internet environment (Deloitte, 2014)

This digitalised and interconnected factory periodically sends and receives information from and to the external world. However, nowadays, this idea of the external world no longer exists. All the entities, and consequently, the SFs are connected to the internet, sharing information with all the modules which could constitute an added value for the entire value chain. SFs, through the internet to Smart Grids, Smart Logistics, Business and Social Web, are connected. This interconnectivity will allow the factories to adapt to external events dynamically. The demanding for a specific product, markets' trends and societal limitations will also allow the factory to autonomously ask for the necessary materials and parts in an optimised way, avoiding delays and problems. Henceforth, through the vast available resources, the factory is more able to deal with constant disturbances and unexpected demands.

According to (F. Shrouf et al., 2014) the following characteristics must be taken into account in order to improve the SFs' sustainability :

- *"Mass customisation"*: With the current volatile market and with more and more exigent customers, the factories of the future must deal with constant market disturbances. The demands unpredictability, quality requirements for the new products' variants increasing and the desire for more specific and personalised products from the customers, the batches' sizes will decrease, and in an extreme situation, the factories could need to handle size one batches (Kagermann et al., 2013).

- *“Flexibility”*: Through the development of intelligent production processes, highly dynamic and capable of dealing with volatile conditions, it is possible to deliver flexible solutions aimed of self-configuring the production resources and capabilities according to relevant aspects, as presented in (Zuehlke, 2010).

- *“Factory visibility and optimised decision-making”*: In an interconnected factory, wherein all the components and stations are connected, it is possible to improve the overall efficiency of the factory. This is achieved through the end-to-end communication and cooperation among the different actors. The factory can optimise itself, e.g. reducing wastes and reducing energy consumption.

- *“New planning methods for factories”*: Consuming the digitalised information, extracted from the manufacturing system, such as availability, performance and energy consumption, can create in the factory the ability to plan present and future activities. This ability will optimise the usage of the physical resources (Zuehlke, 2010). In (Fadi Shrouf et al., 2014) an approach to reduce the energy consumption, based on stations’ usage, is presented. The solution uses the extracted data to optimise the scheduling of future activities.

- *“Creating value form big data collected”*: The manufacturing systems are environments that generate significant amounts of data. Currently, the data is not collected, or in some cases, the data is extracted but not processed and analysed. In the future, throughout the use of IoT devices, all the extracted data can be easily stored in the cloud, where the processing resources are not a problem. Processing this amount of data without problems will enable the factories to generate valuable information that will allow creating knowledge that could and should be used for its optimisation. (Krumeich, Jacobi, Werth, & Loos, 2014; Wang, Wan, Zhang, Li, & Zhang, 2016).

- *“Creating new services”*: The interconnectivity with society, in general, will allow the companies to offer new types of services. The information sharing throughout the cloud will permit the development of smart devices or applications that will add value to the products from the customer’s perspective.

- *“Remote monitoring”*: The data extracted from the shop floor can be used for monitoring and check the state of the factory at any time and any place. This functionality allows the factory to consume different maintenance services that usually are not feasible. For instance, the remote and more efficient maintenance, performed by external suppliers, will be possible with the plus of cost reduction, in comparison with today's' solutions (Iung, 2003).

- *"Automation and change role of man"*: The role of the human, in these new intelligent factories, is not the same as in traditional factories. The intervention of the human in the mechanical process is reduced, which will result in a reduction of defects and an increasing of the system's efficiency. However, the human role cannot be underestimated, since it is the most flexible entity in the entire system.

- *"Proactive maintenance"*: In most of the cases, the data extracted from the manufacturing hardware can be used to predict future problems dynamically and, in this sense, it is possible to implement more efficient maintenance approaches. This behaviour allows the manufacturing companies to reduce time usually required for maintenance tasks and consequently reduce the costs associated with maintenance, usually performed in a reactive way. The analysis of the extracted data will also reveal possible performance and quality issue that somehow could be related to maintenance problems (Mobley, 2002). Self-capabilities, as awareness (Snowden, 2002), healing (Psaier and Dustdar, 2010) or learning (Di Orio et al., 2015) combined with BD processing (Lee et al., 2013) are essential enablers to introduce proactive maintenance.

- *"Connected supply"*: In a SF, the entire supply chain is coordinated by a smart logistics functionality. This functionality is responsible for managing the factory needs and the available suppliers. The coordination of the materials' requests is based on the quality of the required materials, the stock of the materials and time to deliver. Through the internet, the factory can automatically ask for the necessary materials and decide which supplier or suppliers are more appropriate to deliver a specific material. The interoperability also enables the communication among the different transport systems. This communication among them will create an optimised transport system where each transport system behaves as a supplier of the next one (Fox et al., 1993).

- *"Energy management"*: The energy consumption is becoming a crucial indicator for the manufacturing systems. The impact of the energy consumption not only influences the costs but also reduces the environmental impact. The management of the energy consumption also encompasses the management and selection of the energy suppliers. The energy delivered to a SF is transported by a smart grid which is capable of providing the right amount of energy with minimal cost and environmental impact (Amin and Wollenberg, 2005; Farhangi, 2010).

The concepts of interconnectivity and distribution of the information will push companies, and all the different departments to do a massive paradigm shift, in order for the companies to deal and retrieve as much as possible profit from the current industrial revolution. In this stage, the human will have a vital role as flexibility enabler. The capability to adapt and cooperate among the different sectors of the companies will define their success in this new stage. According to (Schuh et al., 2014):

"The impact of the fourth industrial revolution, however, is more extensive and it affects apart from production also the indirect departments, especially engineering processes. That means that the potential of productivity growth particularly lies in the improvement of brainwork and decision-making processes. Collaboration at all levels can help to accelerate this process."

2.2. Traditional Production Systems

Currently, the production systems are mainly controlled and operated using stratified and rigid approaches based on the ISA 95 standard pyramid (Scholten, 2007, p. 95), as presented in Figure 10. This standard defines five layers, starting with the production process until the highest abstraction of the system.

This approach comprehends a monolithic and rigid architecture that is very stable and robust, but at the same time does not allow changes. Changes within these approaches imply a vast investment regarding time and money. Moreover, any improvement or adaptation can involve changes in the entire system (for instance, the introduction of new products' variants and changes regarding product and process parameters).

The stratified ISA 95 approach focuses on the following layers:

- Level 0: Production Process: This layer contains the shop floor's physical components, basically all the resources which manipulate and work on the product. In here are all the physical and chemical steps that add value to the product itself.
- Level 1: Sensors and Actuators: The different sensors and actuators which allow the production process to be controlled and perceived make part of this level. Level

1 is responsible for the interface between hardware (Level 0) and software (Levels 2, 3 and 4).

- Level 2: Automated Process Control: Level 2 is responsible for the process control, which means that the execution and the workflow of the production process is achieved through the coordination of this layer that actuates through the interface with Level 1. To do so, the process control is reached through the use of Programmable Logic Controller (PLC) and Supervisory Control and Data Acquisition (SCADA) systems. At this level, and in most of the cases, the control is performed using a sequential steps' implementation, which controls the dispatch of orders, system's monitoring, diagnosis and error recovery, whenever necessary. Although this solution is very robust, it is very rigid either, because all the control logic is very well defined and optimised for each specific case. Whenever something needs to change, most of the functionalities and the logic inside them need to be updated and changed.

- Level 3: Manufacturing Operations Management: The Level 3 usually is populated with the Manufacturing Execution Systems (MES). The MES is responsible for defining, based on the production plan, which machines and remaining resources must be used to work on the products at each time and in which specific tasks. These orders are sent to Level 2, to be executed according to the schedule defined in this level.

- Level 4: Business Planning and Logistics: The highest level of the ISA 95's pyramid accommodates the Enterprise Resource Planning (ERP). This layer is responsible to somehow interact with business personnel, allowing the definition of the production plan, based on market demands, internal goals and so forth.

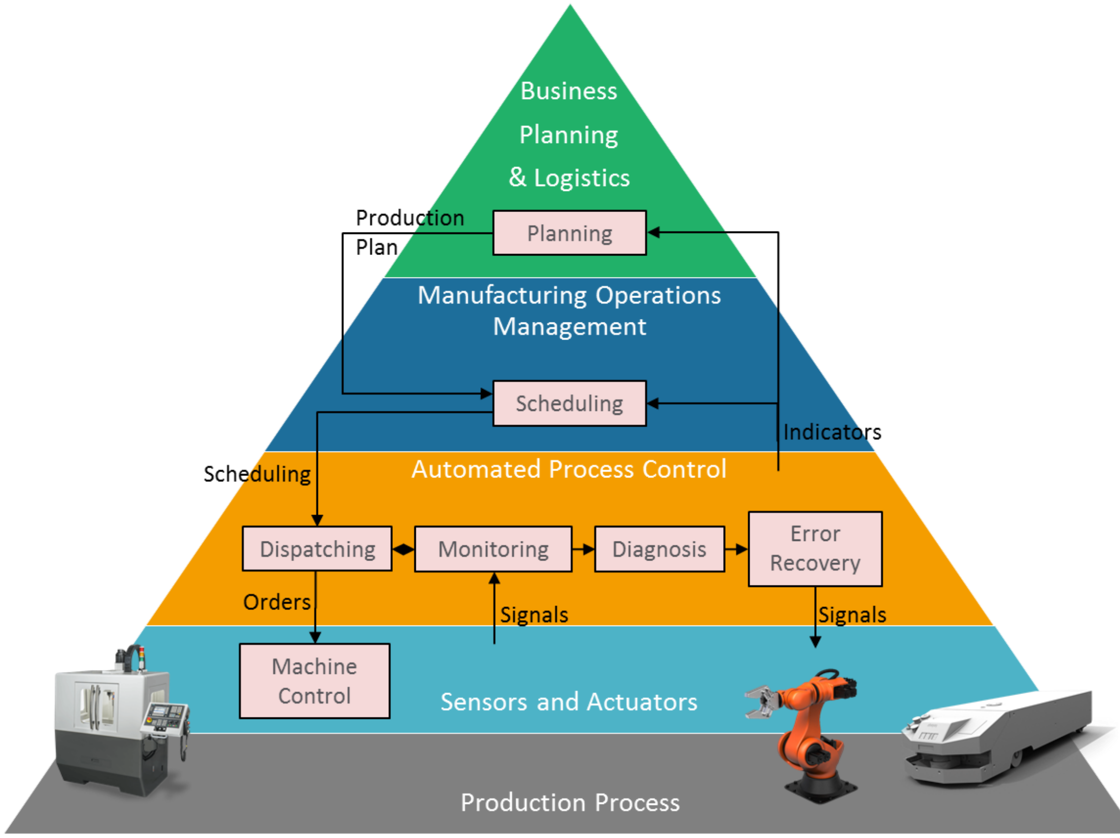


Figure 10 - Traditional manufacturing systems, adapted from (Leitão, 2009)

As previously described, the rigidly of the traditional production systems does not comply with the new market needs. Hence, new intelligent CPPS based on distributed and intelligent systems have been studied and proposed to introduce the capability to deal with these disturbances.

2.3. Cyber-Physical Production Systems

The concept of CPPS emerged from the application and adaptation of the CPS concepts to the production systems. The concept of CPS is recent, and many definitions can be found, according to (Lee, 2008) *"CPS are integrations of computation with physical processes. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa."*

The use of embedded software and intelligence to coordinate and control the hardware is not entirely new. However, it was necessary to wait for the last developments in the ICT to allow the real implementation and development of these embedded solutions. Regarding the manufacturing sector, it is not entirely new the usage of embedded controllers, sensor systems or collaborative robots. However, the limited resources such as controllers' capacity and connection bandwidth strangled the previously proposed solutions.

In a CPS environment, each physical component is virtually represented by a cyber-entity in the cyber world. Only information is exchanged and processed on this level of the system. The cyber and physical parts continuously work in a closed loop to control and monitor the resource and to keep the consistency between physical and cyber levels. Hence, a CPS is composed of several pairs of physical-cyber entities where each cyber entity mirrors the respective physical component, as shown in Figure 11.

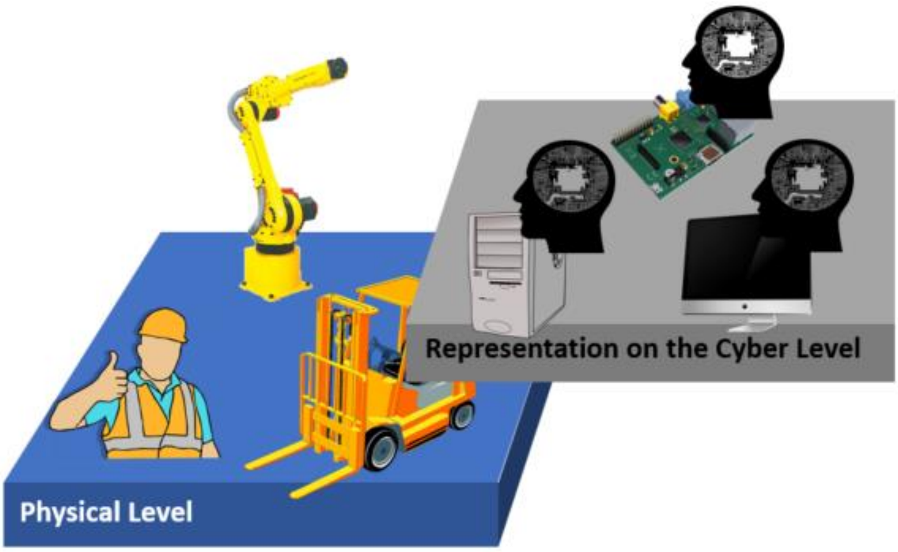


Figure 11 – Cyber-Physical Production Systems

The virtualisation and representation of the physical world through virtual, independent and connected cyber entities, allows the traditional approaches to extend and to improve the system and sub-system levels' capabilities. Through the information sharing among the cyber-entities, it is possible to generate behaviours that emerge from the collaboration and cooperation among the cyber-entities. These behaviours will allow the system to evolve and offer new functionalities.

The applicability of CPS is almost infinite. All the physical systems which could be improved through the cooperation and sharing of information among the entities can be defined and developed as a CPS. According to (Monostori et al., 2016) the last developments in CPS can improve and increase the productivity of several areas, such as:

- Agriculture;

- Energy;
- Healthcare;
- Manufacturing and Industry;
- Society;
- Transportation.

In all the sectors, the representation of the physical components through cyber-entities allows the homogenisation of the information in the cyber world. This homogenisation of the information allows an easy and quick integration of different components.

The usage of CPPS (Monostori, 2014) aims to migrate the concepts of CPS to the industrial environments. Hence, the CPPS allow the development and application of the previously presented SFs, in the context of this I4.0's new era. Regarding the main functionalities, a comparison of today's factories against the I4.0 based factories is presented in Table 1.

Table 1 - Comparison of today's factory and an I4.0 factory (Lee et al., 2015)

	Data Source	Today's Factory		Industry 4.0	
		Attributes	**Technologies**	**Attributes**	**Technologies**
Component	Sensor	Precision	Smart sensors and fault detection	Self-aware Self-predict	Degradation monitoring & remaining useful life prediction
Machine	Controller	Producibility & performance	Condition-based monitoring & diagnostics	Self-aware Self-predict Self-compare	Up time with predictive health monitoring
Production system	Networked system	Productivity & OEE	Lean operations: work and waste reduction	Self-configure Self-maintain Self-organize	Worry-free productivity

In order to achieve such functionalities, it is necessary an interdisciplinary framework responsible for synchronising the work and capabilities of the three very distinct areas, presented in Figure 12.

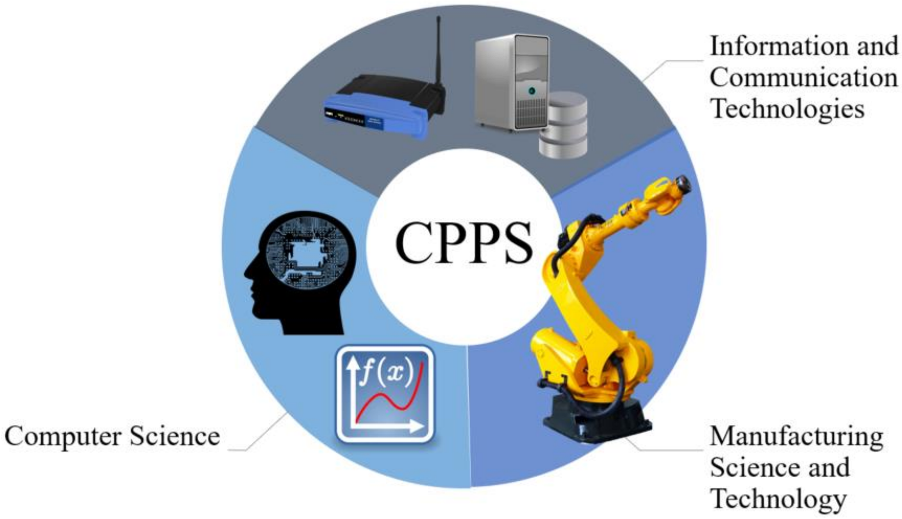


Figure 12 – Interdisciplinarity in CPPS

The combination and overlap of the fields from ICT, CS and Manufacturing Science and Technology (MST) allows the development of CPPS capable of dealing with the new requirements and, as previously presented, implementing the concept of SF.

As presented in Figure 13, it is possible to verify that the evolution of the technology regarding the virtual world, also pushed the development of the production systems, in the figure presented as the physical world.

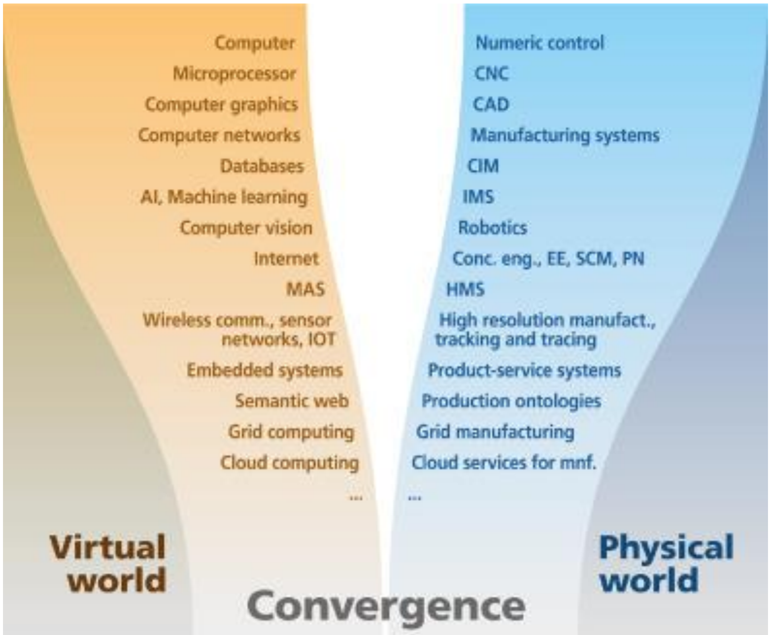


Figure 13 - Interplay between CS, ICT and Manufacturing (Monostori et al., 2016)

The usage of the ICT and CS in the physical world is quite visible in the figure. However, the application of the technologies in industrial environments is very specific and restricted for the industrial systems.

Now, it is possible to verify that the developments on the virtual and physical worlds tend to converge and deliver solutions where it will be difficult to understand what is specific to the virtual and physical world. This convergence guide to the development and implementation of the CPS oriented to the manufacturing sector (CPPS). This evolution will guide the development of cyber-physical components capable and responsible for managing the virtual and the physical world, instead of having a physical world responsible for dealing with the physical actions and the virtual world responsible for dealing with information.

Hence, with the abstraction of the physical components as cyber-physical entities, the functionality, usually available on the virtual world, are embedded in the cyber part of the entities, and each cyber-physical component is responsible for taking care of part of that virtual functionality.

This characteristic allows the cyber entities to perform logical activities and functionalities, such as analysis of trends, diagnosis, error recovery, and so forth and directly apply the results in the physical world. The distributed nature of the cyber representations (which mimics the physical system) forces the usage of CS techniques capable of dealing with the distributed nature of the CPPS.

Hence, the usage of CPPS aims the development of systems highly distributed and capable of dealing with disturbances. The new CPPS approaches are based on a horizontal integration instead of the well-known vertical, rigid and layered ISA 95 pyramid (Scholten, 2007), as presented in Figure 14.

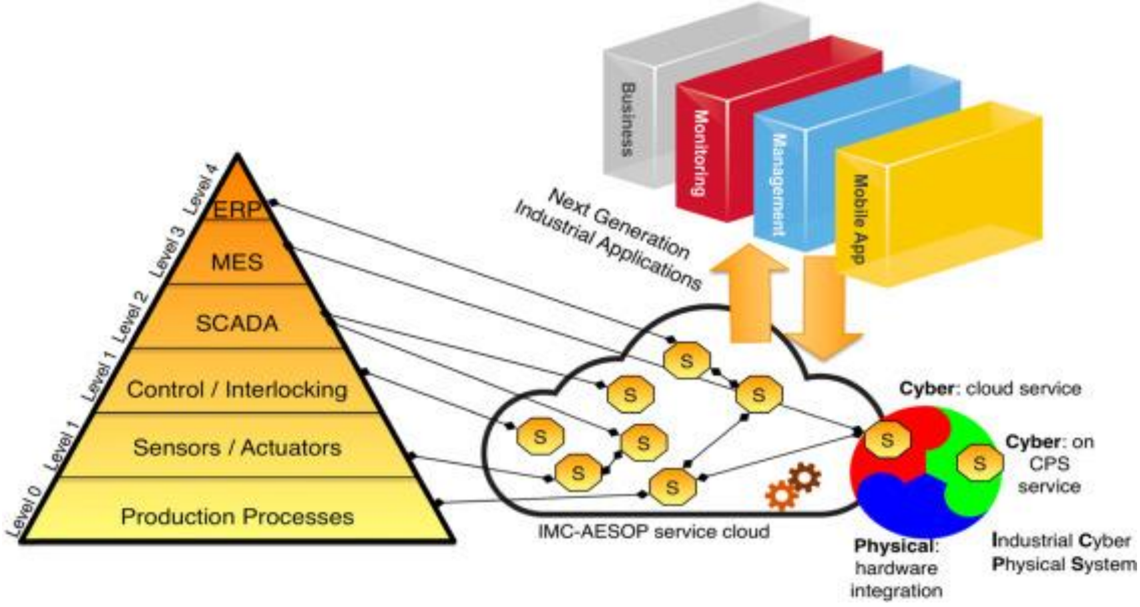


Figure 14 - Horizontal integration of CPPS, from (Leitão et al., 2016b)

With the distributed approach proposed by the CPPS, the capabilities, usually inserted in one of the different system's layers, such as ERP or MES, are now automatically generated and based on the interaction of the different cyber-physical entities available within the eco-system. More than that, the system will be able to solve some problems locally instead of propagating the problem to undesired layers, which usually increases the impact of those problems. The horizontal integration allows an easy integration of new components and functionalities in runtime, without requiring a new reformulation of specific layer and interfaces. Since the information is dynamically shared among the cyber entities, whenever a disturbance is detected, such as the addition or removal of components or functionalities, the system is able to adapt in runtime.

With these functionalities, and from an integration perspective, the CPPS implies an important role. Due to the reduction of the time required and the complexity required to integrate new components or software tools.

Since each resource is represented as a coupled cyber-physical entity, the communication with the further system's elements are made through the homogenised cyber level (Hu et al., 2016). The integration of new components is made through the definition of the coupled module, with the physical and cyber-entity.

The CPPS not only aims to integrate hardware and software but also gives, to the system level, the capability to intelligently manage the information in order to improve and optimise the system as a whole (Lee et al., 2015). The intelligent integration among the shop floor resources, using a CPPS, allows the integration of components from different vendors and capable of performing different capabilities. The CPPS also allows the integration with higher-level services, some of them hosted at the cloud level. The integration with the cloud permits the system to interconnect with external services, such as smart grids, smart transportation, among other services. In Figure 15 it is possible to verify that the CPPS is responsible for abstracting, control and optimise the production line as well as to communicate with external services. The communication with the exterior will allow the previously presented horizontal integration.

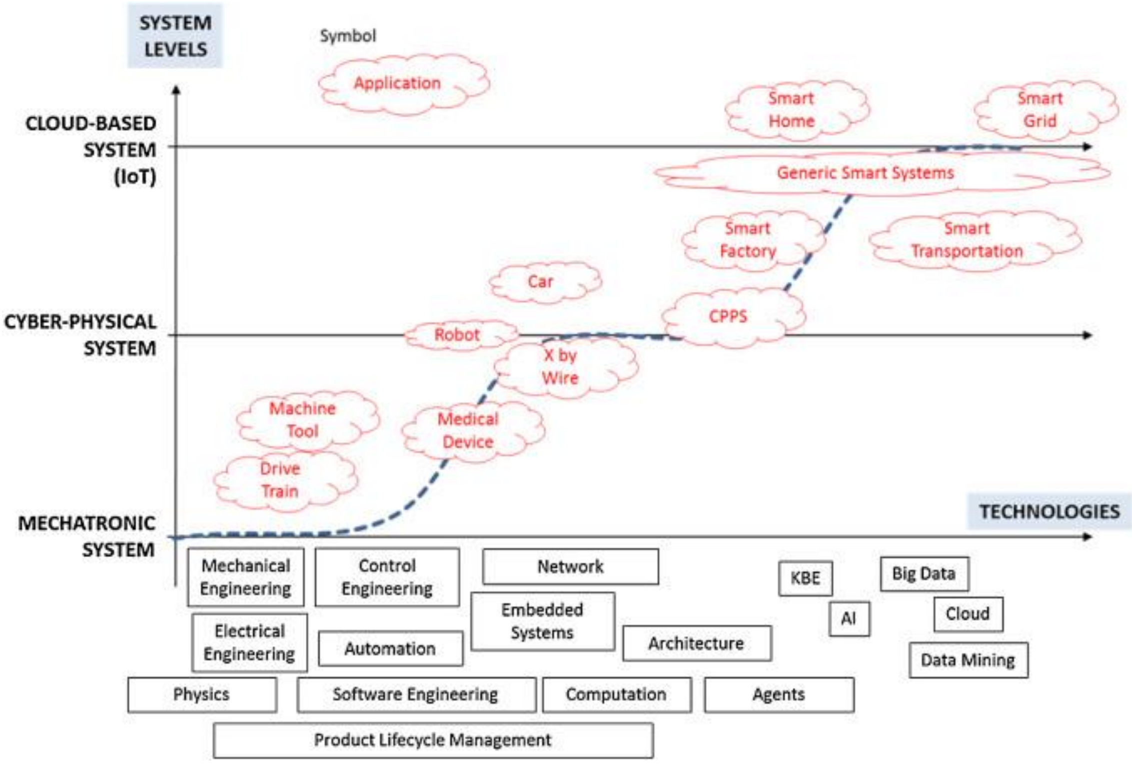


Figure 15 - Transition process from Mechatronics to CPS to Internet of Things. (Hehenberger et al., 2016)

Since one of the primary objectives of the CPPS is to abstract and interoperate different actors in a very heterogeneous environment, it is useful to have a common view of the design of the cyber-physical components.

The Reference Architecture Model Industrie 4.0 (RAMI 4.0) (Adolphs et al., 2015), proposed by the Plattform Industrie 4.0 group and depicted in Figure 16, presents a structured manner showing how to approach the development of these new distributed systems.

The main idea of RAMI 4.0 is to propose a common view with the life cycle and the value stream in a hierarchical approach for the definition of I4.0 components. Hence, the

RAMI 4.0 can be defined as a reference architecture model for the design and implementation of cyber-physical production components.

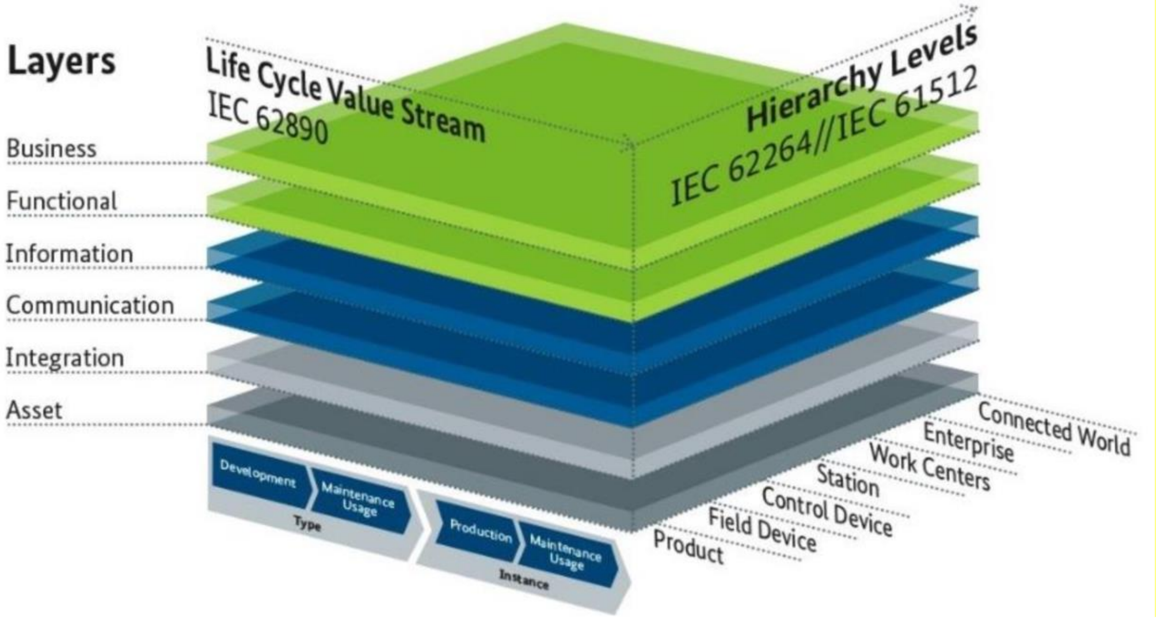


Figure 16 - Reference architecture model for Industrie 4.0 (RAMI4.0)

Therefore, the RAMI 4.0 contains three axes, as presented in Figure 16:

- Hierarchy levels: Based on IEC 62264 and IEC 61512, the horizontal axis aims to present a unified representation and view for all the components which can be present in an industrial environment (Product, Field Device, Control Devices, and so forth). This common view increases the interoperability among the actors, and it allows the creation of the interconnected world.

- Life cycle and value stream: This layer aims to represent the life cycle of the different components, such as physical resources, products or the facility itself. This lifecycle management is based on the IEC 62890, a standard focused on the lifecycle management.

- Layers: The layer represents the evolution in the vertical axis. The objective of the definition of these layers is to define different levels of abstraction. For instance, the level of abstraction is higher in the top layer (Business). Hence, the virtualisation of the physical components is made from the bottom (Asset layer) to the top of this axis.

In the last few years, some projects and research activities pushed the use of the CPPS approaches. These distributed approaches aim to deal with the need to have dynamism and flexibility, allowing the addition and removal of production resources at runtime. However, from the control and reconfiguration point of view, it is still difficult to manage the complexity verified during the design and implementation of modular solutions. The CPPS appears as a solution to manage all the complexity and to facilitate the pluggability of components during production. In (Marseu et al., 2016) a modular approach to performing plug and produce is presented.

Hence, the CPPS must be capable of dealing with disruptive and drastic disturbances. To do so, it is necessary to design and develop resilient systems, capable to not only, survive to the unexpected changes but most important than that, learn and improve its behaviour based on previous experiences. According to (Galaske and Anderl, 2016) a robust CPPS needs to fulfil the following requirements:

- "*Robustness and adaptability*": The system must be able to deal and adapt to external and unexpected influences and disturbances;
- *"Self-regulation and self-recovery"*: A resilient CPPS must be capable of regulating the production execution, and if an external event changes this regulation, the system has to recover the ideal state again, now with the new existent conditions;
- "Short response time": The system must be as fast as possible to recover from disturbances and undesired states. In this sense, reduce as much as possible the time in the disturbed mode;
- *"Intelligent component"*: Each production component or module must have a description of its capabilities, detailing which functionalities can be offered and performed by this module. When launched, the module must announce its capabilities to the system level, sharing its capabilities and availability;
- *"Autonomous decision"*: Every intelligent entity is responsible for verifying the state of the abstracted component (physical level) and the decisions regarding the execution must be taken and coordinated (cyber level) by the entity that abstracts the component itself;
- *"Redundancy"*: To deal with disturbances and unexpected problems, a resilient CPPS must have redundant modules. Whenever a problem is detected the system must be able to dynamically divert the production from one station to another, or

from one station to a cluster of stations that together could provide the same functionality;

- *"Dynamic disruption database"*: A resilient CPPS must have a database with a list of known disruptive scenarios and possible strategies to deal with the unexpected cases. This database must be updated and improved at runtime with the new learnt disruptions and solutions;
- *"Escalation scenario"*: The usage of a simulation environment, to simulate several and different scenarios, must be used to improve and enrich the overall knowledge of the CPPS.

Since the CPPS remains in an early stage of research and development, this kind of systems faces several challenges and difficulties, such as real-time processing constraints, reliability, security and privacy, among others (Yue et al., 2015).

In (Leitão et al., 2016b), the authors present not only a list of the CPPSs' key challenges but also, an evaluation of each key challenge, regarding the difficulty to handle it, the priority for the developments and an expected time to solve and mature the solution.

Table 2 - Key challenges in industrial CPS (Leitão et al., 2016b)

Area	Key Challenges	Difficulty	Priority	Maturity in
CPS Capabilities	Real-time control of CPS systems	High	High	4–7 years
	Real-time CPS SoS	High	Medium	3–5 years
	Optimization in CPS and their application	High	Medium	4–7 years
	On-CPS advanced analytics	Medium	High	3–5 years
	Modularization and servification of CPS	low	High	3–5 years
	Energy efficient CPS	Medium	Medium	3–5 years
CPS Management	Lifecycle management of CPS	Medium	Medium	5–8 years
	Management of (very) large scale CPS and CPS-SoS	High	High	5–8 years
	Security and trust management for heterogeneous CPS	High	High	5–8 years
CPS Engineering	Safe programming and validation of CPS SoS	High	High	5–10+ years
	Resilient risk-mitigating CPS	High	High	5–10+ years
	Methods and tools for CPS lifecycle support	High	High	3–7 years
	New operating systems and programming languages for CPS and CPS SoS	Medium	Low	3–6 years
	Simulation of CPS and of CPS-SoS	Medium	High	3–6 years
CPS Infrastructures	Interoperable CPS services	Medium	High	2–5 years
	Migration solutions to emerging CPS infrastructures	Medium	High	3–6 years
	Integration of heterogeneous/mobile hardware and software technologies in CPS	Low	Medium	2–4 years

Area	Key Challenges	Diffi-culty	Priority	Maturity in
CPS Eco-systems	Provision of ubiquitous CPS services	Medium	Medium	3–5 years
	Economic impact of CPS Infrastructure	High	High	3–6 years
	Autonomic and self-* CPS	High	Medium	7–10+ years
	Emergent behaviour of CPS	High	Medium	7–10+ years
	CPS with humans in the loop	High	High	2–5 years
	Collaborative CPS	Medium	Medium	5–8 years
CPS Information Systems	Artificial intelligence in CPS	High	High	7–10+ years
	Cross-domain large-scale information integration to CPS infrastructures	Medium	Low	6–9 years
	Transformation of CPS data and information analytics to actionable knowledge	High	High	4–8 years
	Knowledge-driven decision making/management	High	Medium	6–10+ years

Due to the immense amount of challenges still needed to be addressed. It is difficult for these developments and approaches to enter in the manufacturing sector.

However, in order to narrow the gap between the research developments and the industrial applications, some learning factories are being developed. These learning factories aim to train the technical personnel to get in touch with the last trends in the manufacturing sector (Thiede et al., 2016), such as CPPS. The usage of these learning factories will diminish the risk of employing this type of systems. These learning factories can be used not only to show the developments but also, to train the operators before start working in a real factory with CPPS based production lines.

Another important issue regards the costs. The companies, in general, are not able to make a significant financial effort, to renew the entire production resources in order to have a CPPS line (Babiceanu and Seker, 2016). Hence, it is vital to push the development of hybrid solutions. These hybrid solutions must link the existent production resources, such as the existent control mechanisms, manufacturing execution systems, etc., with the new CPPS based solutions. Hence, it is possible to implement intelligent and flexible functionalities and step-by-step penetrate the sector and to give the companies some new functionalities without significant efforts.

2.4. Industrial Agents

As stated previously, the use of distributed and cooperative approaches is seen as an effective method to develop manufacturing systems highly flexible, reconfigurable and capable of dealing with external disturbances.

Two schematics, one with the conventional centralised decision making and one on the right with the cooperative approach, are shown in Figure 17. In a conventional system, the master-slave approach is used to coordinate the execution of the different components. This implies that the master knows all the linked slaves and all the potential linked slaves in order to know how to coordinate the execution. The main problems with this approach remain on the incapacity to run the system without the master, and if for some reason the master was not available, the system would not be able to run. Another possible problem is the complexity required to coordinate all the slaves, which could decrease the system's performance accordingly to the executed tasks' complexity that need to be handled by the master.

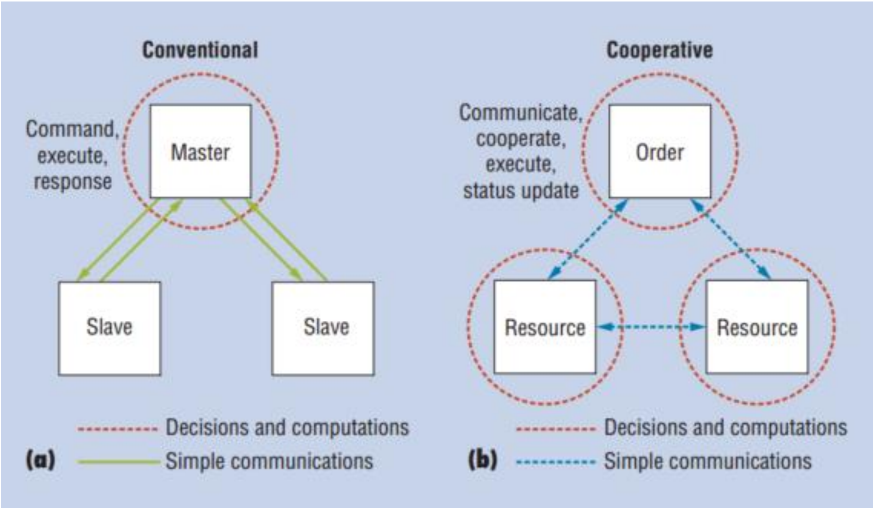


Figure 17 - Decision making: (a) conventional versus (b) cooperative (Marik and McFarlane, 2005)

Regarding the cooperative approach, on the right, all the entities are responsible for solving problems and actively contribute to the problems' resolution. The system will try to find solutions for the existent problems, combining the existent entities without rigid rules and entities, this allows them to find possible solutions without bottlenecks and critical points.

The cooperative approach usually is implemented through the development of intelligent agents, responsible for representing and abstracting each node.

2.4.1.1. Agent and Multi-Agent Systems

The definition of agent is not consensual, and several definitions can be found in the literature. However, some common visions and general concepts can be found. Each agent is designed and implemented to run in a specific environment, as presented in Figure 18.

Figure 18 - The agent and its environment (Monostori et al., 2006)

Nevertheless, the agent is inserted in a specific environment, and usually, each agent only knows the environment partially, because each agent is responsible for some specific tasks. In this case, each agent abstracts a physical resource working on the shop-floor. Hence, the agent knows the sensors' values extracted from the abstracted resource. A critical characteristic of the agents is the capacity to keep their one state along the time, allowing the agent to evolve. This state is updated and created through the observation of the environment, goals and objectives for the execution and the previously generated knowledge. Based on these inputs, the agent decides what the best plan to achieve the goals and acts is. The process runs iteratively during the entire agent life cycle.

Although all the needed characteristics for an intelligent agent were not consensual, some general characteristics and capabilities can be seen in almost all definitions (Camarinha-Matos and Vieira, 1999), presented in Figure 19.

Figure 19 - Main Characteristics of an Intelligent Agent

- *“Autonomy”*: The agent must be able to perceive the environment and, based on its desires, act without the need to interface or interact with third parties. Comparing to the human case, each human is autonomous because even talking and receiving external information, the human itself takes the actions and decisions and not from external entities.
- *“Sociability”*: Since the agent is inserted in an ecosystem, as presented in Figure 20, populated for other agents and each agent only knows the environment partly, the agent must be able to communicate and share some knowledge with the other agents. This ability allows the coordination of efforts in order to achieve the common goals and to share relevant information, not directly accessible from the environment.
- *“Rationally”*: Based on the information received by an agent, from the environment, goals, other agents, its own objectives and generated knowledge, the agent must be capable of reasoning all this information and define which are the best actions to apply.
- *“Reactivity”*: In some cases, an agent needs to react to some changes verified by the environment. Some actions, due to real-time constraints, must be instantly handled in order to avoid more significant problems and propagations.
- *“Proactivity”*: The execution and actions of an agent are triggered by itself and not by external entities. Based on that an agent must be able to check the environment and internal conditions regularly and, if necessary, triggers the needed actions to achieve its goals.
- *“Adaptability”*: During execution, and based on the capability to reasoning and deal with different situations and problems, an agent must be able to learn and adapt its behaviour if in some moment it learnt that changing some characteristics and actions would be better to achieve the goals and objectives.

An ecosystem of collaborative and cooperative agents constitutes a MAS, as presented in Figure 20.

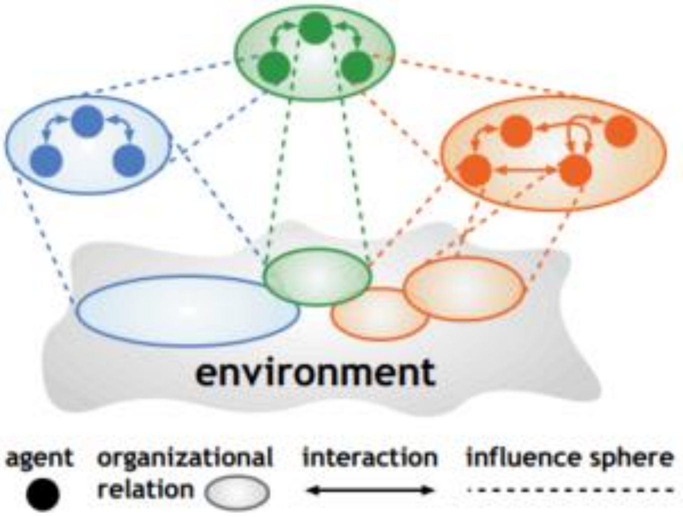


Figure 20 - Generic scheme of a multi-agent system (Monostori et al., 2006)

With the constant evolution of the ICT such as network bandwidth and powerful computers and controllers, MAS are getting more common, and they are being pointed as an enabler to disruptive manufacturing environments' approaches.

The agent-based solutions have been proposed and studied in the last two decades due to the agents' characteristics and potential. Regarding the industrial sector, the agent-based solutions have been proposed to optimise and solve problems regarding three main areas (Marik and McFarlane, 2005):

- Real-time manufacturing control: In this area, the agents can be used to perform several tasks, related to the control logic at the shop floor level. In here, several solutions can be designed to solve different problems. The agents can be used to directly control the production line (Ribeiro et al., 2011), reconfigure the control logic interfacing with standard technology (Rocha et al., 2014), monitoring the execution (Sauer, 2004), among others. In this low level, the time constraints are very critical, the time to react is very reduced, and the systems need to be very reactive. Traditionally the used shop floor's functionalities are extremely rigid and centralised, but with the usage of agents, the systems can be decentralised and capable of evolving in order to cope with unexpected events.
- Complex operations management: At this intermediate level the agents are used to coordinate and optimise scheduling tasks (Leitão and Restivo, 2008), and production planning (Pechoucek et al., 2002). The usage of agents at this level is common, and the previously stated real-time constraints are no longer critical and required at this level. The capability to somehow find unexpected solutions through

the emergent behaviours is one of the leading advantages of using these solutions since the systems have time to compute complex algorithms without relative impact at the shop floor.

- Virtual enterprises' coordination: At the highest level, the coordination and collaboration among different enterprises are getting a considerable challenge, not only for the humans but also for computational solutions. Through simulation and self-organised solutions, the agents can be used to find effective collaborations and optimise the creation of those collaborations. For instance, using CPPS, one of the main required integrations is between the shop floor environment and the different suppliers in order to optimise the different resources, not only internally but from a global perspective.

The primary focus of this work regards the use of agents to control, configure and optimise the shop-floor. Concerning these applications, several approaches have been proposed. Some of the production paradigms proposed of dealing with the new requirements and delivering flexible and adaptable production systems used approaches based on MAS. The Bionic Manufacturing Systems (Ueda, 1992; Ueda et al., 1997), Holonic manufacturing Systems (Van Brussel et al., 1998) and Evolvable Production Systems (Onori et al., 2006) are usually implemented using MAS based solutions due to the paradigms' nature. These disruptive paradigms aim to deliver highly distributed control and reconfiguration approaches capable of dealing with disturbances rapidly. Since these approaches implement self-organized and self-adapted solutions, based on AI algorithms, they are capable of dealing with rapid changes of production orders, the addition or removal of physical components and capabilities in runtime, changes on the availability of the resources, among other functionalities.

2.4.1.2. Agent-based Cyber-Physical Production Systems

Since the beginning, the agent-based approaches have been presented as potential solutions to allow the implementation of distributed CPPS. Hence, most of the presented ones are based on MAS due to the agents' characteristics and benefits (Mařík and Lažanský, 2007; Marik and McFarlane, 2005) of using agents for a manufacturing environment:

- <u>Feasibility</u>: In some cases, due to the high complexity of the problems to solve, agents can be the only feasible solution. With the increase of systems' complexity due to the intensification of devices and data sources, the complexity of the solutions also increases dramatically, and the agents can be the only or the adequate solution;

- Robustness: One of the main required characteristics, for shop floor systems, is the capacity to react and adapt to external disturbances. In this sense, the agents can dynamically adapt and face those disturbances. Capabilities as self-organisation, self-learning, self-awareness, among others, allow the agents to perceive the changes and in runtime find solutions based on the existent conditions. Hence, every time a component is down, in maintenance or a new product needs to be produced, The overall system can offer a solution for that. Another relevant advantage is the distributed nature of the system which removes the central point of failure, reducing significantly the possibility to have stoppages and unrecovered problems;

- Reconfigurability through pluggability: The new market trends force the companies to react and adapt very quick to the shop floor. Usually, the production systems' changes are the most critical and difficult to operate. The agent-based approaches allow the addition and removal of modules at runtime or with small stoppages. Through the design of independent entities or systems of entities (cyber-physical entities) capable of interacting and smoothly interoperate with the existent components, the system can reconfigure itself and operate with the existent and plugged components;

- Redeploy ability: The agent-based approaches can be deployed in different fields and different environments with minor changes and adjustments. Usually, the different solutions use generic entities abstracting components and tasks usually found on the shop floor. Hence, the same design can be instantiated and deployed for different proposes;

- Embedded control and diagnosis: As previously presented, the abstraction of the components at the shop floor with cyber entities is possible through the implementation of the cyber entities as agents, inserted in the cyberspace (MAS). The distributed entities, running locally at the device level, are capable of monitoring, diagnose and control *in loco* the respective component, without requiring a central unit to compute all the extracted data. Local error recovering and monitoring will reduce the possibility of error propagation and almost instantly discover the problems' cause;

- SoS approach: The concept and functionalities of the agents can be used in different levels of granularity, creating the concept SoS. For instance, to reduce the complexity of the system and at the same time increase its scalability, each station can be defined as a group of physical components working together and capable of offering a complex task, combining different resources. In a different level, each factory can be seen as a virtual organisation capable of offering production services for different customers, and it consumes raw materials from different suppliers, without sharing what is made internally.

- Interoperability: The manufacturing environment is by nature very complex and heterogeneous, with components and machinery from different vendors and with different characteristics, such as standards and communication protocols. The usage of agents to abstract the different components and functionalities at the different levels allows the interoperability of the diverse components and areas without translations and complications. Hence, all the components are interoperable and capable of communicating among them;

- Integration: Making usage of the previous characteristics, the capability to locally control and monitoring the module, the unique required integration take place at the device level and for each module. For instance, whenever a new component/station is added, the agent abstraction will be responsible to interface with the plugged hardware (local integration) and communicate with the other agents (cyber entities), but since the communication among them is standardised and known the integration among the different components is performed autonomously.

However, since the agent approaches for CPPS are entirely new, some challenges and difficulties still being found:

- Cost: The implementation of agent-based distributed solutions is more expensive than a traditional centralised approach. Another critical issue for most of the companies regards the difficulty in migrating the traditional working approaches to a new and built from zero solution;

- Unexpected behaviours: Due to the decentralised and self-organised behaviour of the system, it is difficult to define it's exact behaviour. Usually, this characteristic is not well accepted, mainly by the management team, due to the difficulty to predict the profits, the production performance and so on;

- Performance: Today's implementations and solutions have significant problems regarding scalability. The number of agents required to run on a specific system can influence the overall performance of the system, and this creates a problem for agents' usage;

- Training and knowledge: The traditional system's integrators and engineers are not used to develop and instantiate this type of solutions. They were trained and formed to design and implement centralised and rigid control systems without emergent behaviours and distributed control systems. The knowledge concerning these new approaches needs to be transferred, and several training activities need to take place;

- Standards: Standards and interoperability activities need to take place in order to define a standard way to integrate and communicate among the different agent-based implementations. This common approach will increase the usage of agents through the fast and easy interoperability of different solutions.

Since the 90s, several research activities and industrial applications of agents in several factories and demonstrators took place. A list of industrial agents' applications is compiled in Table 3. The table presents the applications, the year of its presentation, the industrial sector, the scope of the proposed solution (control, scheduling, planning, among others) and also an estimated Technology Readiness Level (TRL) (Mankins, 1995) of each proposed solution.

Table 3 - Application of industrial agents for manufacturing proposes

Application	Scope	Industry Type	Year	TRL
BHP Billiton (Mařík et al., 2005)	Process control	Steel milling	1995	TRL 7-9
Yokogawa (Wada et al., 1998)	Manufacturing control	Not applicable	1998	TRL 4-6
PROSA (Van Brussel et al., 1998)	Manufacturing control	Not applicable	1998	TRL 1-3
MASCADA (Daimler-Benz pilot) (Bruckner et al., 1998)	Manufacturing control	Automotive	1999	TRL 7-9
Daimler Chrysler (Bussmann and Schild, 2001)	Manufacturing control	Automotive	2001	TRL 7-9
AARIA (Van Dyke Parunak et al., 2001)	Scheduling	Army manufacturing	2001	TRL 4-6
LIAZ (Pechoucek et al., 2002)	Planning	Automotive	2002	TRL 7-9
Skoda (Pechou-cek et al., 2002)	Planning	Automotive	2002	TRL 7-9
CoBASA (de Oli-veira, 2003)	Manufacturing control	Not applicable	2003	TRL 1-3
Cambridge pack-ing cell (Fletcher et al., 2003)	Manufacturing control	Not applicable	2003	TRL 4-6
Steel rooling (Cowling et al., 2003)	Scheduling	Steel production	2003	TRL 7-9
MAST (Vrba, 2003)	Simulation	Not applicable	2003	TRL 1-3
Watchdog Agent (Djurdjanovic et al., 2003)	Monitoring and di-agnosis	Automotive	2003	TRL 7-9
Holonic diagno-sis (Jarvis and Jarvis, 2003)	Diagnosis	Automotive	2003	TRL 7-9
FABMAS (Mönch et al., 2003)	Process control	Semiconductor wafer fab-rication	2003	TRL 7-9
PABADIS (Lüder et al., 2004)	Manufacturing control	Not applicable	2004	TRL 4-6
ProVis (Sauer, 2004)	Monitoring	Automotive	2004	TRL 7-9
ABAS (Lastra et al., 2005)	Simulation and control	Not applicable	2005	TRL 4-6
Saarstahl (Jacobi et al., 2005)	Planning and monitoring	Steel production	2005	TRL 7-9
NovaFlex (Cân-dido and Barata, 2007)	Manufacturing control	Not applicable	2007	TRL 4-6
AGP (Blanc et al., 2008)	Scheduling	Glass production	2008	TRL 7-9

Application	Scope	Industry Type	Year	TRL
ADACOR-FMS (Leitão and Restivo, 2008)	Manufacturing control	Not applicable	2008	TRL 4-6
Axion-Holding (Andreev et al., 2010)	Scheduling	Electronics production	2010	TRL 7-9
Kuznetsov (Shpilevoy et al., 2013)	Scheduling	Aircraft jet engines production	2013	TRL 7-9
GRACE (Whirlpool pilot) (Leitão et al., 2015)	Reconfiguration and quality control	Home appliances production	2013	TRL 7-9
IDEAS (Ribeiro et al., 2015, 2011)	Manufacturing control and diagnosis	Not applicable	2013	TRL 4-6
ORCA-FMS (Pach et al., 2014)	Scheduling	Not applicable	2014	TRL 1-3
ADACOR2 (Barbosa et al., 2015)	Manufacturing control	Not applicable	2015	TRL 4-6
PRIME (Rocha et al., 2014, 2015)	Reconfiguration and monitoring	Not applicable	2015	TRL 4-6
ADMARMS (Farid and Ribeiro, 2015)	Manufacturing control	Not applicable	2015	TRL 1-3
BIOSOARM (Dias-Ferreira et al., 2016)	Manufacturing control	Not applicable	2016	TRL 1-3

Regarding process control, BHP Billiton (Mařík et al., 2005) has been designed and instantiated to control the process for steel milling. On the other hand, FABMAS (Mönch et al., 2003) was deployed to solve the same issue regarding fabrication of semiconductors.

In order to optimise the production planning, in (Pechoucek et al., 2002) a solution to plan the production of patterns and forms to be used in automotive industry is proposed. In (Pechoucek et al., 2002) is also presented an agent-based solution to plan the production of engines for the same industrial sector.

Saarstahl, an integrated solution capable of optimising the production planning but also monitoring the solution is presented in (Jacobi et al., 2005). This solution was designed and deployed for planning and monitoring steel production. ProVIs (Sauer, 2004) presented an entirely focused solution to perform monitoring in the automotive industry. However, in (Djurdjanovic et al., 2003) an agent-based toolbox is described, this tool is capable of extracting the data and processing it. Based on that processing the system performs machinery diagnosis and try to find components degradation.

The diagnosis is a recurrent problem in several industrial sectors. A holonic based diagnosis approach to verify the correct assembly in an automotive industry assembly line is presented in (Jarvis and Jarvis, 2003). The FP7 IDEAS projects developed a distributed control system with embedded diagnosis functionality. This approach has been demonstrated in two industrial demonstrators, one designed by Festo (Ribeiro et al., 2011) and the other one has been developed by Masmec, focused on conveyor belts based transport systems (Rocha, 2013).

Regarding machinery control, Yokogawa (Wada et al., 1998) presents a distributed agent-based approach to control physical resources. The solution has been tested at the laboratory of the Robotics Institute at Carnegie Mellon University.

Concerning the automotive industry and the design of agent-based solutions for manufacturing control. In (Bruckner et al., 1998) the EU MASCADA project developed an industrial pilot for a Daimler-Benz line. The developed pilot was used to control the painting centre partially. In (Bussmann and Schild, 2001) a pilot developed by Schneider Automation was used to control the production of engine cylinder heads, in a Daimler Chrysler factory. The FP5 PABADIS (Lüder et al., 2004) developed a MAS environment to control the shop floor also. Moreover, the Cambridge packaging cell (Fletcher et al., 2003), NovaFlex (Cândido and Barata, 2007), ADACOR-FMS (Leitão and Restivo, 2008) and ADACOR2 (Barbosa et al., 2015) presented four distinct control solutions using demonstrations in universities to demonstrate the designed solutions. ABAS (Lastra et al., 2005), developed at Tampere University Technology, uses 3D simulation and visualisation to help on the shop floor control. The manufacturing control is one of the most active topics from a research point of view. Some architectures and solutions have been proposed by the academia but tested only in simulation environments, such as the PROSA (Van Brussel et al., 1998), CoBASA (de Oliveira, 2003), MAST (Vrba, 2003), ADMARMS (Farid and Ribeiro, 2015) and BIOSOARM (Dias-Ferreira et al., 2016) cases.

The application of industrial agents to solve the scheduling problem is also commonly used, and it is applied in different sectors. AARIA (Van Dyke Parunak et al., 2001) was designed to schedule army manufacturing. In (Cowling et al., 2003) an agent-based solution is responsible for scheduling the production of steel rolling, and for the (Blanc et al., 2008) case focuses the glass production. In Axion-Holding (Andreev et al., 2010) and Kuznetsov (Shpilevoy et al., 2013), two solutions have been designed to schedule the production of electronics and aircraft engines respectively, more recently ORCA-FMS (Pach et al., 2014) has been proposed but not deployed so far.

FP7 GRACE (Whirlpool pilot) (Leitão et al., 2015) presented an approach, using industrial agents, to verify the quality of the product along the line and dynamically adapt and reconfigure the production parameters. The reconfiguration aims to improve products' quality or solve already detected problems.

The FP7 PRIME presents the usage of two agents based solutions. One of them to perform reconfiguration of standard and legacy production systems, this functionality was demonstrated at the University of Nottingham (Antzoulatos et al., 2014). The other is capable of extracting and pre-processing the extracted information to perform optimised and flexible monitoring, demonstrated in two industrial cells from automotive industry (Rocha et al., 2015).

A summary of the industrial agents' applications scopes, based on the previously presented list, is shown in Figure 21. It is possible to verify that several different functionalities can be offered for the production environment (diagnosis, monitoring, scheduling or planning). However, the main focus is on the control problematic. This trend can be understood as a general push to develop new emergent control environments capable of dealing with the, previously presented, new market variations and disturbances. Although the market is, of course, pushing in this direction, it is important to refer that the traditional tools, such as monitoring, or diagnosis are very obsolete and incapable of dealing with the verified distributed and volatile shop floors. More than that, the emergent CS and ICT technologies and techniques can be used to develop more effective and useful functionalities and not only to solve the control problem.

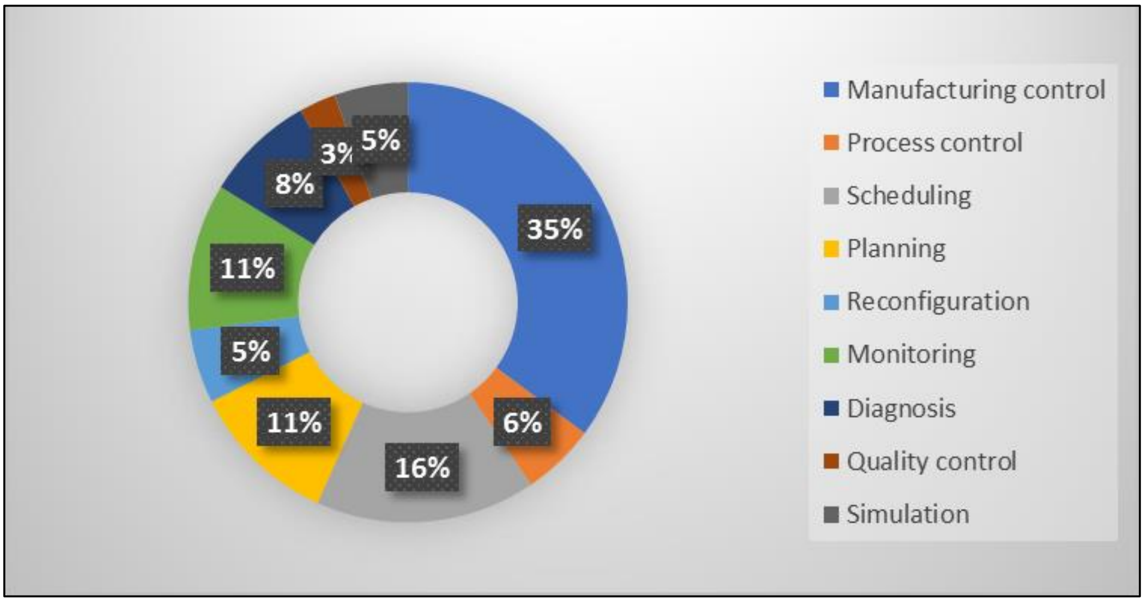


Figure 21 - Distribution of the industrial agents' applications scope

Although, to use agents in industrial environments, in the last decade's many proposals have been presented, in fact, most of the proposals do not use common guidelines and only recently, some work has been presented to define good practices in the design and implementation of agent-based CPPS solutions. A representation of the different layers of design of an industrial CPS is presented in Figure 22.

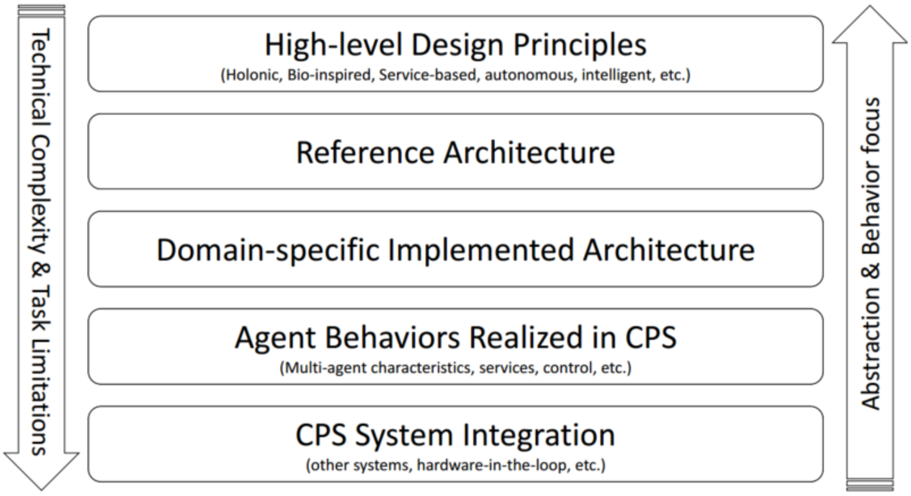


Figure 22 - Design principles for agent-based CPS (Leitao et al., 2016)

This representation is composed by five different levels of abstraction, from the High-Level Design Principles (Holonic based design, Bio-inspired, and so forth) until the lowest level of abstraction where the CPS is integrated with the hardware itself.

The representation aims to specify the different levels of abstraction, that posteriorly will result in the different stages of the design of a CPPS. The High-Level Design Principles focuses in the definition of generic behaviours and characteristics of the system such as the production paradigm. The lower stage encompasses the definition of a reference architecture. In this stage a generic architecture is defined. In the end of this stage, all the generic work is done, and the design of case specific solutions start. The Domain-specific Implemented Architecture stage aims to define a customisation of the Reference Architecture for a specific case. Hence, in the end of this stage, the architecture is ready to be implemented, and in the lower stage the behaviours of the agents is specified. The lowest stage focuses the integration of the designed CPS with the hardware or simulation tools.

This design-oriented work is particularly relevant to guide the engineers and developers during the design of CPPS, and for the systems' integrators to understand what is necessary to do. Based on the presented methodology, the systems integrators' effort is only required on the lowest level (CPS System Integration).

2.5. Summary

This chapter presented a literature review, focused on the design of CPPS using agent technology. Due to the agents' characteristics, this technology can be beneficial to implement CPPS capable of dealing with nowadays' requirements, an I4.0 oriented CPPS.

In this sense, the concept of CPPS was presented, and the main functionalities were described and discussed. Afterwards, an exhaustive list of industrial agents' usages was detailed. It is crucial to understand that, in the last decades, several agent-based solutions were proposed and deployed. However, the approaches still having problems to enter in the industrial sector (Leitão, 2009).

The justification for this misapplication is not unique, and several barriers and gaps were identified during the literature review (see Figure 23). However, some barriers are straight related to the inertia to change. For instance, the necessity to change hardware. However, even other barriers as advanced control logic update, costs, lack of knowledge and the usage of different standards are enormous barriers and very difficult to solve.

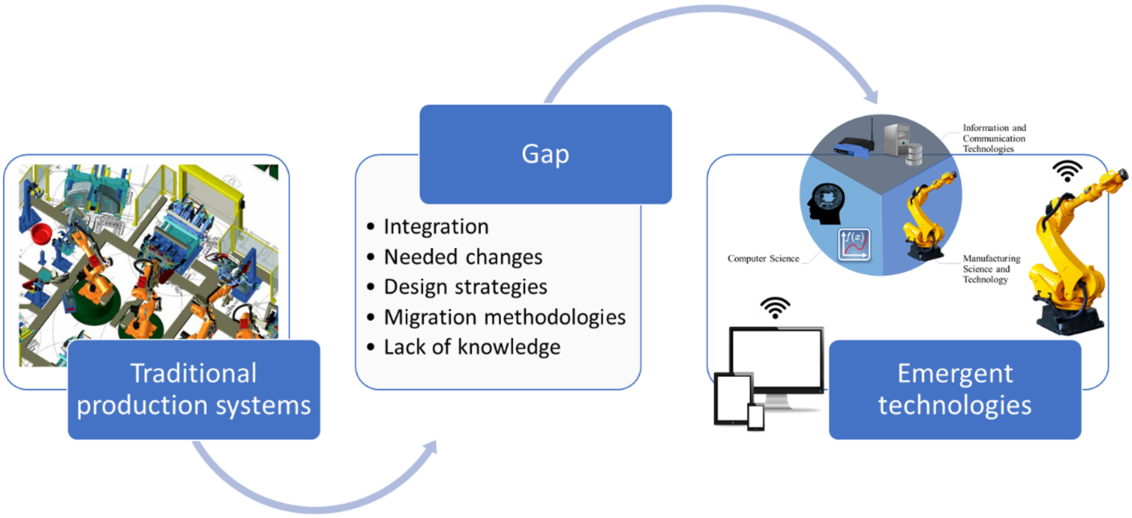


Figure 23 - Gap between the traditional and the proposed emergent manufacturing technologies

Although the trend is the migration from the traditional approaches to CPPS, the time necessary to mature the change is long, more than ten years, according to Table 2. However, the higher-level functionalities developed with this kind of solutions are very promising. Although the ideal case is the development and deployment of solutions designed from scratch, the truth is that the existent factories will continue to produce until new factories were built. Hence, in order to increase the usage of these solutions, it is crucial to develop strategies in order to guide the development of hybrid solutions were the current production system could benefit from functionalities delivered by CPPS, running upon the existent production system. Therefore, the companies could benefit from functionalities like production optimisation, reconfiguration, and so forth, in the existent facilities. The author believes that a list of design properties with clear guidelines to implement CPPS capable of integrating with the existent ones without requiring many changes will increase the adoption of these solutions in this transactional phase.

3. Functional Requirements

This chapter aims to clarify some important definitions as well as to introduce the FRs needed to develop a minimally invasive agent-based CPPS. The functionalities and the production system already running in the factory are associated with the called native system. Hence, for the purpose of this work it is important to specify which characteristics are considered part of the native system or not. The new software proposed to deliver the new functionalities is described as the cyber system. Hence, this chapter starts with a description of these new systems. Furthermore, the FRs collected for both systems in order to implement a minimally invasive systems are presented.

Accordingly to the literature review, it is known that the existent solutions are not being adopted by the industry as expected. Hence, the proposed work aims to, firstly, collect which requirements are mandatory for the companies in order to consider and benefit from the adoption of such solutions.

Before assessing the solutions and start collecting the requirements, it is relevant to define which are the objectives and the main ideas about the design of a minimally invasive agent-based CPPS.

Henceforth, to develop a minimally invasive CPPS, it is necessary to deliver a cyber system which runs upon the existent native system. That being said, the proposed cyber system must not control the existent native system, and in this sense, it is essential to define a boundary between the two systems. The boundary must be defined as an interface through which the two systems can communicate.

In Figure 24 an overview of the CPPS is presented, with the cyber and native systems. The native system encompasses the entire production components and actors that already exist in a specific factory.

The cyber system, as presented in the figure, is responsible for abstracting the existent native system, and managing the available information. The cyber system must deliver new functionalities that the native system is not capable of offering without additional software. Moreover, the cyber system must be able also to interface with external software tools. This characteristic allows the new cyber system to scale and to deliver new functionalities built on top of it.

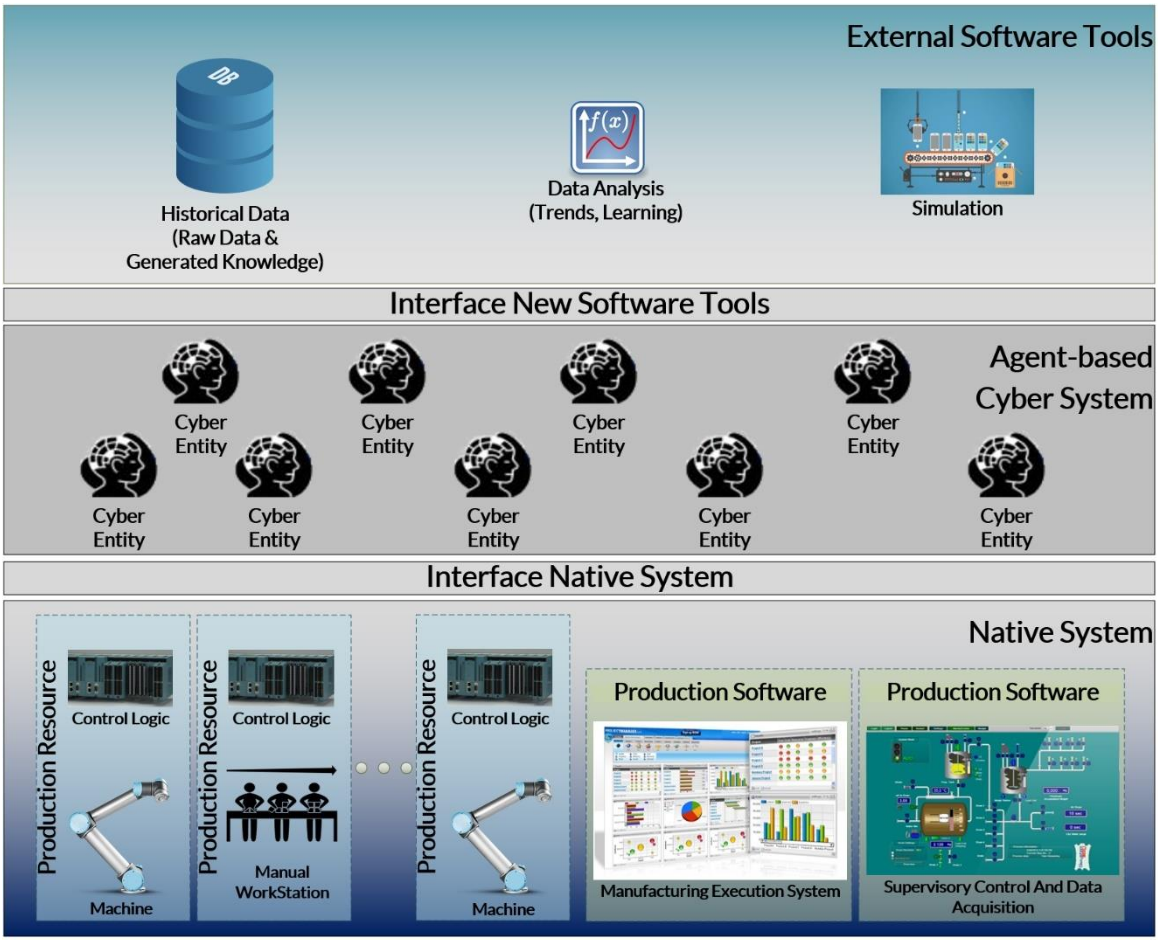


Figure 24 - Cyber and native systems

Based on the definitions of cyber and native systems, and the functionalities usually required for these new approaches, it was compiled two lists of requirements. One of the lists focuses the cyber system, describing which functionalities must be fulfilled by this system. The other list aims to compile which characteristics must be present in the native system, in order to allow the deployment of the cyber system.

The lists of requirements aim to describe which functionalities must be delivered when a minimally invasive agent-based CPPS is designed. Hence, it is crucial, for the research proposed, to introduce a definition of minimally invasive agent-based CPPS, using the previously presented concepts:

A minimally invasive agent-based CPPS is a system designed to deliver an agent-based cyber system, capable of implementing advanced capabilities, requiring minimal changes to the existent native system, such as the hardware, control logic or existent software tools.

Hence, the idea of designing a minimally invasive agent-based CPPS encompasses the development of solutions capable of delivering functionalities such as reconfiguration without stopping the line, data extraction and processing for optimisation, and so forth. That proposed solutions must be designed in order not to force the companies to change the existent native system drastically.

3.1. Functional Requirements

The collection of requirements took place in informal conversations and meetings during the developments for the FP7 IDEAS (Onori et al., 2012) and FP7 PRIME (Rocha et al., 2014) projects. As presented in chapter one, these two projects aimed to design and develop agent-based CPPS for different purposes (control, diagnosis, reconfiguration and monitoring). In these projects were possible to assess, with industrial partners from different sectors (automotive, home appliances, precision manufacturing and systems' integrators), the main challenges and difficulties. With these conversations, it was possible to verify the gaps when a company desires to replace or improve the existent native systems with the new solutions proposed, such as the agent-based CPPS.

Figure 25 presents the workflow used during the FRs elicitation.

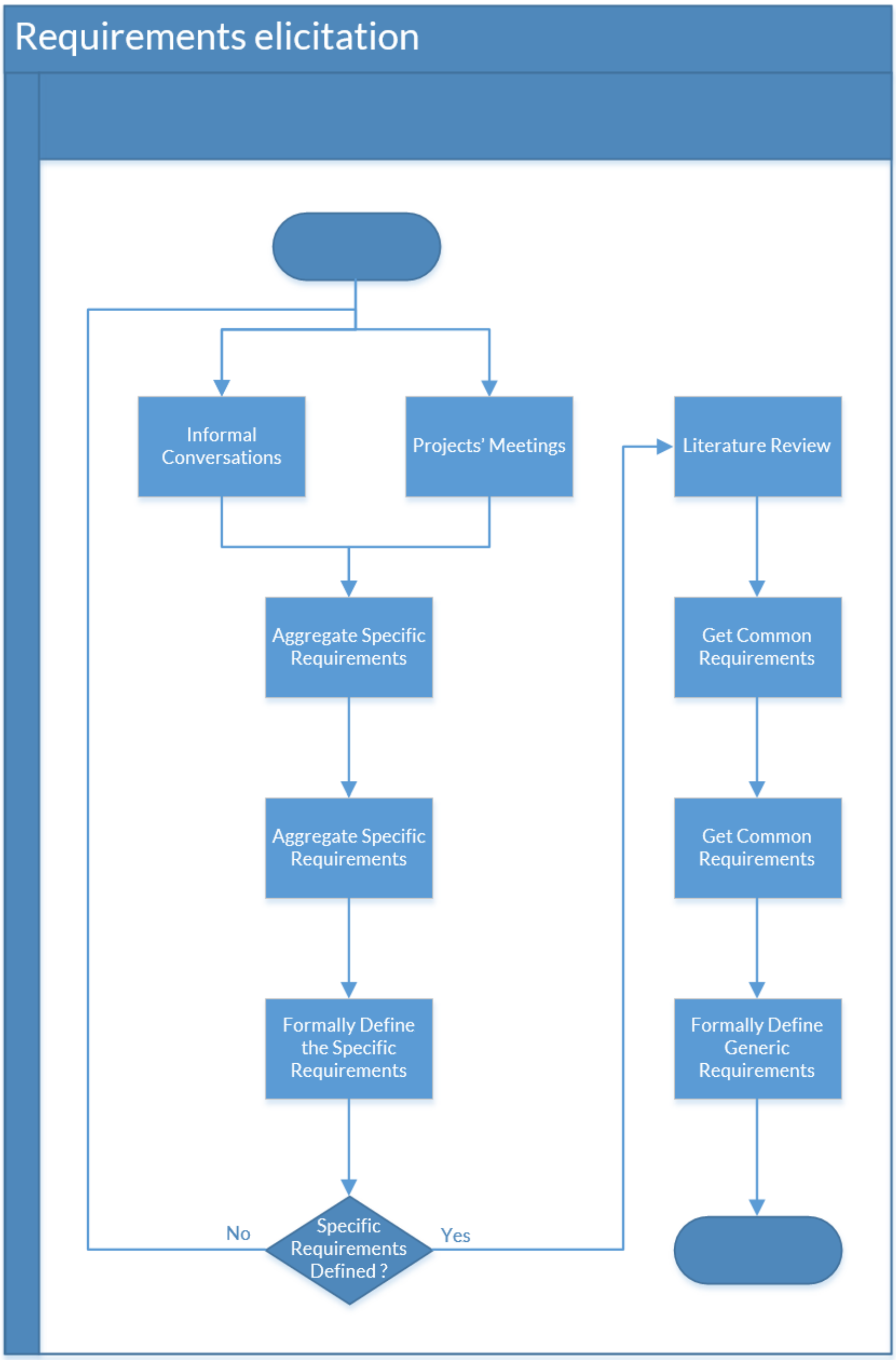


Figure 25 - Requirements elicitation workflow

The information collected during the conversations and meetings, posteriorly merged with the information collected from the literature review resulted in two lists of FRs. One list aims to describe all the requirements that must be fulfilled by the cyber system, and the other one presents the requirements that must be fulfiled by the native system. In this case, these requirements will allow the industrial companies to know if the native system fulfils the characteristics needed to receive and interface with a cyber system.

The elicitation of the FRs for both systems was performed iteratively. Firstly, it was created a first general requirement, from that requirement it was created a list of requirements that fulfil the requirement previously created and so on and so forth. The process stops when the definition of new requirements implies the description of characteristics related to the specific solution. Since, in this work, the objective is to generate requirements at the generic level only.

Although there are several requirements for both systems, the cyber and native one, a general requirement can be defined, and it presents the primary constraint and concern by the industrial partners.

Hence, the first proposed FR that represents the general and primary requirement for all the minimally invasive CPPS is:

- FR-0: *"The CPPS shall operate at least to keep its native KPIs within the desired boundaries or to improve them."*

This FR is the basis of all the requirements that were defined posteriorly, for the cyber and native system. After the definition of FR-0, it was proposed a list of FRs that describes the first level of FRs for the cyber system.

3.1.1. Functional Requirements for the Cyber System

In this section, it is presented a list of all the collected and defined requirements that must be kept in mind when designing the cyber system of a minimally invasive agent-based CPPS.

Using FR-0 as a reference, the following FRs were proposed, showing all the FRs that must be fulfilled by a cyber system, in order to fulfil the previously defined FR-0.

- FR-01 – *The cyber system shall scale up or down mirroring the needs and opportunities of the native system.*

This FR (FR-01) guarantees that whenever a physical resource is added or re-moved, the cyber system is capable of increasing or decreasing the number of physical resources abstracted. This functionality allows the companies to update the native system without requiring changes in the cyber system.

- FR-02 – *The cyber system shall adapt to changes occurring in the native system and adjust its operation accordingly.*

With different resources and topologies at the native system, it could be possible to generate different functionalities and abilities for the entire system. Hence, the cyber system must be able to adapt the functionalities at the cyber system according to the native system (FR-02).

- FR-03 – *The cyber system shall be modular.*

This functional requirement (FR-03) is particularly important in the new production systems and necessities. Due to the necessity to add and remove physical resources in runtime. This allows the companies to rapidly adapt the native system and consequently the functionalities delivered by the production system.

- FR-04 – *The cyber system shall interact with the native system dynamically and without creating permanent dependencies between both.*

One of the main concerns regarding the deployment of these solutions is the necessity to implement decoupled solutions. This avoids dependencies between the native and cyber system (FR-04).

- FR-05 – *The cyber system shall provide interfaces that support the different stakeholders in their interactions with the system.*

To increase the adoption of these solutions, it is crucial to help the different stakeholders during integration, configuration or production. Hence, the cyber system must guide the usage of the new system.

- FR-06 - *The cyber system shall provide standard interfaces that enable the interaction with the native system as well as other systems (software tools) operating collaboratively with it.*

FR-06 aims to define standard interfaces which must be used by the systems integrators to integrate new hardware or software. This FR will facilitate the work of the systems integrators as well as reduce the costs required for the integration.

The list of FRs presented previously shows the first level of FRs. These FRs were designed to be satisfied uniquely by the cyber system. From the previous list, it was analysed each of them, and for each of them it was verified if any sub-requirement must be added or not.

Hence, from the FR – 01, a new requirement was generated, and this requirement must be fulfilled to guarantee that the previous one is fulfilled.

- FR-011 - *The cyber system should mimic the availability of the production resources.*

In order for the cyber system to correctly know which functionalities and resources are available at the native system, the abstraction at the cyber system must mimic the current topology of the native system (FR-011).

This process has been used to generate a new level of requirements, for each one of the requirements generated for the first level. For the second cyber systems' FRs, the following ones were compiled.

Since the agent-based cyber system is a highly distributed and flexible environment, it is essential to deliver an infrastructure capable of dealing with the communication among the different cyber entities (FR-021).

- FR-021 - *The cyber system should provide the interaction mechanisms to reorganise the cyber level accordingly to the available resources.*

Moreover, the cyber system, based on the capabilities offered by the physical resources, must be able to know which capabilities can be offered by the cooperation among the different resources (FR-022).

- FR-022 - *The cyber system should compile the capabilities offered by the native system in order to provide new collaborative-based capabilities.*

Usually, the modularity of the cyber system is a challenging and very restrictive characteristic of the software. Hence, for the FR-03, focused on the design of a modular cyber system, four new sub-requirements were added.

- FR-031 - *The cyber system shall contain physical agents that mimic the native system's resources in one-to-one relation.*

Each physical resource must be abstracted in the cyber system through a unique agent (FR-031). This physical agent is the only cyber entity capable of communicating with the hardware of the physical resource.

- FR-032 - *The cyber system shall be capable of mimicking and operating heterogeneous resources and native systems.*

The physical agents presented in FR-031 must be designed as generic agents, capable of interfacing any kind of hardware (FR-032). This allows the integration of heterogeneous production systems as well as the reduction of the complexity of the cyber system.

- FR-033 - *The cyber system shall interact with the native system without interfering with the native control logic.*

In order to develop a cyber system which require minimal changes in the existent native system, it is required that the cyber system not directly control the hardware (FR-033). The cyber system changes the behaviour of the native system through the reparameterisation of the existent control logic.

- FR-034 - *The cyber system shall encapsulate all the information relevant to each resource with the physical agent that mimics that resource.*

Since the physical agent is responsible to abstract only one physical resource and each physical resource is abstracted by one physical agent, the physical resource must encapsulate all the information that describes the physical resource abstracted (FR-034), such as the capabilities offered by the resource, physical location, and so forth.

Regarding FR-04, focused in guarantee that the communication between the cyber and the native system does not imply and obligates any dependencies, it is vital to satisfy the following restrictions.

- FR-041 - *The cyber system shall modify the behaviour of the physical system through reconfiguration.*

The cyber system only changes the behaviour of the system through the reconfiguration of the existent control logic (FR-041). This reconfiguration can be performed through the reparameterisation of the control logic or changing the value of specific variables inside the memory of the controllers.

- FR-042 - *The cyber system should not control the physical system at the field level.*

The control of the hardware is performed by the native system exclusively (FR-042). The software existent in the line and already running must not be changed. This will reduce the impact and the changes required to deploy the new cyber system.

- FR-043 - *The cyber system shall act upon the physical system in a transactional and atomic way.*

The cyber and the native system must run in parallel and without dependencies. Hence, in order to avoid these dependencies, the two systems must communicate through transactional and atomic strategies exclusively (FR-043).

- FR-044 - *The cyber system should interact with the physical system in a bidirectional way.*

The communication between the two systems needs to be performed in parallel (FR-044). This behaviour allows the two systems to send information to the other system without delays.

Proceeding with the list of FRs from the first level, it follows the FR-05. This FR indicates that the cyber system must help the different stakeholders, whenever they need to interact with it. To achieve that, it is crucial to guarantee the following FRs.

- FR-051 - *The cyber system shall provide the user with the ability to control its autonomy and the behaviour of the physical system.*

Although most of the CPPS proposed so far aimed to optimise the production system autonomously, it is required by the companies that the autonomous behaviour of the system must be controlled by the operators who are supervising the execution of the system (FR-051).

- FR-052 - *The cyber system shall comply with the safety, security and other regulations that limit its action.*

The new functionalities proposed by the cyber system must guarantee that the existent policies are still being followed by the new integrated system (FR-052).

- FR-053 - *The cyber system shall provide the user with a transparent view of the operations.*

Since the cyber system can be a complex and very dynamic environment, it is important to keep the human informed about what is happening at the logical level (FR-053).

Finally, regarding the FR-06, the last FR presented in the first level of FRs, the following two sub-requirements were added.

- FR-061 - *The cyber system shall interact with the native system through Hardware Abstraction Layers (HAL)* (Ribeiro, 2017) *to harmonise the cyber level intentions with the equipment' control logic.*

Defining a standard interface between the physical resource and the physical agent, the effort required to integrate a new module is reduced (FR-061). The standardised interface reduces the time required to understand how to integrate the modules.

- FR-062 - *The cyber level shall provide an interface to allow and harmonise the interaction with external software tools.*

Regarding the integration with external software tools, and similarly to the FR-061, the cyber system must deliver an interface to allow the integration of new software tools. That interface allows the developers to easily integrate the new tools with the cyber system (FR-062).

Summarising the first list of requirements was created and refined. However, some new requirements were necessary to specify the first level of FRs more deeply. As shown in Figure 26, regarding the design of the cyber system, six new requirements were derived from the general FR (FR-0).

This first level of FRs constitutes all the primary requirements that need to be fulfilled by the cyber system, to guarantee that the first requirement is fulfilled. However, it is possible to see, that new lists of requirements were created. Similarly to the previous case, in order to fulfil the requirements from level 1, all the requirements in level 2 associated to that requirement must be fulfilled.

The elicitation of the FRs continued to level 3. However, during the elicitation of the requirements for this new level, it was verified that most of the FRs are not generic. Hence, the FRs created in this new level force the specification of requirements that are related to functionalities desired for a specific solution. For instance, in this level it is expected to define functionalities related to reconfiguration, monitoring, optimisation, etc. Since the objective of this work is to assess and collect the requirements needed to develop minimally invasive agent-based CPPS, in general, the process stopped at the level where it still possible to work at a generic level, in this case level 2.

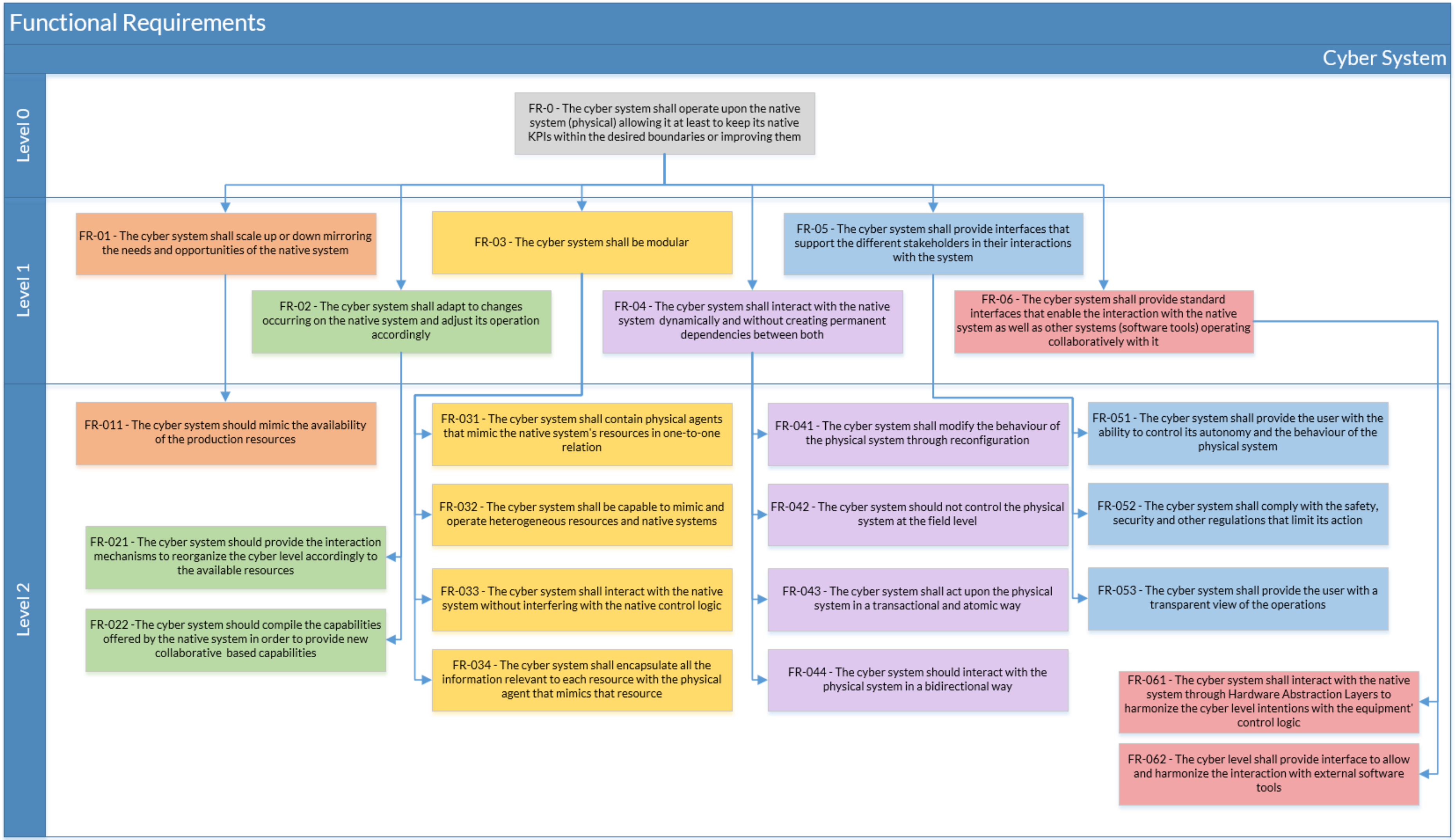


Figure 26 - Functional requirements' scheme regarding the cyber system

3.1.1. Functional Requirements for the Native System

Similarly to the previous section, a list of FRs was created for the native system. This list of requirements describes the characteristics that must exist in the native system whenever it is desired to integrate a cyber system with the previously presented functionalities.

The first list of requirements was collected and refined. However, some sub-requirements were necessary to more deeply describe the first level of requirements, as in the cyber system's case.

The first iteration of the FRs elicitation resulted in the following FRs.

- FR-01 - *The native system should be extensible.*

This FR suggests that a native system capable of adding or removing physical resources can increase the application of the functionalities delivered by the cyber system (FR-01). The ability, delivered by the cyber system, to scale according to the existent native system allows an easy extension or reduction of the native system with a reduced effort.

- FR-02 - *The native system should be modular.*

Native systems with hardware and control logic designed as modules increase the potential benefits of a new modular cyber system (FR-02).

- FR-03 - *The interface shall guarantee that the native system normally runs in the absence of the cyber system.*

Although the implementation of new interfaces to allow the communication between the native and cyber systems is required, the design and implementation of these new interfaces must guarantee that the native system runs with or without the deployment of the new cyber system (FR-03).

- FR-04 - *The native system shall provide interfaces that enable the cyber system to interact with it dynamically.*

The communication between the two systems is performed dynamically and without keeping communication channels and dependencies (FR-04). Hence, the native needs to expose interfaces that can be used by the cyber system to send and receive information at any time and without interfering with the execution of the native system.

From the previous list, only three more FRs were added. The second level's requirement for the FR-01 is the following.

- FR-011 - *The native system should communicate to the cyber system when a new physical resource is available.*

In order for the cyber system to update the representation of the native system, and the functionalities offered by the system, it is recommended that the native system inform the cyber system about the addition or removal of physical resources (FR-011).

Regarding FR-03, the following requirement was generated to guarantee that the cyber system can detect new modules automatically.

- FR-031 - *The native system's control logic shall run without the interference of the cyber system.*

The native system must guarantee that independently of the information received from the cyber system, the execution of the native system is performed by the existent control logic without any kind of interference (FR-031).

Finally, FR-04 generated the last native system's FR.

- FR-041 - *The native system should interface using known communication protocols.*

In order to reduce the effort necessary to integrate a cyber system, it is suggested that the native system uses known communication protocols (FR-041) such as Open Platform Communications – Unified Architecture (OPC-UA).

Henceforth, as shown in Figure 27, regarding the native system, four new requirements derived from the general functional requirement (FR-0).

This first level of FRs, for the native system, constitutes all the requirements that need to be fulfilled by the native system, in order to guarantee that the general requirement is fulfiled. However, it is possible to verify, that a new level of requirements was created in order to describe the previous ones better.

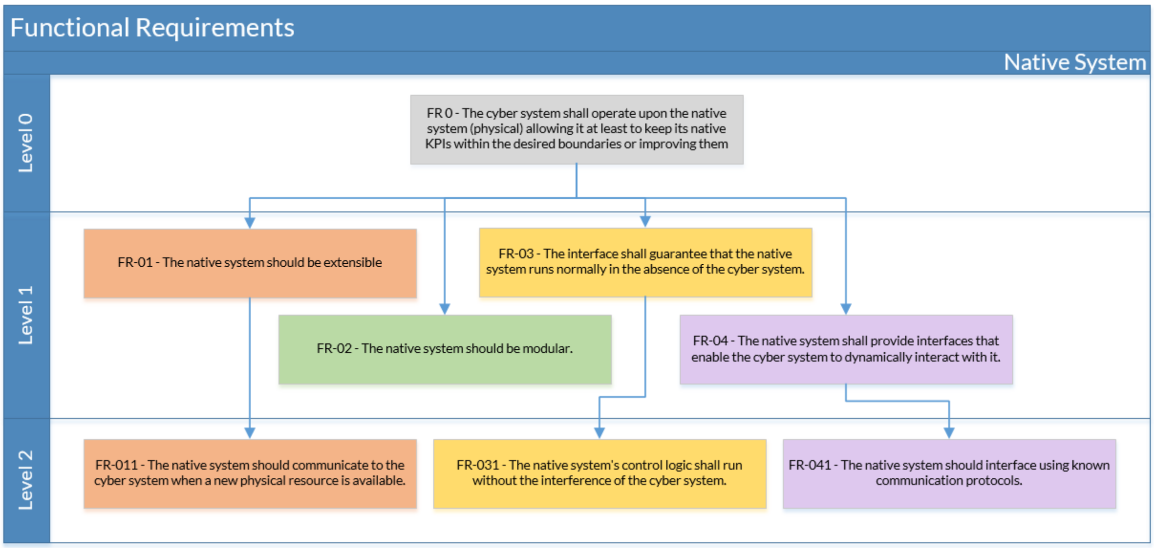


Figure 27 - Functional requirements' mind map for the native system

As described at the beginning of this chapter, the collected FRs were used as a reference for the following three experiments. Each one of the experiments was designed to influence as less as possible the native systems, but at the same time increase the overall performance of the system.

4. Design Properties

This section presents three minimally invasive agent-based CPPS experiments to implement three different functionalities. The design of the experiments generated three trees of DPs which guided the developments, one for each case. These trees were created matching the DPs with the FRs previously defined. Based on the lessons learnt from the three experiments, a tree of generic DPs is presented. This tree of generic DPs presents the DPs that must be used for designing any minimally invasive cyber system.

4.1. Design Properties for Minimally Invasive Agent based CPPS

This section aims to describe and guide the DPs' definition through the different experiments, in order to understand which DPs were generated, and how the experiments handled each of them. Two experiments were designed under the FP7 PRIME project. One of them focuses the development of a reconfigurable agent-based CPPS capable of reconfiguring the control logic of an existent production system. The other experiment designed in this project aims to deliver an infrastructure capable of extracting raw data from the hardware and generate new knowledge based on the raw data extracted. The last experiment was developed during a master thesis. The idea was to study the applicability of self-healing capabilities based on the usage of Artificial Immune Systems (AIS) algorithms.

4.1.1. Reconfigurable Agent-based CPPS

The generic FRs previously collected in conjunction with the functionalities specifically required for the reconfigurable cyber system guided the definition of the following DPs. The created DPs guided the development of such a system in order for the developed system to fulfil all the requirements.

That being said, and starting from the general and principal DP (this DP aims to deliver a functionality capable of to fulfil the FR FR-0):

- DP-0 – *The reconfigurable cyber system runs upon the existent native system keeping the performance of the control logic but optimising the used physical resources.*

The DP-0 focuses the design of the cyber system as a whole. The objective of the experiment is to develop a cyber system, which allows the entire production system of dealing with disturbances, such as the addition and removal of physical components or change the product variant being produced. Furthermore, the overall performance of the system needs to be kept and if possible improved.

Based on that, and making usage of the first level of FRs related to the cyber system, the following DP was formed (FR-01):

- DP-01 - *The MAS is deployed accordingly to the existing native system's topology, mimicking the native system at the cyber level.*

Hence, the designed multi-agent environment is responsible to abstract the hardware as well as the higher-level actions, such as the combination of different tasks that can be provided by the hardware. Each task provided by each resource is described as a skill. The offered skills can be consumed by the products. The overall picture of the proposed reconfigurable agent-based architecture is presented in Figure 28.

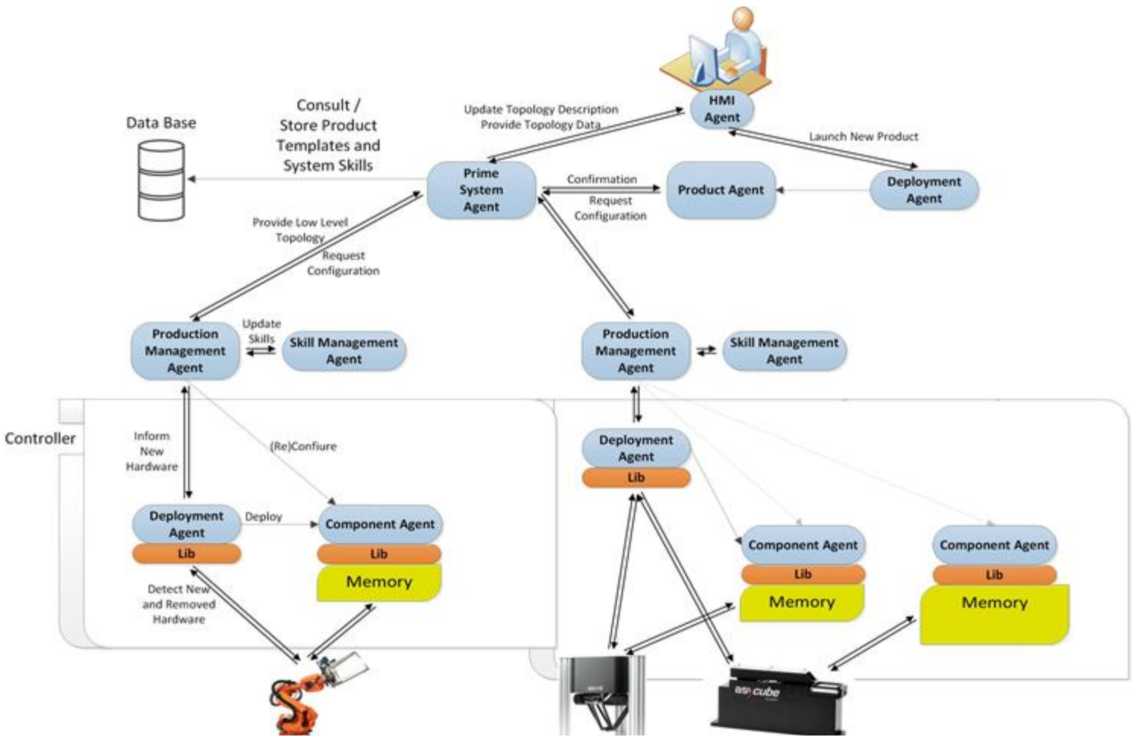


Figure 28 - Reconfigurable agent based CPPS

As shown, the overall architecture is constituted for eight types of generic agents. Each agent is responsible to abstract different functionalities and manage different information respectively:

- Prime System Agent (R_PSA) – The R_PSA is the highest-level entity in the entire system. This agent is responsible for receiving the messages from the products and automatically trigger the reconfiguration process. Since the R_PSA is the highest-level entity, it knows the global view of the entire system, regarding the available capabilities and resources. The R_PSA also manage the PRIME Common Semantic Model, storing and updating the current state of the system. Whenever a new product variant is launched, the R_PSA checks the skills available at the system and also checks the configuration files, stored in the database. If all the skills and configurations required to apply the control logic to perform the skills, the R_PSA sends the new configurations for the respective Component Agents (R_CA).

- Production Management Agent (R_PMA) – The R_PMA is a logical representation of a group of resources or other groups abstracted by other R_PMAs, that can collaborate on the shop floor. For instance, whenever a station is plugged, with different resources (Robot with a Gripper for example), a R_PMA should abstract this station to orchestrate and discover possible collaborations. With these entities, the system can have different levels of granularity according to the different layers of R_PMAs.

- Skill Management Agent (R_SMA) – Each R_PMA agent launches one R_SMA to periodically check the available skills in the sub-system and if possible generate new skills. Whenever a new resource is associated or the rules to combine the skills change, the R_SMA verifies if it is possible to associate and combine the new available skills in order to create new Composite Skills.

- Component Agent (R_CA) – Each physical resource is abstracted by a R_CA. This agent is responsible for interfacing with the hardware. The R_CA interfaces with the hardware whenever a new configuration must be loaded or whenever the R_CA desires to retrieve raw data from the hardware.

- Human Machine Interface Agent (R_HMIA) – To easily integrate an external HMI, capable of launching new products and consulting information associated to the topology of the system, the R_HMIA works as a gateway which allows the integration with an external tool. The R_HMIA publishes a set of services that can be used by the external tool to interface with the cyber system.

- Product Agent (R_PA) – Each R_PA is a logical representation of a product's variant. Whenever an operator desires to change the product to be produced, a new R_PA is launched, with a detailed description of the Skills that need to be performed in order to produce the desired product (list of necessary skills with the associated parameters). The R_PA automatically sends this list to the R_PSA, triggering the reconfiguration process.

- Deployment Agent (R_DA) – The R_DA is responsible for checking if new hardware is plugged or not. Basically, inside all the machines with processing capability, a R_DA should be launched in order to launch other agents if necessary, for instance, R_PMA and R_CA agents.

This design proposes to virtualise all the production components, collecting all the information necessary to manage the different possible configurations. Hence, all the production components are abstracted using physical agents (CAs) and higher-level agents (R_PMAs and R_SMAs) are used to aggregate the available functionalities, in order to offer collaborative based functionalities to the existent production system.

The reconfiguration process is triggered by the R_PA. Whenever the operator desires to produce a new variant of a product, the operator defines a new abstraction of the product variant and a new R_PA agent is launched in order to start the reconfiguration of the production line.

To match the capabilities required by the product type with the capabilities offered by the resources available in the native system, both product variant and the capabilities available in the native system are abstracted as skills. The skills describe the tasks offered by the physical resources and consumed by the R_PA, which abstracts each product variant. The configuration is triggered by the R_PA and it is applied to the R_CAs available and capable of to offer the skills required by the product.

The availability of the physical resources connected is one of the functionalities required for any CPPS, and for this specific case, it is imperative to know which physical components can be reconfigured (FR-011).

- DP-011 - *The CAs are launched whenever a new physical resource is plugged and periodically check the availability of the abstracted resource.*

The physical agents (in this case, the R_CAs) are responsible for sending the information that describes the physical resource, and it knows how the MAS can communicate with the physical component (hardware). For instance, whenever a new component is plugged, the R_DA launches a new R_CA to abstract the component. The R_CA will automatically consult (in a local library or remote database) all the information that is relevant to the cyber system. Hence, the R_CA knows the skills that can be performed, the physical location, and the library that must be used to interact with the hardware.

After being launched, the R_CA shares that information with the cyber system. Now, this new component is ready to be used and reconfigured whenever necessary.

Whenever a component is not available due to an energy or connection problem, the R_CA is removed from the multi-agent platform. This update automatically informs the cyber system that the resource is no longer available because the agent is no longer alive. On the other hand, if the R_CA is alive the abstracted component is ready to be reconfigured. Furthermore, the R_CA checks periodically if any failure or maintenance task was trigger by the operator. If the R_CA announces this new state, the resource is removed from the list of possible reconfigurable resources. On the opposite, if a R_CA is alive, and any information has been sent, to the highest-levels, regarding the availability, the cyber system recognises that the component is available and ready to produce and to be reconfigured. These cyber system's behaviours are shown in Figure 29.

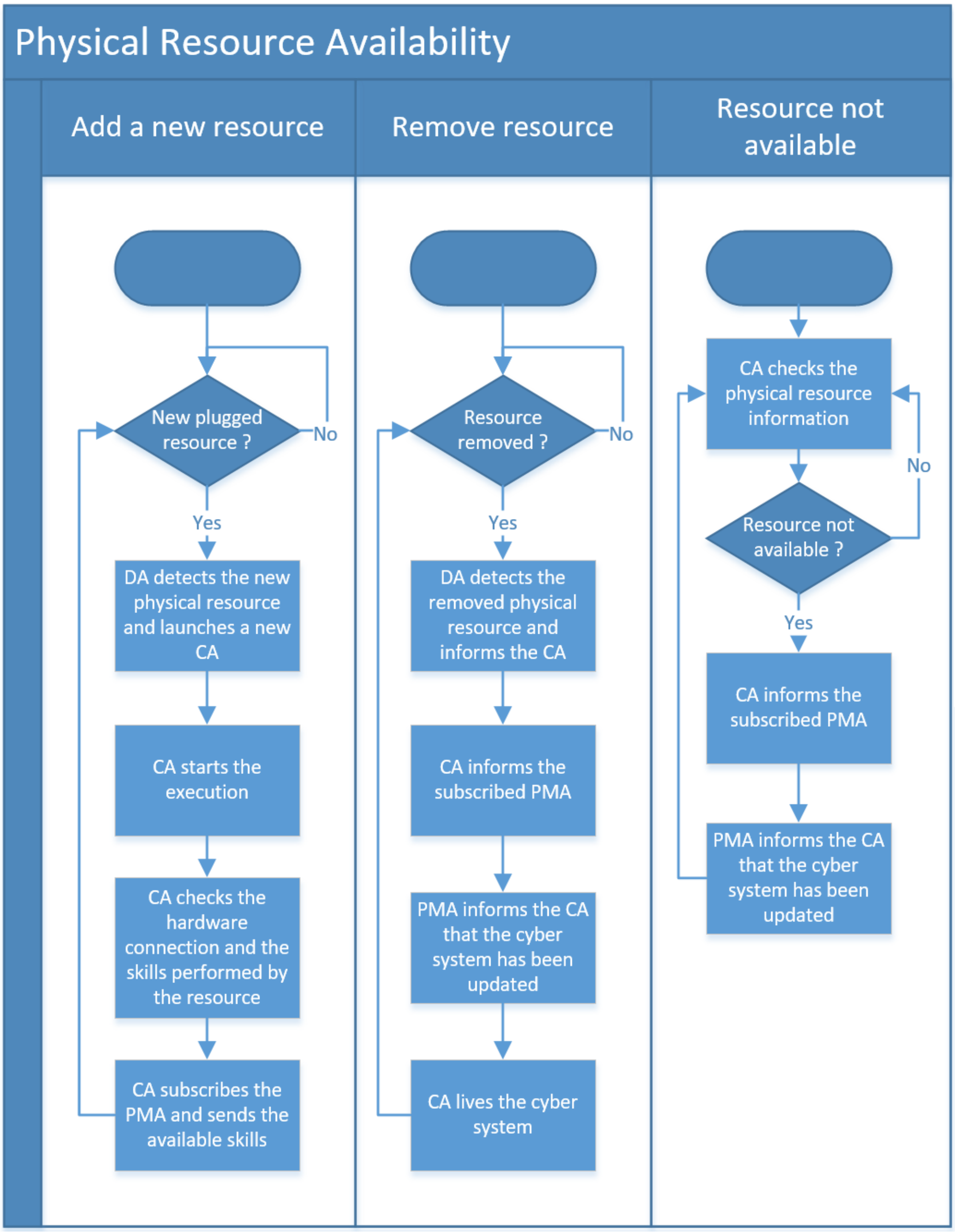


Figure 29 - Checking the physical resources' availability

In order to allow the addition or removal of physical components, the cyber system was designed and implemented in order to deal with disturbances. The cyber environment is structured as a tree of R_PMA agents, which abstracts the sub-systems that can be found in a native system. A sub-system is an area or location where different resources, or lower-level sub-systems can work together and collaborate with each other. An example of this tree is presented in Figure 30.

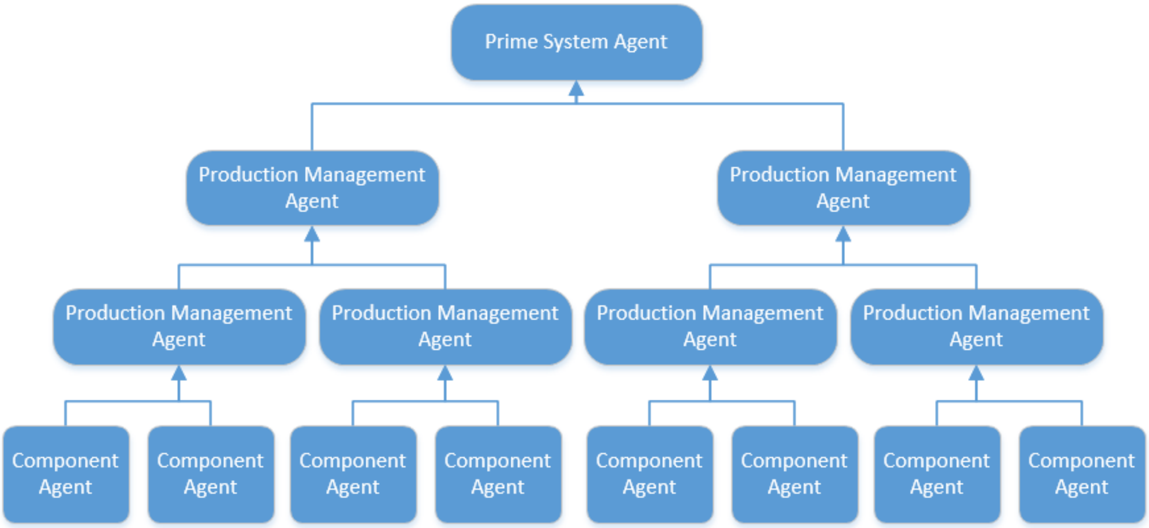


Figure 30 - Tree of R_PMAs

The platform, to instantiate the cyber system, must start with the R_PSA. The R_PSA is the agent responsible for managing the topology of the system and, the database containing the configurations that can be deployed. After launching the R_PSA, the tree of R_PMAs is instantiated. Each R_PMA launches an associated R_SMA, and for each R_PMA it is instantiated only one R_SMA. At the end of this process, the entire tree of R_PMAs and the R_PSA are running, the CAs can be deployed, in order to abstract the native system.

The tree with the R_PSA, R_PMAs and the associated R_SMAs constitutes the core of the cyber system. This core is responsible for updating the cyber system when disturbances occur (FR-02).

- DP-02 - *The cyber system adapts the tree of R_PMAs and R_PSA topology and available capabilities accordingly to the connected physical resources.*

Whenever a new R_CA is plugged, this new R_CA subscribes the R_PMA responsible for the sub-system where the R_CA is operating. During the subscription, the R_CA sends a list with all the skills offered by the physical resource. Hence, the R_PMA can update the list of the skills available in the sub-system. Whenever the skills available in the sub-system change, the new list of skills is sent to R_SMA. The R_SMA, matching the available skills with the rules defined for the generation of collaborative based skills, will try to generate higher-level skills, which can be offered by the R_PMA. If any new skill is generated, the R_PMA, similarly to the R_CA case, sends the skills offered by the R_PMA to the R_PMA above it, and this higher-level R_PMA will start the process locally. This behaviour proceeds until the tree the R_PSA is achieved. At the end of this process, the R_PSA will have a global view of all the skills available in the system and where that skills are offered.

The update of the cyber system is made using a communication mechanism. This mechanism was developed for all the R_CAs, R_PMAs, R_SMAs and the R_PSA. This mechanism allows the update of the topology and the skills available at the cyber level (FR-021).

- DP-021 - *The agents communicate through FIPA compliant protocols to update the tree of R_PMAS, R_SMAs and R_PSA.*

The MAS is entirely FIPA compliant (Bellifemine et al., 1999) and all the agents communicate using the defined FIPA protocols. Moreover, the update of the cyber system is made through messages exchange among the agents (R_CAs, R_PMAs, R_SMAs and R_PSA). Whenever a new R_CA is launched, the R_CA starts a conversation with the R_PMA in order to inform that a new R_CA is available in the sub-system abstracted by that R_PMA. On the other hand, when the R_PMA receives the message from the R_CA with the update, it starts two other conversations, one with the R_SMA and another with the R_PMA above itself. If there is no any sub-system above the current R_PMA, the node of the tree above it is the R_PSA. The tree is updated until the R_PSA is achieved, as presented in Figure 31, with the sequence diagram.

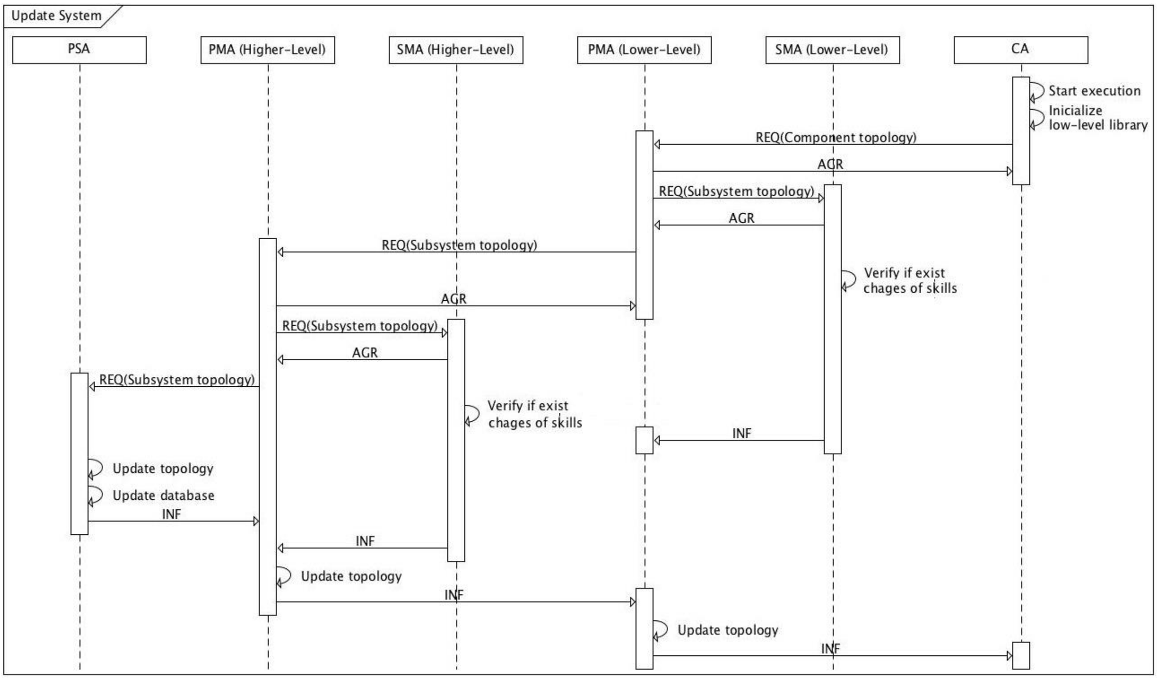


Figure 31 - Communications when the cyber system is updated

The communication with the R_SMA aims to send the new capabilities available in the sub-system (received from the R_CA or a R_PMA above the sub-system). The communication with the R_PMA from the higher-level encompasses the update of the entire tree of R_PMAs.These process proceeds until achieving the root (P_PSA). When the R_PSA receives the information, it updates the topology of the system.

The capabilities offered by the native system, posteriorly consumed by the R_PA to indicate the production steps required, are described as Skills. The definition of Skill is always complex and no consensual due to the required complexity to describe all of the possible tasks that can be provided by the physical resources, and at the same time describe the capabilities that need to be consumed by the product.

Hence, a minimal set of information is required to identify a Skill as unique and easily understandable, not only by the cyber system but also by the human. The Skills are essential for the integration process. The systems' integrators will need to describe, to the cyber system, which capabilities can be performed by the different resources. Whenever a new product variant is added, the human operator must design the product variant as a set of Skills that need to be performed.

As previously described, the proposed cyber system can combine and provide higher-level capabilities. Those higher-level capabilities describe a combination of other skills, that combined can offer collaborative capabilities (FR-022).

- DP-022 - *The R_SMAs can compute the sub-system's capabilities and generate new collaborative capabilities, with the capabilities offered by the subsystems' CAs and R_PMAs.*

Hence, it is possible to describe the offered Skills as:

- Simple Skill (SSk): All the skills offered by a R_CA are described as SSks. A skill of this type describes a task that is performed by a physical resource and it can be described as an atomic task. It is executed by one physical agent (CA) only;
- Complex Skill (CSk): Whenever two or more SSks can be combined, to offer a new higher-level ability, a new CSk is generated. Similarly, it is possible to combine CSks in order to generate new CSks, or even combine SSKs with CSks. Hence, whenever a R_SMA communicates to the

associated R_PMA that a new CSk is available, the R_PMA will announce the new available skill to the entire cyber system.

For the R_SMAs to generate CSks, it is necessary to define which rules must be followed for them in order to know how to combine the skills associated with the sub-system. To do so, the database managed by the R_PSA contains the rules that define all the possible combinations of Skills.

During the execution, it is possible to add and remove rules to the R_PSA. The R-PSA update the database with the new rules dynamically. Whenever a rule is added or removed, it is triggered a routine to update the rules into the R_SMAs. Hence, the R_SMAs will recompute the CSk and update them within the R_PMA, based on the new rules. In Figure 32 it is shown how the cyber system updates the rules used to create new CSks.

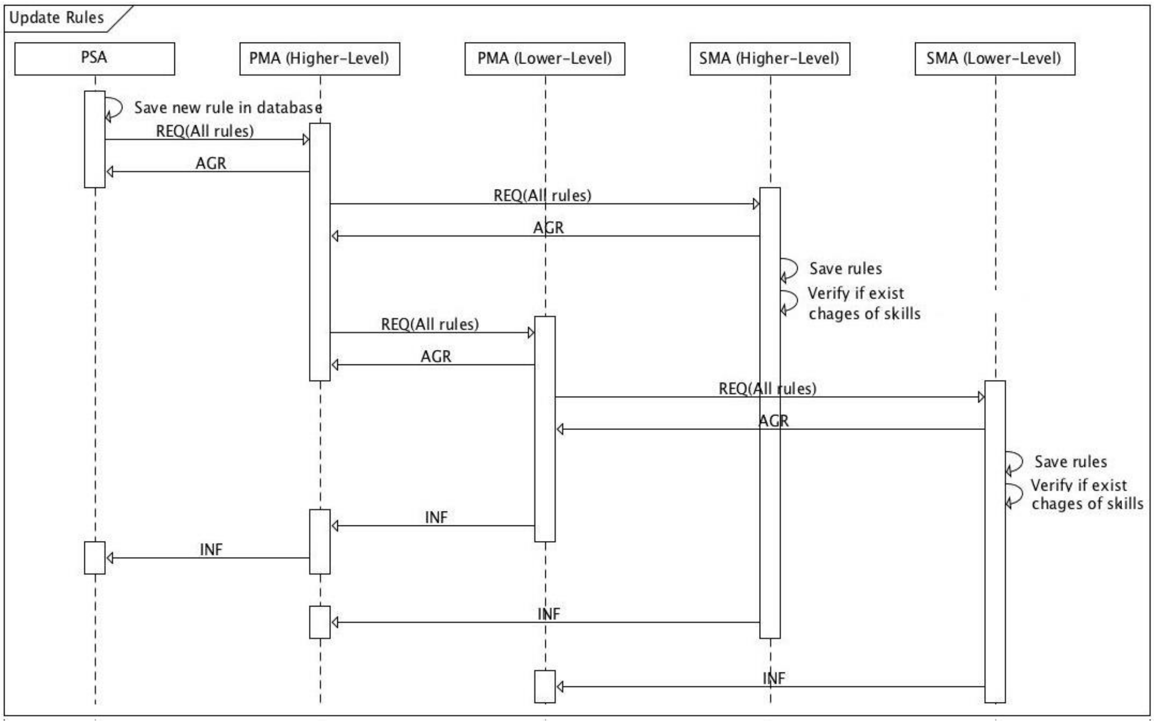


Figure 32 - Rules' update

The rules identify not only which Skills can be associated in order to generate collaborative based Skills, but also specifies which parameters must be verified in order to generate the new one. As presented in Figure 33, a Skill is described through an ID, and all the information to identify the Skill, as well as a list of associated Parameters.

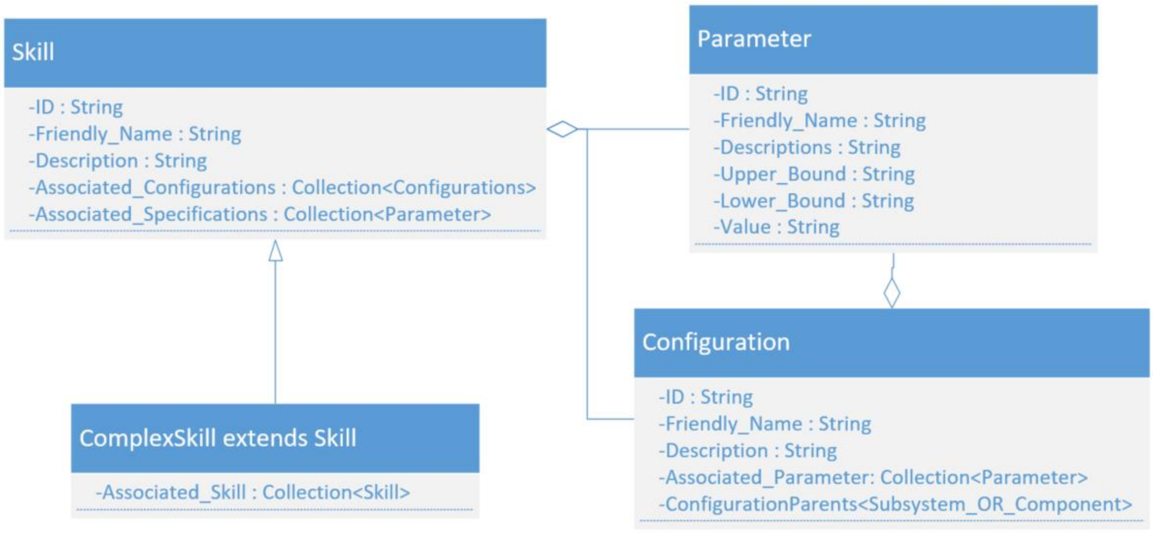


Figure 33 - Simplified data model

A rule is represented as a CSk, containing all the Skills that need to exist in order to create it. For each Skill, as it is possible to see in the Class Diagram, a list of Parameters describes which Parameters must be verified by that Skill and the boundaries of a specific value.

The Upper_Bound, Lower_Bound and Value were designed to describe any necessary verifications. Table 4 presents how these values must be filled for the different cases. For instance, if a Parameter needs to be offered with a value below 50 and it there is no lower limit, only the Upper_Bound must be filled with the value 50.

Table 4 - Rules definition

		Verifications			
		Above the limit	Bellow the limit	Between the limits	Specific value
Variable	Value	Not Defined	Not Defined	Not Defined	Defined
	Upper_Bound	Not Defined	Defined	Defined	Not Defined
	Lower_Bound	Defined	Not Defined	Defined	Not Defined

Hence, the cyber system is capable of understanding the Skills available in each sub-system and it is also capable of generating collaborative based capabilities.

The inherent capabilities are offered by each plugged physical resource on the native system. Each physical resource and the respective abstraction within the cyber system is done through a modular approach. Henceforth, the following DP was defined (FR-03):

- DP-03 – *Each physical resource is abstracted through an agent, and this aggregation constitutes a module.*

Whenever a physical resource is plugged into the native system, the cyber system launches a R_CA capable of communicating with the new resource. This modular approach allows the cyber system to deal with disturbances as plug or unplug of components without requiring stoppages or any reprogramming effort.

The first necessary DP that emerged from this specification is (FR-031):

- DP-031 - *The R_CA works as a physical agent, responsible to abstract only one physical resource and interfaces with the hardware.*

The R_CA is responsible to abstract a physical resource, in this sense the proposed R_CA connects and interacts with the hardware. This agent is able to reconfigure the native control logic running on the physical component. This agent is responsible for detecting when the physical resource is connected and ready to produce. Moreover, the R_CA must also know, all the information needed to describe this component, from a logical point of view. For instance, all the SSks that can be performed by the component and the physical location must be retrieved by this agent and announced to the other cyber entities that need to know this information, such as R_PMAs and R_PSA.

Whenever an interaction between the cyber system and the physical component must be performed, only the R_CA responsible to abstract that physical component can perform this task, since the R_CA is the only agent that knows, through the library for that purpose, how to communicate with it. Hence, whenever a new configuration must be deployed in the hardware, all the physical agents (CAs) must be informed about it and each R_CA will deploy the respective configuration in the abstracted component.

With the abstraction of a physical resource through the R_CA, it is important to guarantee that the generic implementation of the R_CA is capable of abstracting different hardware, from different vendors and with different specifications (FR-032).

- DP-032 - *The R_CA is a generic agent which communicates through a standard interface, and in this sense, the R_CA can reconfigure and interface any physical resource.*

Through the usage of a standard interface, to isolate the physical agent from the control logic and hardware specific code, each physical agent (R_CA) is a generic entity that communicates with the hardware through this interface. Hence, it guarantees that all the possible physical components (from different vendors, with different capabilities, control logic's technologies or standard) can be connected and abstracted using similar physical agents. The difference between the R_CAs are the parameters that describe the abstracted component, and the library used to interface with the hardware and control logic. All the other aspects, such as the internal description of the components, communication with the other different cyber entities and so forth, are generic and in this sense, the platform can, generically and without any programming effort, dealing with heterogeneous modules.

Using the standard interface, the R_CA can configure or parametrise the existent control logic. However, and since the objective was to design a minimally invasive solution, the following DP emerged (FR-033).

- DP-033 – *The reconfiguration process only updates some variables' and parameters' values without directly interfering with the existent control logic.*

The reconfiguration process only takes place calling a method for that specific task. Through that method, the physical agent sends all the parameters and information needed to perform the reconfiguration. In most of the cases, the reconfiguration consists in changing some values inside the Programming Logic Computer (PLC) or a similar controlling device. However, in other cases, if that functionality is available, the entire control logic can be generated and changed inside the lower level library. This lower level library implements the interface between the physical agent and the hardware.

The information necessary to perform the reconfiguration is specific for the abstracted component and it is defined for each case. However, as far as the configuration data is available in the database, the cyber system can manage it and send the available configuration file or parameters for the respective R_CAs. After receiving the configuration objects, the R_CAs apply the configuration to the hardware.

To allow the cyber system to perceive which resources are plugged and the current state of each component, the R_CA needs to know which physical resource is being abstracted. That being said, the R_CA needs to have all the information necessary to describe the physical resource, such as capabilities, physical location, among others (FR-034).

- DP-034 – *The R_CA contains and manage all the critical information regarding the abstracted resource, such as physical position, capabilities, applied configuration, among others.*

The description necessary to describe each physical component is managed and encapsulate by the R_CA (physical agent) that abstracts the component. In the proposed cyber system, whenever a new physical component is plugged, a new R_CA is launched to abstract and mimic that physical component. Immediately after being launched, the physical agent consults a library where is stored all the information related to the mimicked resource. In this library, the R_CA will find which capabilities can be offered by the component, the current position where the component is plugged, the sub-system where the physical component is working and also the library that must be instantiated and used to communicate with the hardware to extract data and apply the configurations.

Appling the previously presented DPs, a reconfigurable cyber-physical production component can be described as presented in Figure 34.

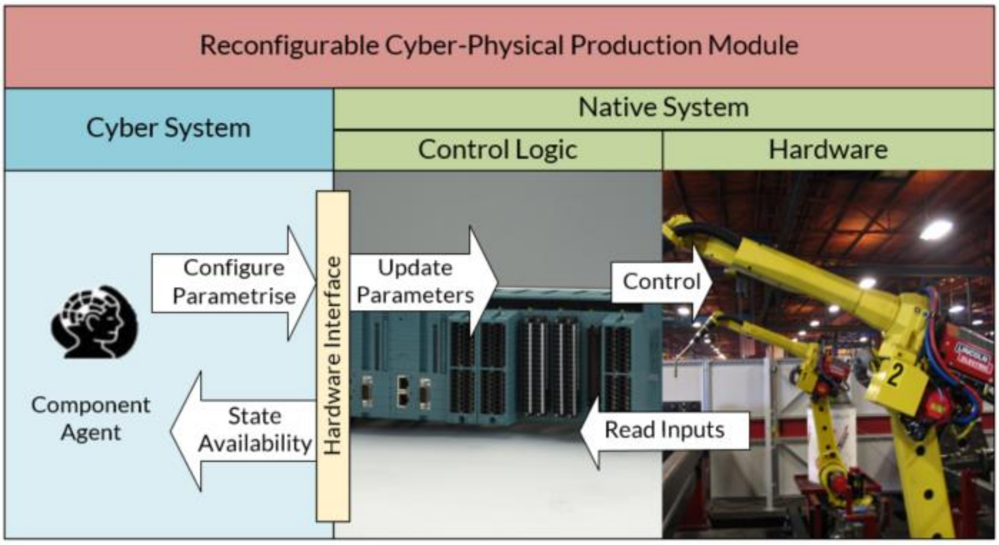


Figure 34 - Reconfigurable cyber-physical production module

Although the proposed cyber system aims to improve the usage of resources, reducing the time and effort necessary to reconfigure the existent native system, one of the most critical requirements is the need for the native system continue to run without the cyber system, avoiding external dependencies. Hence, one of the most critical and imperative DPs is (FR-04):

- DP-04 - *The cyber system can be deployed and removed at runtime, and the native system continues the production normally without this functionality.*

Hence, the cyber system communicates through standard interfaces designed to avoid that dependency.The execution of the native system's control logic does not depend on the cyber system since the configuration does not change the control logic's code. The reconfiguration process only updates parameters and values inside the controller or PLC.

The proposed cyber system cannot directly control the native system. In fact, the only action that the cyber system can provide to the native system is to send a list of new configurations that must be applied by the native system. Based on that, the following DP was proposed proposed (FR-041):

- DP-041 – *The cyber system changes the behaviour of the system whenever a new product variant is desired. Launching a new product variant, the cyber system will apply the configurations (changing the variables' and parameters' values).*

Whenever the operator chooses a new variant to be produced, a new R_PA is launched into the platform. This new R_PA sends its ID to the R_PSA together with the parameters defined by the operator (Skills). Consequently, the R_PSA starts the routine to reconfigure the system.

Firstly, the R_PSA verifies if the Skills requested by the product are available in the system by consulting the common semantic model, which is updated whenever the topology of the system changes. If all the capabilities necessary are available, the R_PSA consults the database to verify whether there exist available configurations to be sent to the plugged components. After this verification, the R_PSA selects the best configurations for each requested capability according to predefined priorities. The R_PSA then sends each selected configuration to the R_CA that must be reconfigured. The R_CA, in turn, writes this configuration on its corresponding hardware, an overview of this process can be seen in Figure 35.

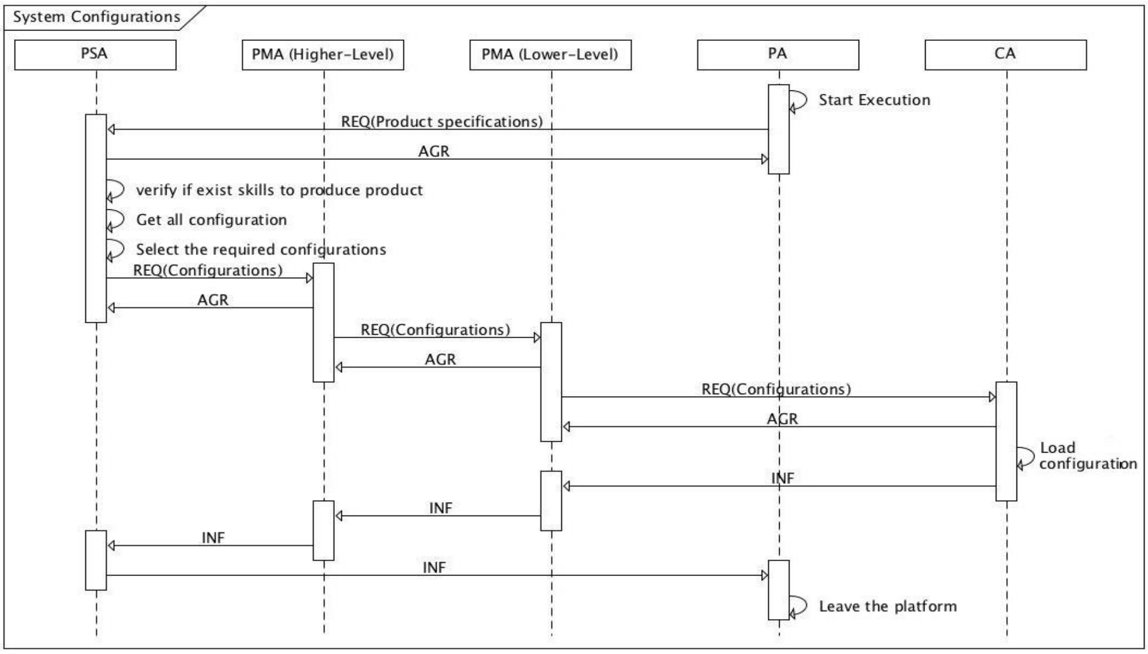


Figure 35 - Configuration process

The configuration process needs to take place without the R_CA directly controlling or interfering the native standard control logic (FR-042).

- DP-042 - *The R_CA is not able to directly control the hardware. The R_CA was not designed in such a way, and the standard interface does not allow the access to the I/Os' hardware.*

The field level of the native system is controlled by the already implemented control logic only and using the standard technology for each of the components. So, each hardware is controlled by the native programming language and following the standards proposed by the vendors. The only interaction by the cyber system is the reparameterization of some values that will posteriorly be used by the native control logic to change behaviours. The reparametrisation will force the existent control logic to trigger different tasks or routines as well as to use different parameters when executing the already running tasks.

So, to not interfere with the existent control logic the standard interface, when implemented, needs to deliver a new parametrisation in an atomic and transitional way (FR-043).

- DP-043 – *Whenever the R_CA desires to send information (configuration) to the native system, and vice-versa, whenever the native system wants to send information to the CA a method is called to deliver that information.*

All the communications through the library, responsible for integrating the two systems (cyber and native system), are transactional and atomic. That being said, whenever information needs to flow from the physical agent to the native system or vice-versa a method is called in the library, and the information is delivered from one point to another without any kind of communication channel and without keeping an open channel or dependency.

Another requirement that is important to guarantee is the capability for the cyber and native system to communicate in parallel and simultaneously (FR-044).

- DP-044 – *When the R_CA is calling the method to send information to the native system, the native system can call the method to deliver information to the CA, and vice-versa.*

The communications between the two systems are fully asynchronous and do not depend on each other. That being said, when a physical agent is sending a configuration to the native system, it can also receive information regarding the state of the hardware.

This guarantees that the overall performance of the system is not degraded to the fact that the system is waiting to deliver any message. More specifically, if the physical agent is sending information to the native system does not obligate the native system to wait to send a message to the cyber system.

On top of all the functionalities already presented, it is crucial to guarantee the communication with external systems and tools to support the interaction of the human with the cyber system (FR-05).

- DP-05 - *The cyber system provides a list of services that can be used by external tools to show what is happening within the cyber system and to trigger the reconfiguration.*

With the development of intelligent and self-organised systems, it is imperative to guide the humans and to keep the human personnel updated about what is happening and what is required by the cyber system at any time (configuration, integration, support, among others).

Hence, the first DP regarding this functionality is focused on the configuration process (FR-051).

- DP-051 - *The configuration of the native system, through the cyber system, is triggered by the operator using an external tool for that.*

Providing a set of services, available to trigger the configuration process and to provide all the information needed for the configuration, any software provider can develop tools to interface with the cyber system.

The R_HMIA hosts a set of services that can be called by external tools in order to consult the current state of the system and select a variant of a product from the set of variants defined and stored into the database.

Whenever a human operator intends to produce a new variant, the R_HMIA sends a message to the R_PSA. The R_PSA, with the information received from the interaction with the R_HMIA, such as a variant of the product, parameters, and so forth, checks the available topology of the system, verifying the plugged and available resources, and the Skills that need to be performed to produce the desired variant. If all these conditions are verified, the R_PSA triggers the reconfiguration, sending the configurations necessary to the R_PMAs, which will send to the R_CAs.

The safety, security and other regulations are fulfilled by the new proposed cyber system since the cyber system maintains all the initial regulations verified by the native system.

Since the proposed system only runs upon the existent system and does not change any parameter or characteristic of the native system that could create issues regarding the existent regulations (FR-052).

- DP-052 – *Since the cyber system does not change critical control logic parameters, such as security, as far as the native system guarantees the regulations, the whole keeps guaranteeing those regulations.*

The R_HMIA aims not only to allow the reconfiguration process but also to offer to the human operator the possibility to visualise which components are available to use and which variants are being produced (FR-053).

- DP-053 – *Through the R_HMIA it is possible to deliver external tools capable of showing the human operator which resources to use and the product's variant being produced.*

Hence, whenever the cyber system is running, an external tool, for instance a web-based or mobile tool, can be used for the operator to consult and visualise what is happening in the cyber system. Whenever a new product variant is started, the operator can see which devices are being used to produce and what variant is being produced.

As previously specified, the interfacing with external tools and hardware needs to be performed through standard interfaces designed for that purpose (FR-06).

- DP-06 - *The cyber system provides two standard interfaces that must be used by the systems' integrators or developers to develop and integrate new functionalities.*

The integration between the hardware and external software tools is always a big challenge. The proposed cyber system offers a standard interface that must be used by the systems' integrators to integrate each hardware module and another standard interface that allows the integration with external tools to visualise the system and to allow the operator to launch new product variants (FR-061).

- DP-061 – *Each R_CA uses a Hardware Abstraction Layer (HAL) to communicate with the native system and vice-versa. The designed HAL is generic and equal to all the R_CAs.*

This interface exposes the methods which can be used by the physical agent to write a new configuration.

The HAL is the same for all the instantiated CAs. Hence, all the communications with the physical components use a similar HALs.

The following methods, defined in the HAL, must be implemented by the systems' integrators when a new resource needs to be integrated:

  - getComponent(): This method retrieves all the information needed to describe the plugged resource, such as skills, location, and so forth;
  - intHardware(Component): This method must be called at the beginning of the execution to check if the communication between the hardware and the R_CA is possible and it is working properly;

- getState(): This method must be called by the CA to check the current state of the hardware;
- configure(Configuration): This method is called whenever a new configuration is sent to the hardware.

Regarding software tools, an abstraction layer has been defined with a set of services, allowing external tools to visualise the state of the native system and to trigger the configuration process (FR-062).

- DP-062 – *The R_HMIA exposes a set of services that must be used to interface the cyber system. This interface allows external tools to show the current state of the cyber system and start the reconfiguration routine.*

The interaction with external tools is allowed through the methods exposed by the R_HMIA.

The exposed services allow the external tools to retrieve information about the current situation, such as available resources and the variant which currently being produced. Moreover, they also allow the possibility to trigger a reconfiguration of the native system in order to start the production of a new variant.

The following methods, defined in this abstraction layer, must be implemented by the systems' integrators in order to integrate a new tool:

- getTopology(): This method retrieves all the information regarding the topology currently running at the shop-floor since the cyber system mimics the native system;
- getProducts(): This method retrieves a list of all the products that can be produced with the current native and cyber system;
- startConfiguration(Product): This method must be called by the operator whenever a new product variant is desired. This service sends a description of the product variant to the R_PSA. R_PSA checks if the Skills and Configurations currently available are enough to produce the product variant, if yes, it starts the process;

A scheme of the defined DPs for the reconfigurable multi-agent environment can be found in Figure 36.

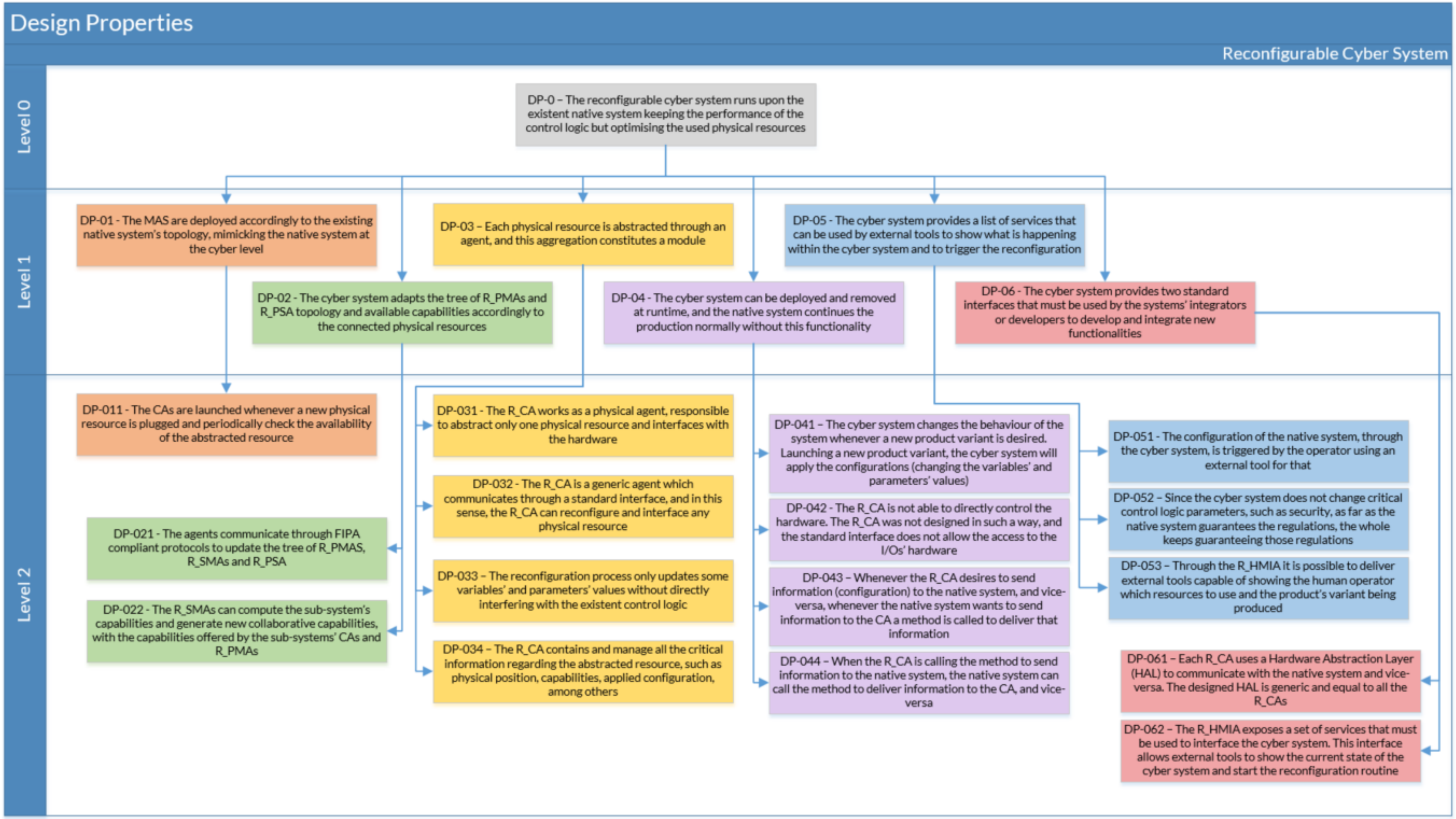


Figure 36 – Design properties for design a reconfigurable cyber system

4.1.2. Self-Monitoring Agent-based CPPS

The previously collected FRs in conjunction with the functionalities required for the self-monitoring cyber system generated a list of DPs. The generated DPs guided the development of the system capable of to fulfil the requirements collected.

Data extraction is usually viewed as a problem related to system integration. Usually is seen as the retrieval of all the stored data and the data provided in real time by the system, not only by the controllers (e.g. PLCs) but also by any other devices that can provide, and store data related to the system's execution.

However, usually the provided data is raw data, and consequently, it is useless both for monitoring purposes and to describe the behaviour of the system. Most of the time the raw data stored in the PLC / Controller's memory is just bits that can describe a state referent to the sensors and actuators' values, such as activations and positions.

In some more advanced systems, the PLC contains some values that can be used for monitoring and to describe the system in an understandable way for the operators, with times or counters. However, the usage of this data is not so easy because it is hard to collect this data from an industrial machine and display it in a common user-friendly visual interface.

The proposed system's main goal is to provide a middleware capable of collecting data from the lowest level devices, such as controllers or PLCs, and offer it to external entities, such as databases, cloud-based processing entity or HMIs. The extracted data provided from the resources can be of two types: useful or useless data. The useful data can be consumed by the external entities automatically, and the useless data is used by the agents to understand what kind of routines have been performed by the resources and in this way compute the useful data values so that they can be used later by the external entities.

The proposed CPPS aims to use the latest ICT developments to reduce the computational load usually done on the shop-floor to perform monitoring. This approach has two big effects:

- Improvement of system performance – By reducing the monitoring load inside the controllers, the control system has more hardware capacity available. With this the control system improves its performance and consequently the performance of the entire system;

- Increasing the possibilities of monitoring – Using an external entity, such as a remote server or a data centre with a substantial computational capacity, the amount of data that can be processed and the number of variables and combinations that can be modelled and analysed is increased, resulting in better and more accurate monitoring.

That being said, and starting from the general and central DP, the system must :

- DP-0 – *The cyber system only extracts and analyse data, maintaining the control logic performance of the native system.*

The DP-0 (FR-0)focuses the design of the cyber system as a whole. The objective of the design is to develop a cyber system, which self-monitoring manufacturing resources and sub-systems in different levels. The cyber system was defined to be generic and useful for any manufacturing system, providing a generic middleware which is capable of interfacing with libraries responsible for extracting and sending data to and from the multiagent environment.

Three generic types of agents compose the proposed agent-based architecture. Each type is responsible for abstracting different parts of the manufacturing system. The main goal of the proposed architecture is to perform monitoring at different levels and within any system and technology.

In Figure 37 a global overview of the entire multi-agent environment is presented with all possible types of agents and external connections, such as connections to collect or to send data to external environments.

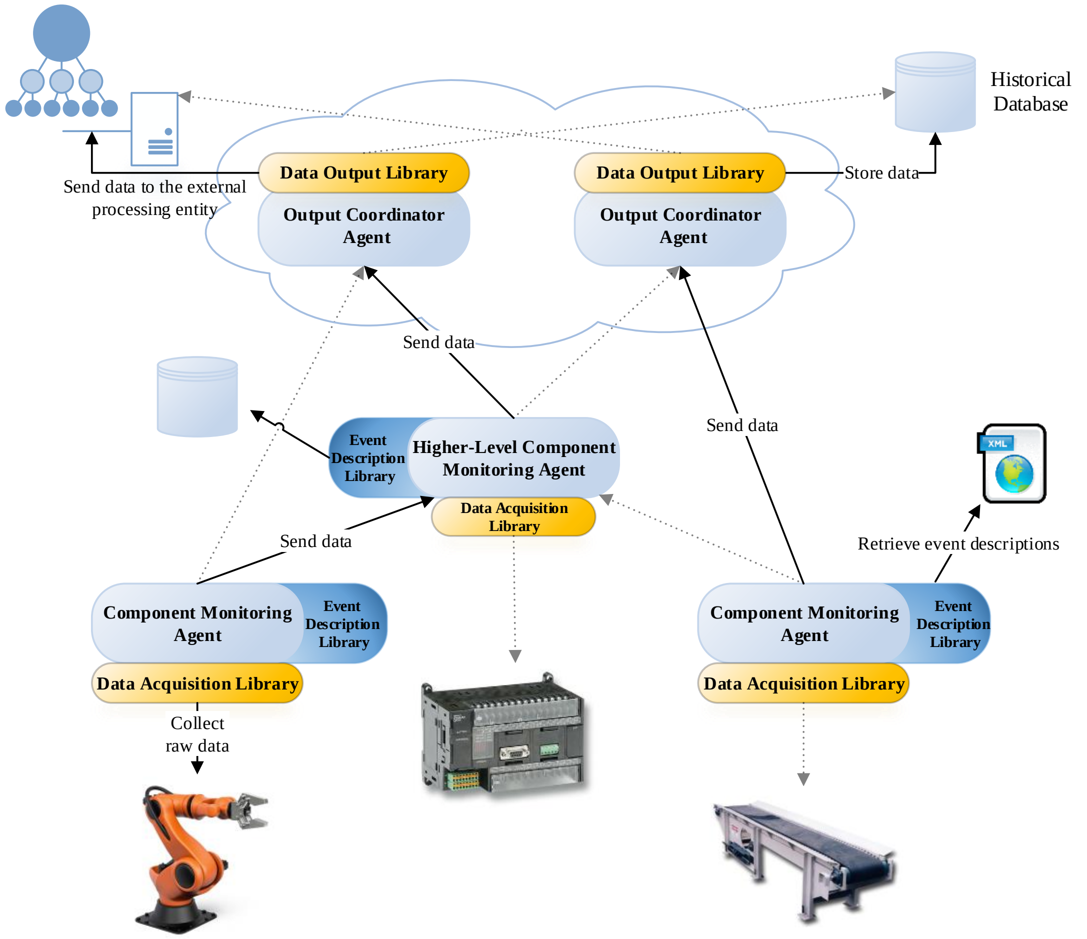


Figure 37 - Self-monitoring agent b3ased CPPS

The proposed reference architecture is composed of four different generic agents, namely the Component Monitoring Agent (M_CMA), Higher-Level Component Monitoring Agent (M_HLCMA) and the Output Coordinator Agent (M_OCA):

- Component Monitoring Agent – The M_CMA represents the lowest level entity in the architecture. Each M_CMA abstracts a physical resource such as a robot, conveyor belt or tool, summarising a cyber-physical device which composes part of the production line. This agent periodically collects raw data from the physical device using the Data Acquisition Library to interface the respective hardware. This library is specific to each resource and makes the raw data available for the M_CMA. After the collection of the raw data, the M_CMA pre-processes the information to retrieve important values from the component's performance, such as transition times and action times. To allow this, each M_CMA contains a list of rules that describe possible events that can be retrieved from the raw data provided by the component. These rules are loaded at the beginning of the agent's execution, using the Event Description Library to access the repository (XML file or database) and respective rules.

- Higher-Level Component Monitoring Agent – When a set of resources work together to perform higher-level capabilities in the line it is crucial to have higher level monitoring running at the same level as well, having, in fact, the same granularity as the aggregation of capabilities performed by this set of components. Hence, the M_HLCMA is responsible for abstracting a certain set of cyber-physical components which can cooperate to offer higher-level capabilities. In sum, each M_HLCMA abstracts a subsystem in the line (station, workgroup, etc). The M_HLCMA receives raw data from a computational device, such as a computer or a PLC containing data referent to the abstracted subsystem and receives pre-processed data from the M_CMA which subscribed this M_HLCMA. All the cyber-physical devices that can cooperate must subscribe the same M_HLCMA. As the M_CMA, the M_HLCMA has inherent rules used to pre-process the data received from the computational device and M_CMAs. This type of agent implements the Data Acquisition Library and the Event Description Library as in the M_CMA to allow the same capabilities in the M_HLCMA. In this case, the raw data received from the Data Acquisition Library belongs to the subsystem and not to a cyber-physical component (Ex. Cycle time of the station, emergency stop, among others).

- Output Coordinator Agent – All collected data from the system needs to be stored and processed in order to improve the monitoring of the system as a whole and to detect more detailed behaviours and trends of the components and subsystems. Hence, all the data collected and pre-processed by the M_CMAs and M_HLCMAs is sent to the highest-level entities in the entire environment, the M_OCAs. The M_OCA level works as a cloud of agents with the same responsibilities. Each M_OCA is responsible for allowing an easy interaction between monitoring agents (M_CMA and M_HLCMA), historical repositories and external applications and environments capable of computing massive amount of data. The M_OCA implements a Data Output Library that allows the agent to store the data in a remote repository and send the information to the external environment to be computed.

- Deployment Agent (M_DA) – The M_DA is responsible for checking if new hardware is plugged or not. Accordingly to that information, the M_DA will launch the other agents.

The proposed cyber system was designed to dynamically and in runtime mirroring the available physical resources and clusters of them. The proposed four generic agents were designed to be deployed and removed from the platform in such a way that the cyber environment can mimic any native system, as presented in Figure 38.

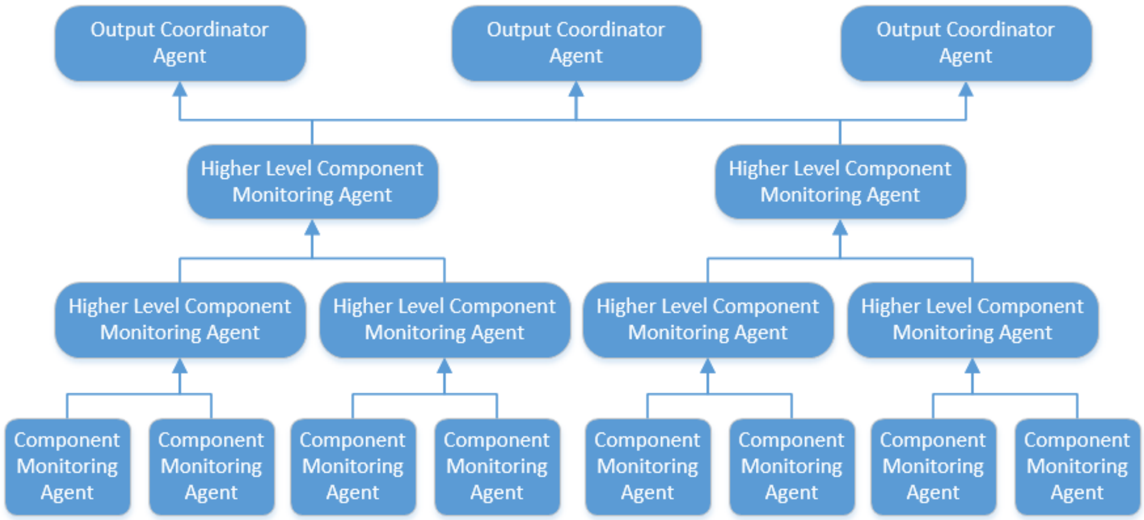


Figure 38 - Self-monitoring multi-agent system topology

When the cyber system is launched the current topology of the system is collected from the a which describes the system and all the M_CMAs are launched accordingly. During execution, if a new physical resource is connected a M_CMA is launched to abstract that physical resource, and vice-versa when a physical resource is unplugged. Regarding the other two generic agents (M_HLCMA and M_OCA), they aim to replicate and abstract logical links that occur at the cyber level.

Based on that, and making usage of the first level of FRs directly related to the cyber system, it has been shaped the following DP (FR-01):

- DP-01 – *All the four generic agents are deployed according to the topology running at the native system. The deployment of the different agents intends to mimic the native system.*

The M_HLCMA aims to abstract the sub-systems where different components can collaborate. In this sense, the collaboration among the different resources can influence the execution of each other. Hence, the M_OCAs aims to receive all the information collected (raw data) and generated (knowledge generation) from all the M_CMAs and M_HLCMAs associated with that sub-system. All the data extracted and generated by the M_CMAs and M_HLCMAs is sent to the M_OCAs in order to be consumed and processed by external tools. Hence, the deployment of the M_OCAs will depend on the number of M_CMAs and M_HLCMAs as well as the number and type of tools which will receive the generated data.

One of the main focus of the proposed system is to mimic the available resources, and in this sense, the following DP arises (FR-011):

- DP-011 - *The M_CMA constantly checks the availability of the native system. With this information, the cyber system can mimic the availability of the physical resources.*

More than the connection or not of the controller that is managing the process with that resource, the cyber system receives information about the availability of the abstracted resources. Whenever a physical component is plugged, and any other information is sent to the cyber level, it is assumed that the physical resource is ready to produce. However, the native system can information to the cyber system announcing that a specific resource is no longer available due to maintenance issues or failures.

Hence, the cyber system can know at runtime, which resources are available for production and which resources are not available. This information is used by the cyber system to know which resources are useful for the extraction new knowledge. Furtherrmore, the availability of the physical resources is propagated to the M_OCA in order for the external tools to show or process that data too.

So, all the agents responsible for analysing data (M_CMA and M_HLCMA) contain a list of events that must be analysed, based on the data collected from the hardware or received from other agents (FR-02).

- DP-02 – *The extracted and analysed data at the different levels of processing is adjusted based on the plugged native system's components.*

The data collected by the agents allows them to understand the system's behaviour and retrieve useful values that can be used by other entities to show, model, and improve the system. The most common cases, of raw data that can be used to process and retrieve useful values, are related to transitions. A transition time describes the time that a resource need to move from a certain position to another. Figure 39 presents an example of a transition time extraction. The presented transition is performed by a clamp, where the first state represents the immediate state before starting the transition, the second one the transition itself and the last one the state immediately after the transition is finished, when the sensors indicate that the clamp is in the final position. The transition time, in this case, reflects the time needed by a clamp to close. Basically, the time passed since the clamp starts the movement from the initial state until it achieves the final state.

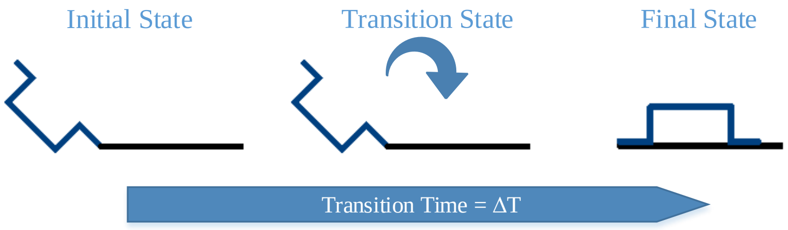


Figure 39 - Transition time

Transition time is a value that the companies are very interested in monitoring because, in general, an increase in this value probably indicates that maintenance is required for the correspondent resource. Using this time and continuously computing its average value makes it possible to recognise abnormal behaviour and avoid stoppages on the line and irreversible problems with the resources.

Another important value that can be extracted by processing raw data is the action time. Action time represents the delay verified since the input is given until the resource starts the movement. This value indicates the responsiveness of each component. Figure 40 illustrates in more detail an action time extraction, during which a clamp performs the same task as in the previous example.

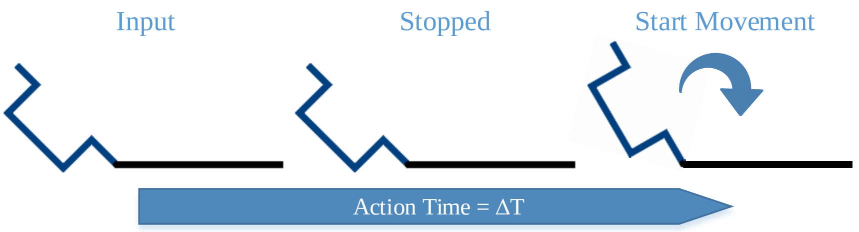


Figure 40 - Action time

In this case, the extraction of the action time is based on inputs and outputs that describe each state. The initial state is defined as the moment when the input is triggered, and the clamp is in the home position, and the final state is the moment when the clamp starts the movement leaving the home position. Hence, the action time is the time since the output is triggered by the resource stars the movement.

In order to understand all possible and useful events that should be discovered and computed by each agent, it is necessary that each agent has a knowledge base of all possible events that the abstracted component will perform. Therefore, each M_CMA and M_HLCMA loads a library which implements a method responsible for retrieving the rules that describe all events. These rules can be provided by any technology or storage entity, such as an XML file or database. Each event description must describe all conditions which need to be fulfilled to discover the event based on the raw data provided by the resource. An event description contains a list of all states verified during the execution of the event. Backing to the Figure 41 example the description of this event must contain three states, with all relevant input values that must be verified and the correct order.

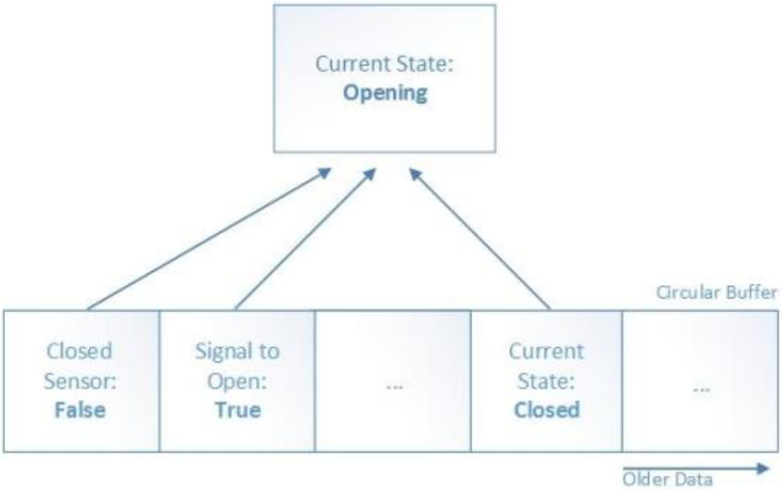


Figure 41 - State assessment example

Whenever a change is verified at the native system, the cyber system can deal with the new extracted and generated knowledge, through the adaptation of the states that can be computed and the data that can be retrieved from the native system. For instance when a new resource is plugged on unplugged, or a resource changes the state from available to unavailable or vice-versa.

Hence, whenever a change is detected and propagated, the M_HLCMAs will automatically update which knowledge is now available or unavailable to be generated, based on the resources that were added or removed. Hence, the M_HLCMAs can recognise a change and update the information needed to know which data must be processed and how (FR-021).

- DP-021 – *Whenever a M_CMA is deployed the M_HLCMA subscribed will receive raw and pre-processed data sent by the new deployed M_CMA.*

The M_HLCMA can get the update regarding the states that can be now computed, based on the library that stores the states used to generate new knowledge. On the other hand, when a M_CMA informs the M_HLCMA that the resource will leave the system, the M_HLCMA will update the rules. After this update, the M_HLCMA knows that the data retrieved from the M_CMA is no longer available.

The communication mechanism is based on FIPA compliant messages, presented in Figure 42.

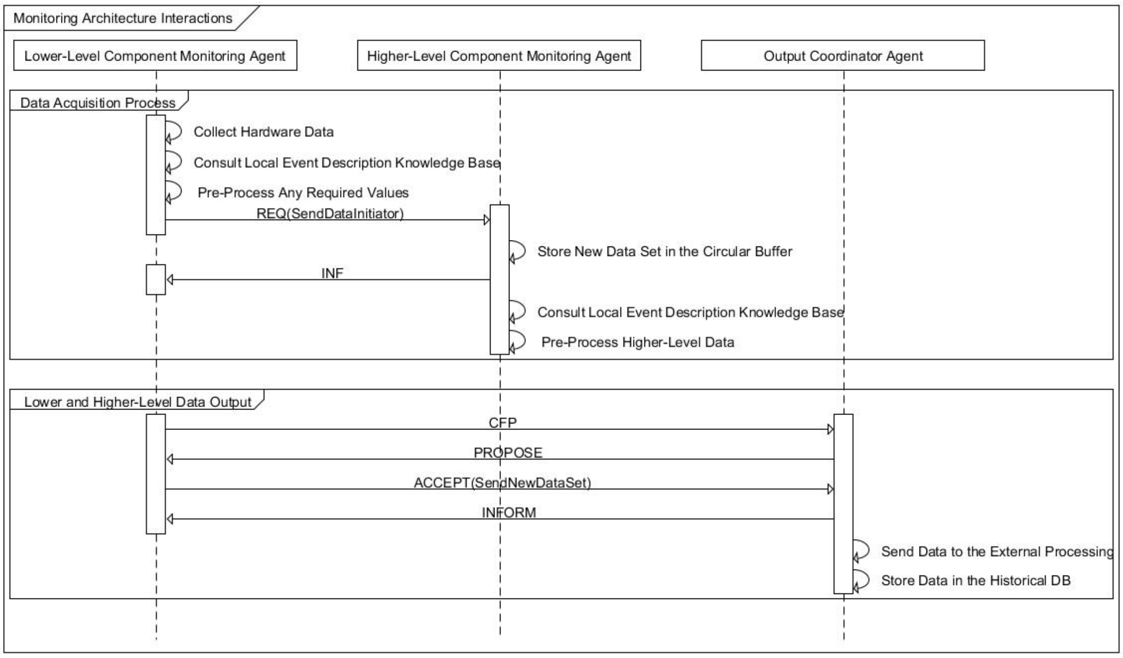


Figure 42 - Communication during data extraction and knowledge generation

The generation of new knowledge is performed by an algorithm implemented by the M_CMA and M_HLCMA. This data processing algorithm combines the information received from the hardware or other agents (M_CMAs or M_HLCMAs) and based on the rules that define possible states, action times and reaction times, computes the times which are relevant. These agents keep the data in a circular buffer that maintains the data extracted temporarily. This functionality allows the agents to have enough data to compute the states and events as well as avoid problems with too much information stored in memory.

The algorithm which allows the data processing is presented in Figure 43.

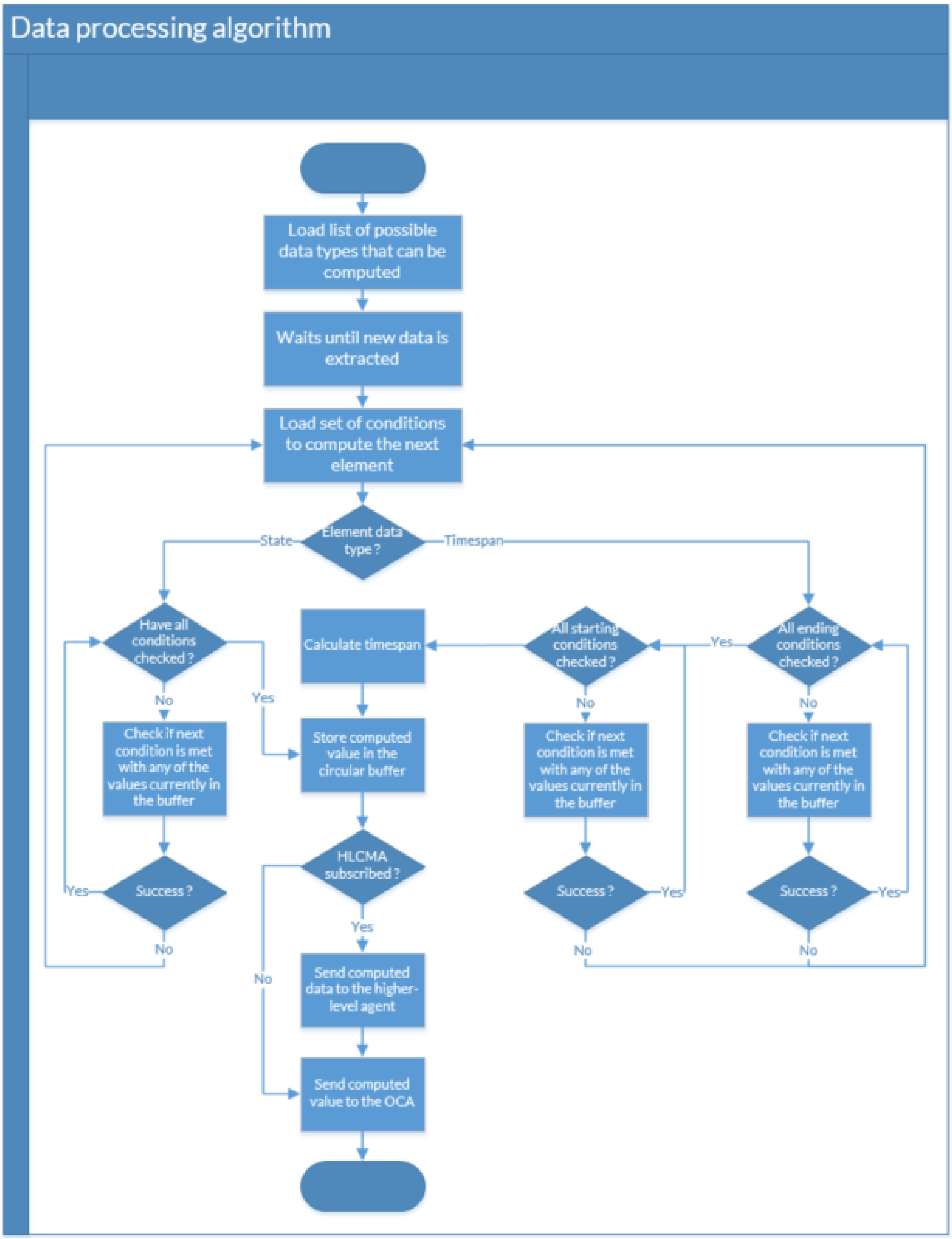


Figure 43 - Data processing algorithm

Since the rules that describe the possible events, can use raw data well as data generated from other agents, the M_HLCMA can compute new values based on the execution of different M_CMAs and M_HLCMAs. Hence, all the collaborative behaviours occurring in a sub-system can be analysed and new knowledge can be generated (FR-022).

- DP-022 -*All the sub-systems such as stations or cells are abstracted as M_HLCMA. The M_HLCMA compile and process the data retrieved and generated by the lower-level M_CMAs and M_HLCMA and process the data regarding collaborative tasks among the different M_CMAs and M_HLCMAs.*

In order to give flexibility to the cyber system and to adapt the behaviour accordingly to the available physical resources. The M_CMAs were designed based on a modular approach (FR-03).

- DP-03 – *All the physical components running at the native system are described within the cyber system as modules, with a cyber representation.*

Hence, the proposed cyber system was designed in such a way that each physical resource, when plugged to the native system, is abstracted by a cyber entity that represents a module which can retrieve information from the native system and generate knowledge based on the data retrieved from the physical resource.

More specifically, each plugged physical resources is abstracted by one M_CMA that is responsible for interacting with the hardware in order to retrieve the execution data from the native system (FR-031).

- DP-031 – *All the plugged physical components are abstracted and monitored by a M_CMA. The M_CMA is launched as soon as the physical component is detected.*

The M_CMA also knows all the movements and flags that can be useful for the analyse of the resource. The information that describes the movements of the resource is extracted from the events description's library.

Since the physical agent is the only link between the cyber system and the physical resource, and since the shop-floor contains resources which perform different tasks as well as resources provided by different suppliers, the following DP was defined (FR-032):

- DP-032 - *The M_CMA communicates with the native system through a standard interface, and all the M_CMAs describe the components generically. Hence, the M_CMAs can deal with heterogeneous components.*

Through the usage of a standard interface to split the physical agent from the control logic and hardware specific code, each physical agent (M_CMA) will be a generic entity that interfaces with the hardware through this interface. Hence, it guarantees that all the possible physical components (from different vendors, with different capabilities, control logic's technologies or standard) can be connected and abstracted by a similar physical agent. The difference between the M_CMAs are the event descriptions that can be used to generate knowledge and the library used to interface the hardware. All the other aspects, such as the internal description of the components, communication with the other different cyber entities and so forth are generic and in this sense, the platform can generically and without any programming effort dealing with heterogeneous modules.

The DP-0 says that the performance of the existent native system must not be degraded. Since the physical agent is the only connection between the cyber and the native system, the modules were designed based on the following DP (FR-033):

- DP-033 - *The M_CMA, responsible for extracting the raw data from the native system does not interfere with the existent control logic. The M_CMA uniquely extract values from variables and parameters.*

The proposed solution only extracts data and does not write or change any parameter or memory position in the native system. Moreover, the proposed M_CMA can extract the data from the native system in two distinct ways. Firstly, the M_CMA can periodically verify a specific memory position or parameter and retrieve this value from the hardware. The other possibility is to subscribe a list of variables, which the M_CMA wants to collect whenever some variable changes the value. For instance, this is particularly important for actuators or sensors used to detect a part.

Since the extraction of information does not require any change of the control logic, the solution guarantees that deployment of the CPPS will not degrade the existent execution.

All the information necessary to handle the module is stored and managed by the M_CMA. Whenever a new M_CMA comes alive, the M_CMA knows which library must be instantiated to communicate with the hardware, which events' descriptions must be handled and, if existent, for which M_HLCMA it must share the information (FR-034).

- DP-034 - *The M_CMA is responsible to abstract the component it is responsible for monitoring. To do that, the M_CMA needs to know critical* *information such as the sub-system where the component is working or the events that can occur.*

Appling the previously presented DPs, a cyber module can be described as presented in Figure 46.

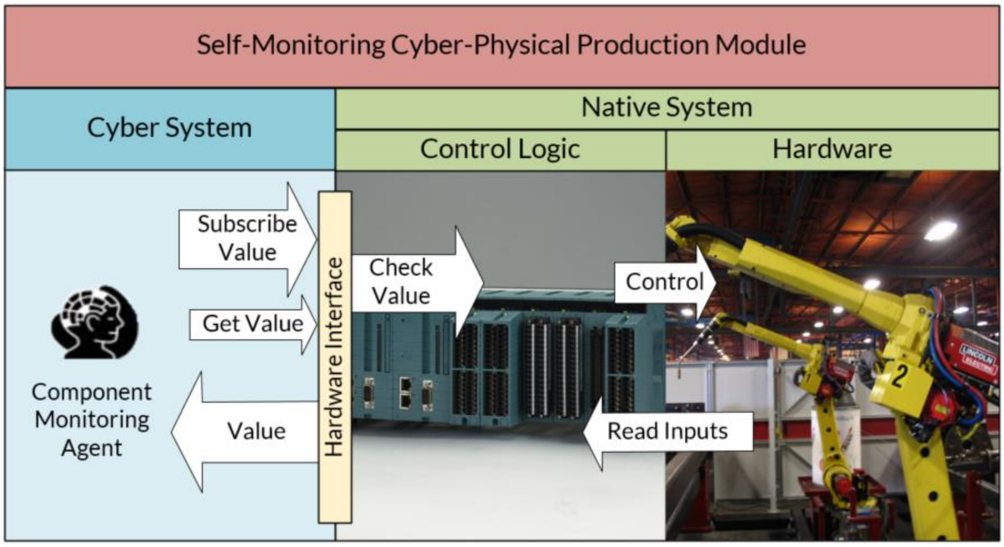


Figure 44 – Self-monitoring cyber-physical production module

The most critical identified requirement is the need for the native system to continue the execution without the cyber system, avoiding external dependencies. So, one of the most critical and imperative DPs is (FR-04):

- DP-04 - *The cyber system runs upon the native system without creating dependencies. The cyber system can be deployed and removed, and the native system continues running without any interference.*

The execution of the native system does not require the deployment of the cyber system. Since the M_CMAs (the only cyber entity interacting with the hardware) does not require any change from the existent control logic, the execution of the native system is not compromised by the existence or not of the cyber system (FR-042).

- DP-042 - *The CMA does not control the native system.*

To avoid the dependencies and also to avoid performance's degradation, the M_CMA and the native system communicates through a method that works in a transactional and atomic way. All the communications performed, through the library responsible for integrating the two systems (cyber and native system), are transactional and atomic (FR-043).

- DP-043 – *To avoid the dependencies, whenever information needs to be sent from the cyber system to the native system and vice-versa, that exchange of information occurs through a method that delivers the information atomically and without any communication channel.*

Whenever information needs to flow from the M_CMA that wants to check a specific value from the native system, the M_CMA calls a method that checks the value and retrieves the value, without time and channel constraints. On the other hand, when the native system wants to deliver a value subscribed by the M_CMA, a method is called, and the value is delivered to the agent without dependencies also.

The communications between the two systems are fully asynchronous and do not depend on each other. When the M_CMA is retrieving a value from the native system, the native system can also trigger a value that changed on the other way around (FR-044).

- DP-044 – *The M_CMA and the native system can call the method to deliver information to and from the two systems simultaneously.*

This characteristic guarantees that the delivering of information does not degrade the performance of the native system and that the data is delivered as much fast possible without any delays regarding the synchronisation of methods or channels.

The primary functionality of the proposed cyber system is the ability to extract data and generate knowledge to be used by humans. Hence, it is particularly important to guarantee that the extracted and generated knowledge is properly shown to the humans. Hence, the infrastructure was designed to share the information with external tools developed to show the data and the cyber system properly (FR-05).

- DP-05 - *The proposed cyber systems can provide information to external databases and tools to visualise and process the generated knowledge.*

The developed external tools must guarantee all the regulations already defined for the native system (FR-052).

- DP-052 – *The cyber system does not change the native system's behaviour, and hence the system fulfils the regulations already fulfilled by the native system.*

The safety, security and other regulations are fulfilled by the proposed cyber system since the cyber system maintains all the regulations verified by the native system. The proposed system only runs upon the existent system and does not change any parameter or characteristic of the native system that could create issues regarding the existent regulations.

Through the information sent by the cloud of M_OCAs and the event description file, it is possible to design external visualisation tools. Using the services exposed by the M_OCAs, it is possible to write the information in historical databases or send the data directly to external visualisation tools, such as local, web-based or mobile tools (FR-053).

- DP-053 – *The M_OCAs allow the cyber system to send information to external databases and tools which will use the generated knowledge to show it and create more knowledge.*

In these external visualisation tools, it is possible to visualise raw data, information regarding the overall performance of the system, performance of specific resources, failures and other issues. The specification and the presented data is defined by the customer and by the people that will consult the visualisation tool.

The possibility for the personnel to consult and visualise the cyber system execution is a key element in the development of minimally invasive solutions. However, the integration with the hardware and external software tools is always a big challenge: The proposed cyber system offers a standard interface that must be used by the systems' integrators to integrate each module. Another interface allows the integration with external tools to visualise the system and the knowledge generated by the cyber system (FR-06).

- DP-06 - *The cyber system shall provide standard interfaces that enable the interaction with the native system as well as other systems (software tools) operating collaboratively with it.*

The data exchanged among the hardware, cyber system and external software tools, is sent in a defined format. The MonitoredSystemValue class, presented in Figure 45, describes how the data extracted and processed must be stored and shared.

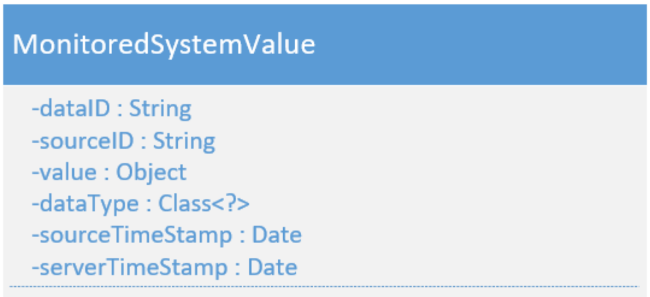


Figure 45 - MonitoredSystemValue class

As already mentioned, the communication between each physical agent and the hardware that the agent is abstracting exclusively occurs through the standard interface, called HAL. Each M_CMA instantiates a specific library that fulfils the HAL and allows the M_CMA to call the defined methods. This method will allow the M_CMA to extract data from the hardware or subscribe new values that need to be sent to the M_CMA. Vice-versa, when the native system wants to communicate that a subscribed value changed, a generic method must be called through the HAL by the native system. The HAL is equal for all the instantiated M_CMAs. Hence, all the communications with the physical components use a similar HAL (FR-061).

- DP-061 – *A HAL has been designed to allow any M_CMA to communicate with any possible physical resource, through generic methods, such as get a value or subscribe a value, among others.*

The following methods, defined in the HAL, must be implemented by the systems' integrators when a new resource needs to be integrated:

- intHWConnection(Component, SubscriptionList): This method must be called at the beginning of the execution to check if the communication with the hardware is possible and if it is working correctly. Moreover, SubscriptionList contains the values that must be sent to the M_CMA whenever they change;
- closeHWConnection(): This method must be called by the M_CMA when the resource is unplugged;
- readHWValue(id): This method must be called by the M_CMA to know the current value of a sensor or actuator.

The interaction with external tools is allowed through the methods exposed by the Data Output Library, implemented by the M_OCA. The implementation of this library, one for each M_OCA, allows the M_OCAs to send this data to external tools or databases. The implementation of these libraries allows the external tools to receive information about the current situation, such as current topology and values regarding the performance of the system (FR-062).

- DP-062 – *A standard interface was designed to be used by M_OCAs to send the generated knowledge to external endpoints.*

The usage of historical databases is imperative because industry continuously needs to consult historical data in order to understand unexpected behaviours and the performance of the system. This data can be used either by the external entity or by other entities such as a Human Machine Interface.

The usage of an external remote entity, running in powerful machines with capacity to store any amount of data provided by the system, allows it the capacity to model and retrieve more information from raw and pre-processed data. During execution, the external entity is capable of continuously computing all received data, modelling the behaviour of the system and triggering events that can result in important alerts, avoiding line stoppages and allowing the operators to perform preventive and predictive maintenance.

The following method, defined in this abstraction layer, must be implemented by the systems' integrators when a new M_OCA is instantiated to share the extracted and generated knowledge with external tools:

  - sendOutput(MonitoredSystemValue): This is called by the M_OCA whenever a new value is received from the M_HLCMAs and M_CMAs.

A scheme of the defined DPs can be found in Figure 46.

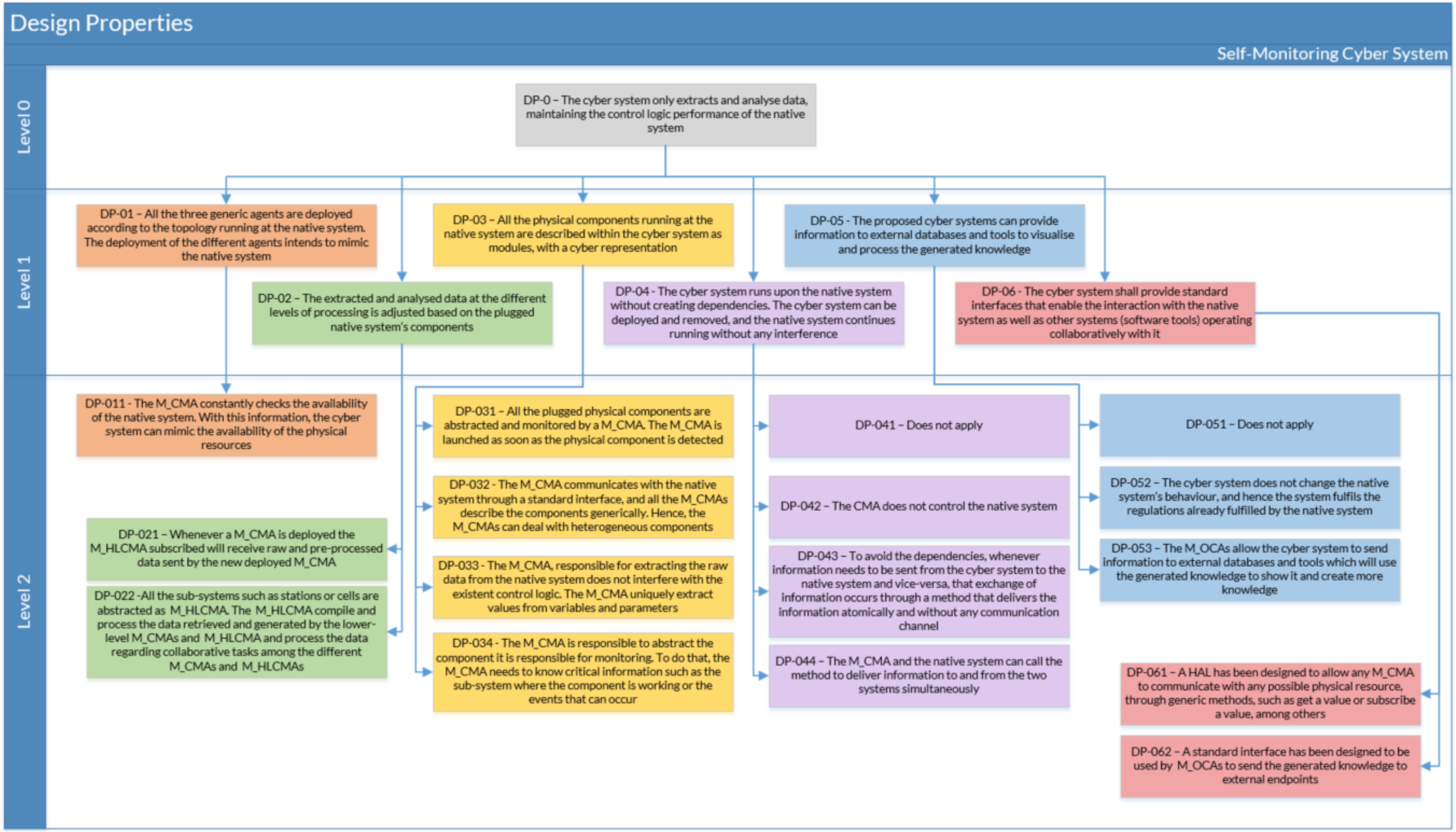


Figure 46 - Design properties for design a reconfigurable cyber system

4.1.3. Self-Healing Agent based CPPS

The proposed distributed approach opens the possibility to perform a different type of diagnosis and error recovery. With the different modules presented in the production environment, it is possible for each module to diagnose itself. Self-diagnosis is an important and efficient solution to face disturbances and failures during execution while not decreasing the performance of the other components since the problems are solved locally.

The proposed solution uses Artificial Immune Systems (AIS) to detect and recover from execution errors, which is believed to be an effective solution. This type of systems is based on bio-inspired algorithms which aim at replicating the functioning of the Human Immune System, used to fight diseases. This paradigm is mainly supported by four different algorithms which aim at mimicking the functioning of the Human Immune System.

Hence, the first DP which guided this cyber system is (FR-0):

- DP-0 - *The native system does not degrade the native system's execution. However, it improves the errors' recovery, improving some KPIs.*

In Figure 47, it is represented a conceptual overview of the proposed multiagent architecture. The proposed architecture has two main responsibilities. The Diagnosis and Recovery Layer is composed of agents which are responsible for detecting failures or disturbances during the execution, understand what kind of problem was issued and apply a cure, once known. The Evolution Layer is composed of the Cure Provider Cloud. The agents inside it are responsible for providing cures for each problem discovered in the Diagnosis and Recovery Layer and for trying to create new possible cures, based on the known ones.

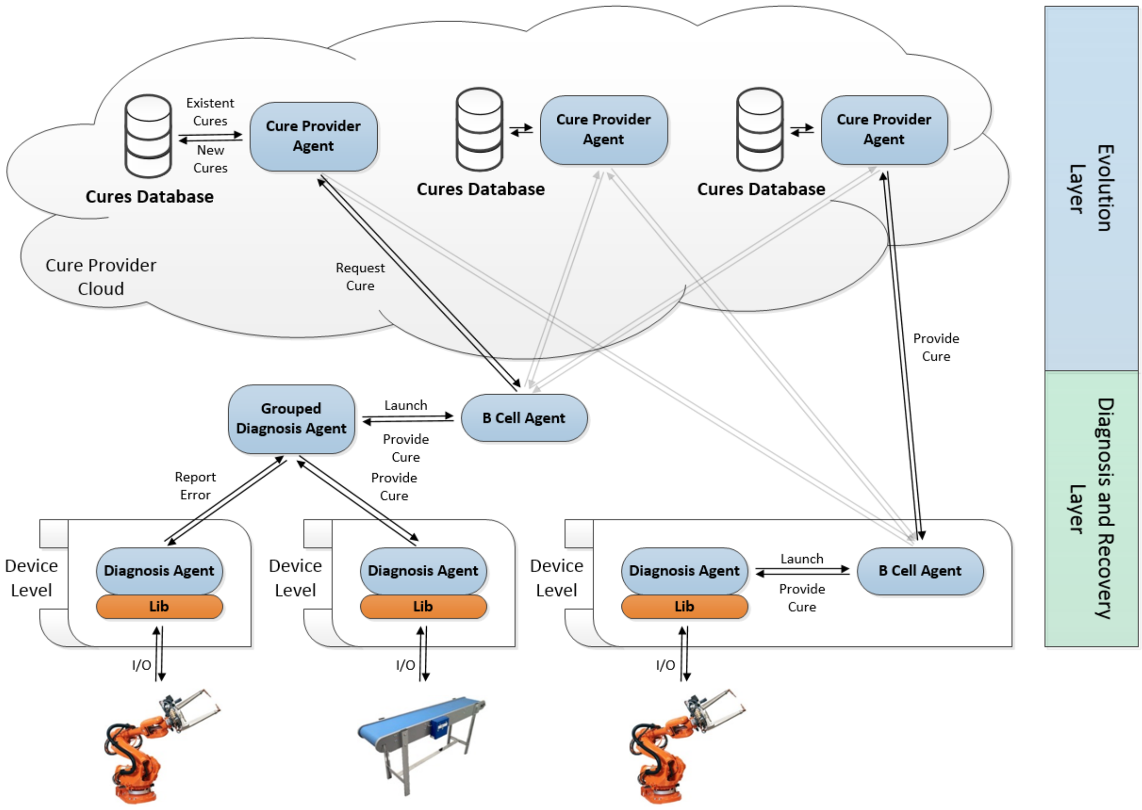


Figure 47 - Self-healing agent based CPPS

The proposed framework is composed of four generic agents with different responsibilities. Some agents should abstract the hardware and others are more related to higher level actions, such as a combination of different sources of information:

- Cure Provider Agent: The Cure Provider Agents (H_CPA) constitute a cloud that covers the totality of the system, meaning this cloud is available to provide cures to all the lower level entities. Each entity has an associated database where all the known possible cures are stored. Hence, when a lower level agent asks for a possible cure to solve a given problem, the H_CPA queries the database for possible cures to the problem in hands. If there is no solution in its database, the H_CPA as the capability to create new cures based on the existent ones through Genetic Algorithms (GA).

- Grouped Diagnosis Agent: The Grouped Diagnosis Agent (H_GDA) constitutes a group of physical devices that work together and consequently influence the execution of each other. These entities are capable of collecting the malfunctions of the resources and understanding if the errors are correlated or not. When the H_DAs are grouped, the resolution of the errors is the responsibility of this higher-level entity. Therefore, when the H_GDA agent does not have a cure for a specific problem, a B Cell Agent is launched to find new possible cures.

- Diagnosis Agent: The Diagnosis Agent (H_DA) is the lowest level entity of the entire system. It is responsible for the diagnosing of a physical resource, such as a robot or conveyor belt. During the execution, the H_DA constantly verifies the methods of a specific resource trying to find errors and failures. Once a malfunction is detected, the H_DA verifies if there are available cures for the detected error; if so, the H_DA performs the cure immediately; if not, the H_DA runs the AIS algorithm to define what kind of B Cell Agent should be launched.

- B Cell Agent – The B Cell Agent (H_BCA) emulates a real B Cell. Hence, when a malfunction is detected, and neither the H_GDA nor the H_DA has cures for the error, a H_BCA is created to find new possible cures. The creation of these entities is defined by the AIS algorithm running inside the H_GDA and H_DA, accordingly to the error that needs to be solved.

The designed multi-agent environment can use any AIS algorithm as error recovering mechanism. In this sense, it is possible to embed at the H_DA and H_GDA one of the algorithms that mimic the immune system.

AIS is a diverse area of research that attempts to bridge the divide between immunology and engineering. It is developed through the application of mathematical and computational modelling to immunology. These immunologic models are abstracted using algorithm (and system) design and implemented in the context of engineering. For this experiment four algorithms were studied: Negative Selection, Clonal Selection, Immune Network Theory and Danger Theory.

- Negative Selection: Its purpose is that of allowing some degree of tolerance for self-cells (those normally present in the organism). Dealing with the Immune System's ability to detect unknown antigens (harmful cells) without prejudicing its cells. The Immune System's generated B Cells, which fight these antigens, are formed of a pseudorandom genetic rearrange. Those of which that react against the antigens are used to destroy it and replicated in the organism as matured cells (Zuccolotto et al., 2015).

- Clonal Selection: The main idea behind this algorithm is that only the B Cells that recognise the antigen will thrive and replicate. This principle describes the basic characteristic of an immunologic response to an antigen caused stimulus. This algorithm's main characteristics may be enumerated as follows:
    - New cells are copies of those they derive from and are then subjected to a high rate mutation mechanism (somatic hypermutation).

- Procedural elimination of the new cells that, after mutation, endanger the survival of the non-prejudicial cells for the organism (self).
- Further cloning and mutation of the cloned cells that respond positively to the antigen.

This mechanism allows for a faster response to the antigen (El-Sharkh, 2014).

- Immune Network Theory and Danger Theory: The Immune Network Theory algorithm is based on the assumption that the Immune System maintains a regulated network of interconnected B Cells with the purpose of easing antigen detection. These cells stimulate and suppress each other with the ultimate purpose of providing stability. The connection between two B Cells is proportional to their affinity to each other (Zhong and Zhang, 2012).
- The Danger Theory algorithm is based on the idea that the Immune System is more preoccupied with the entities that damage it than with those that are strange to it. Hence, this algorithm assumes that the Immune System is activated by dangers signals emitted by the damaged cells (affected by the antigens or with mechanical damage). Therefore, the main challenge of this algorithm is to distinguish between the real and the fake signals. Whenever a signal is emitted, the antigen-present cells are activated. Hence, this would stimulate the B Cells (Yin et al., 2012).

To effectively perform the error recovering and to increase the solution's effectiveness, the multi-agent environment has been designed to mimic the native system (FR-01).

- DP-01 - *The H_DAs, H_GDAs and CDAs are instantiated and launched accordingly to the native system's.*

The usage of lower level agents (DAs) and the possibility to have different layers of H_GDAs allows the cyber system to mimic any native system, no matter how many physical resources are running or which type of topology is being used.

The H_GDAs allow the cyber system to mimic the different layers and interactions among the different H_DAs and in some cases among others H_GDAs. A production station must be mimicked as a H_GDA and all the resources within the production station must be abstracted as H_DAs, as shown in Figure 48.

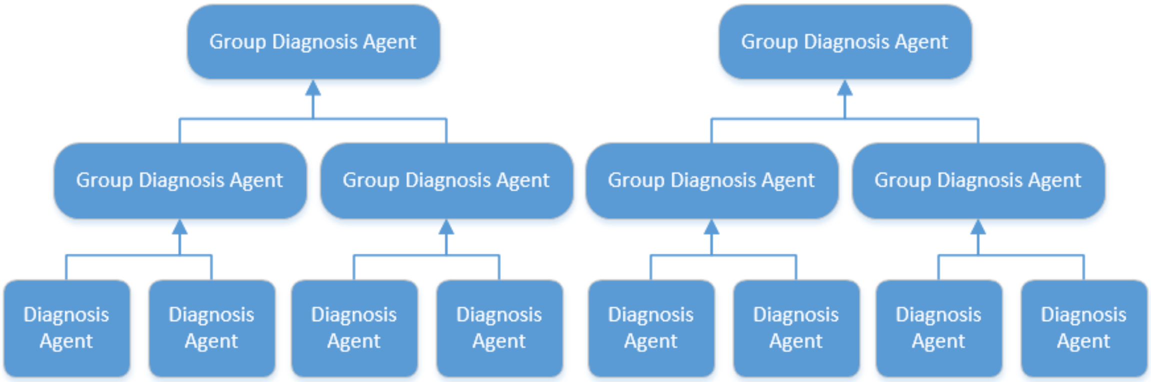


Figure 48 -Self-healing multi-agent system topology

Hence, whenever the topology changes, the only requisite is to update the agents that abstract each physical component or cluster of components or groups. As far as the topology is updated the cyber system can mimic the native system.

Regarding the lowest level of the cyber system, the H_DAs must mimic the availability of the production resources (FR-011).

- DP-011 -*Through the detection of the plugged resources and the information retrieved from the native system, the cyber system can update the availability of each resource.*

The H_DAs abstract each physical resource. Abstracting each resource by a dedicated agent allows the cyber system to mimic the physical resources running on the native system by knowing which H_DAs are running at that time. The H_DA execution is shown in Figure 49.

•

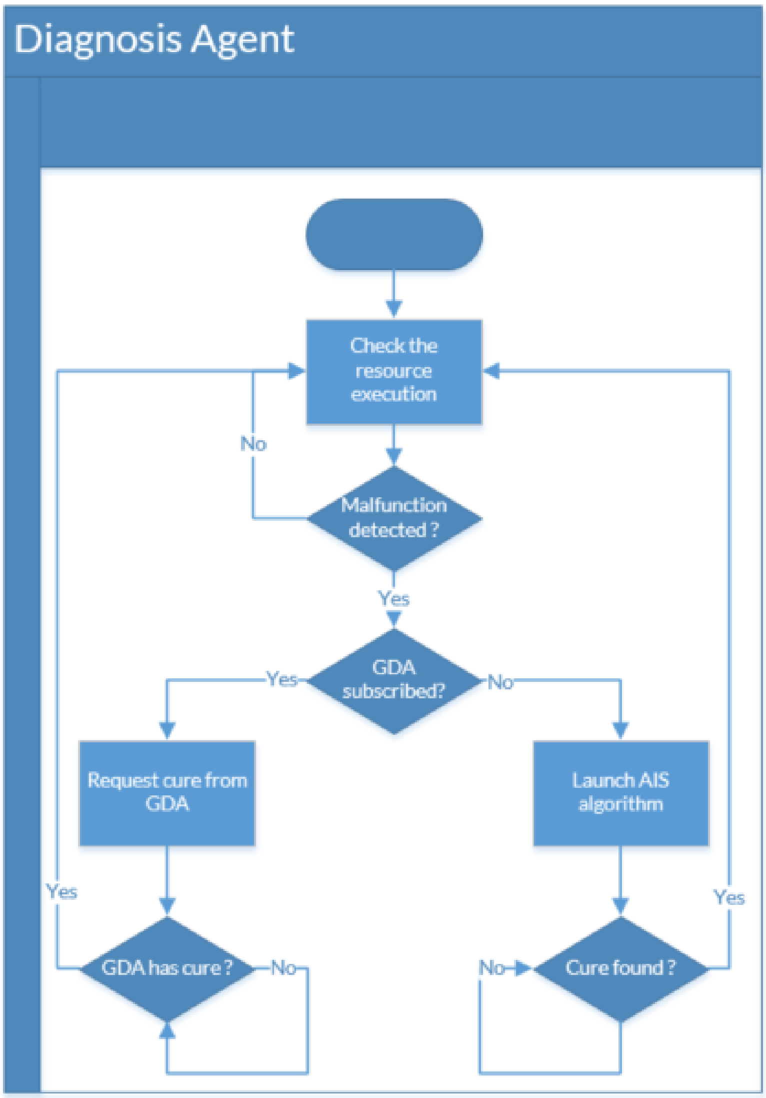


Figure 49 - Diagnosis Agent execution

Whenever a new component is added to the native system, a H_DA which abstracts that physical component must be launched. The opposite behaviour occurs when a resource is unplugged. Giving to the system, the possibility to plug and unplug components in runtime guarantees ease with the integration and configuration as well as the mimic among the native system and the cyber system.

Hence, whenever one of the previous cases occur, the cyber system is updated in order for the different cyber entities update the system's state.

- DP-02 - *The cyber system, whenever the native system changes can adapt the cyber abstraction through the adaptation of the agents which abstracts the native system (DA, H_GDA).*

That being said, it is necessary for the cyber system to share the information among the agents which should receive the updates. So, whenever a new H_DA is plugged, it knows, from the information received during the launching process, which H_GDA is responsible for the area where the H_DA is operating, if any. With this information, the H_DA, immediately after coming alive, informs the H_GDA about the new component. Hence, the H-GDA, when analysing the future malfunctions, will take into account the existent of this new component at the sub-system abstracted by itself (FR-021).

- DP-021 - *The communication among the H_DAs and H_GDAs through FIPA compliant messages allows the cyber system to adapt the cyber system to process more or fewer malfunctions according to the availability or not of the resources.*

The subscription of a higher-level H_GDA allows it to receive all the errors detected and reported by the H_DAs and H_GDAs working at the abstracted sub-system. The H_GDA aims to function as an abstraction entity where the collaboration of the different resources or sub-systems can be analysed. That abstraction can provide solutions for the problems, which for some reason, occur based on the collaboration. For instance, problems that can result from a propagation of an error or whenever is necessary to analyse the errors provided by the different entities within the H_GDA (H_DAs and H_GDAs) (FR-022).

- DP-022 -*The H_GDAs work a cluster of H_DAs or other H_GDAs that can work collaboratively, in a specific sub-system. In this sense, the H_GDAs intends to receive the malfunction from the lower agents to try to understand malfunctions' dependencies or roots.*

Whenever a problem is detected, that information is sent to the H_GDA in order for this to understand if that error influences errors in future activities, or if this error is a propagation of previous errors. Based on that, the H_GDA can try to solve errors that could be not solved at the physical resource level due to the nature of the error, as presented in Figure 50.

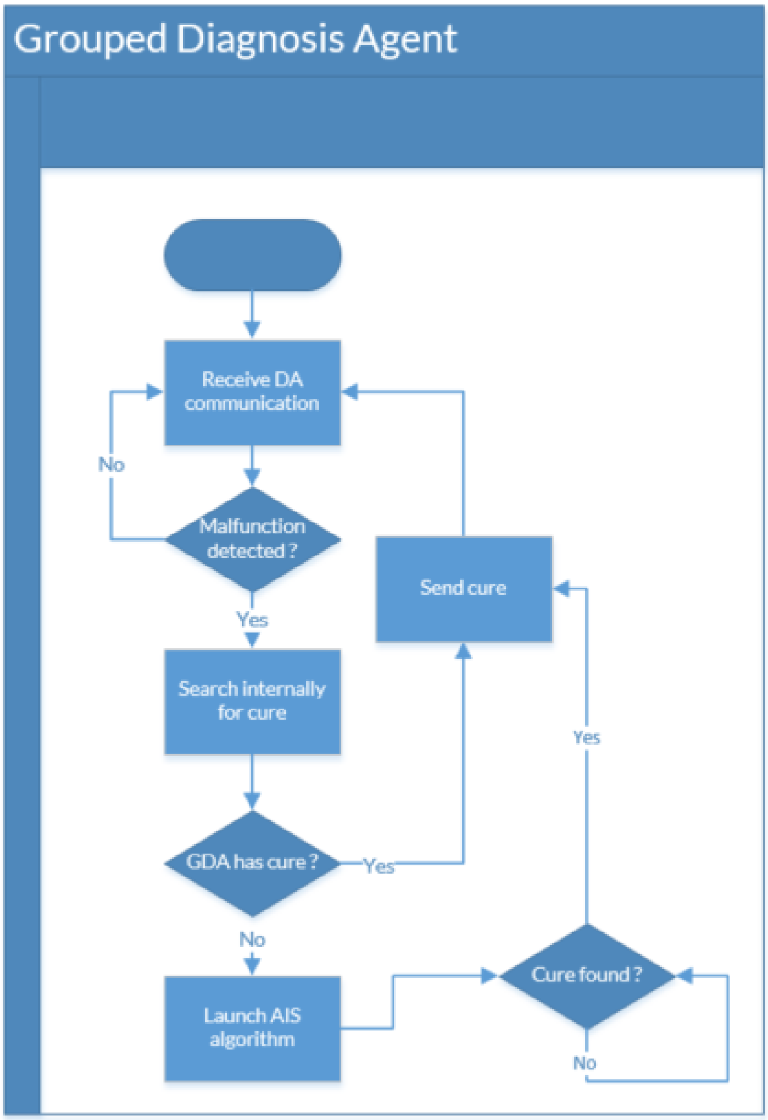


Figure 50 - Grouped Diagnosis Agent execution

During execution, the framework has different stages and functionalities. Figure 51 presents the steps performed by each entity when a H_DA detects a malfunction. First of all, when a H_DA detects a malfunction, it verifies if it has a possible cure for this malfunction; if so, it performs a cure based on the locally stored cure. Another possibility is when a H_DA has no local cures to solve the malfunction and has no group (H_DA does not subscribe any H_GDA). In this case, the H_DA launches new H_BCAs, based on the AIS algorithm results. The H_BCAs are responsible for finding a new possible cure for the founded error.

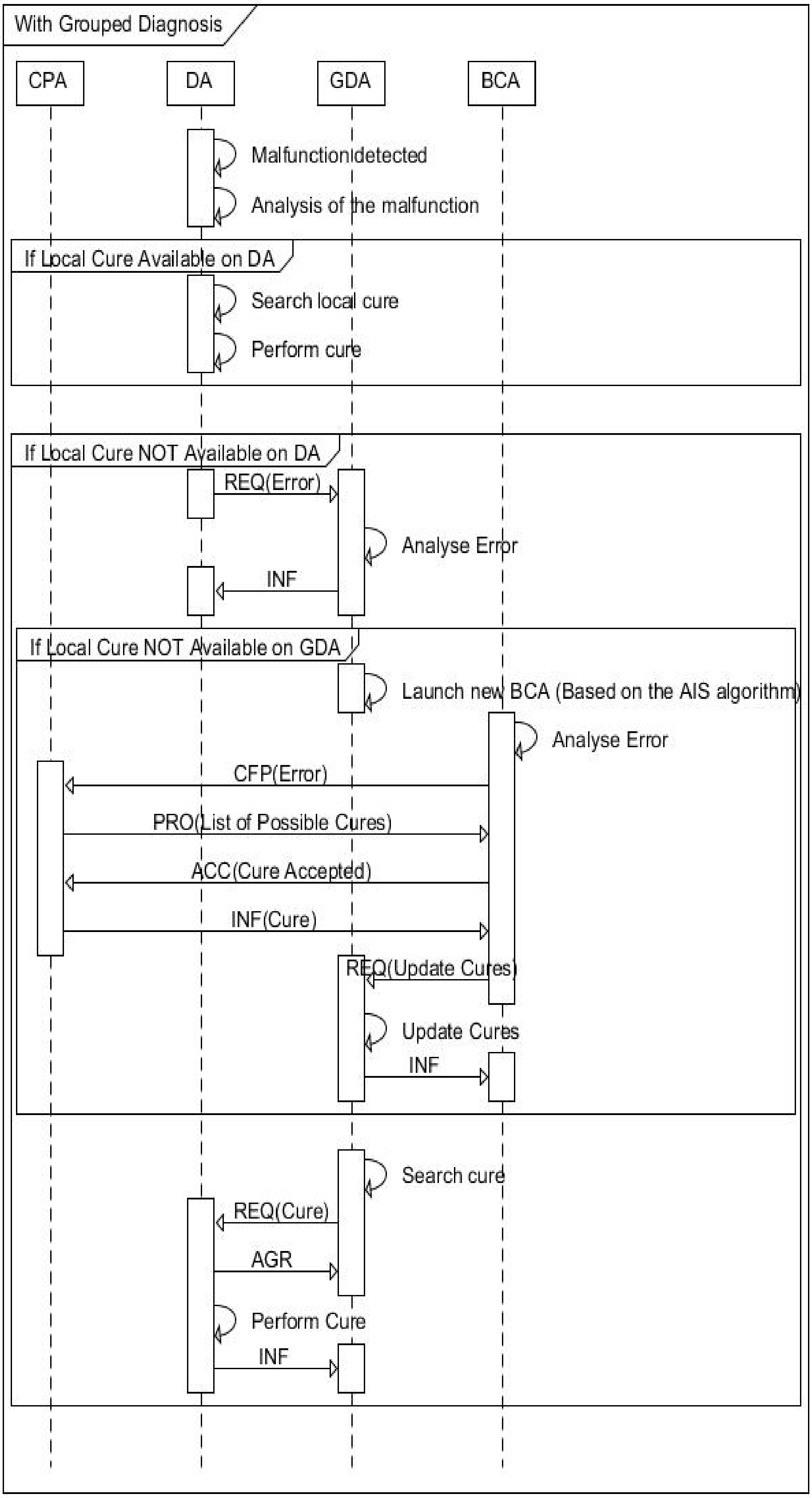


Figure 51 - Self-healing cyber system execution

When a H_DA is grouped to a H_GDA, the H_DA sends the diagnosed error to the H_GDA. From now on the H_GDA is the one responsible for managing the error. The first step is to understand if the H_GDA already has cures for the error. If so, It sends the cure to the H_DA and immediately solves the problem. On the other hand, if a local cure is not available, the H_GDA launches a set of new H_BCAs (based on the AIS algorithm result) which are responsible for providing a new cure.

When a new H_BCA is initiated, its responsibility is to provide a new cure that could solve the malfunction. To do so, it analyses the received error and the cure generated by the AIS algorithm, if the cure provided by the algorithm can be applied, the H_BCA sends the cure to the H_GDA. If not, it starts a negotiation process with the H_CPAs in order to find possible cures, stored in the cloud environment.

The proposed cyber system was designed in such a way that each physical resource, when plugged into the native system is abstracted by a cyber entity, the H_DA. Each H_DA represents a module that can retrieve information from the native system, such as the production steps and the detection of errors during the execution (FR-03).

- DP-03 - *The abstraction of the physical resources is done to the definition of modules which encapsulates the physical component with cyber representation, in this case through a H_DA.*

Each module must be abstracted by a physical resource. Hence, all the plugged physical resources are abstracted by one H_DA that is responsible for interfacing with the hardware in order to verify the execution and verifies the detected errors as well as to apply apply cures (FR-031).

- DP-031 - *Any plugged and running physical resource has a H_DA abstracting the execution and functionalities of the cyber system.*

The H_DA continually verifies the information provided by the control logic in order to detect malfunctions and problems during the execution. When the H_DA detects a malfunction, it must verify if any recovery routine is known in the local database, if yes, the H_DA must apply the cure and see the result. If the cure was not succeeded and the H_DA is connected to a H_GDA, the H_DA must send the detected error to the H_GDA, to find new solutions.

Through the usage of a standard interface to split the physical agent from the control logic and hardware specific code, each H_DA is a generic entity that interfaces with the hardware through this interface. Hence, it guarantees that all the possible physical components (from different vendors, with different capabilities, control logic's technologies or standard) can be connected and abstracted by similar H_Das (FR-032).

- DP-032 - *The H_DA is a generic agent that implements a standard interface that allows the communication with different heterogeneous components. In the cyber system the H_DA is a generic entity, and hence the cyber system can deal with any components.*

The difference between the H_DAs is the internal information regarding the cures that can be applied to the abstracted component for some known errors (known from previous experiences), the information that describes the module, and the library instantiated to communicate with the native system.

All the other aspects, such as the internal description of the components or communication with the other different cyber entities are generic and in this sense, the platform can generically and without any programming effort dealing with heterogeneous modules.

Through the previously presented interface, the H_DA can communicate with the physical resource. The H_DA extracts the data which describes the execution of the native system and writes the cures, which will basically trigger some recovery routines or adjustment of some parameters.

Since the extraction of information and the application of the cure do not require any change of the control logic, the solution guarantees that deployment of the cyber system will not degrade the existent execution (FR-022).

- DP-033 - *The interfacing of the cyber system is through a reparameterization of the existing control logic. Hence, the existing control logic stills responsible for controlling and executing the process without external interferences and dependencies.*

In order to allow the H_DA to interface with the hardware and at the same time share the information that describes the physical resource within the cyber system, the H_DA must manage all the information necessary to describe and interface the resource (FR-034).

- DP-034 - *The H_DA contains all the information regarding the physical component, such as known cures, sub-system, etc.*

All the information needed to describe and operate the physical agent, in this case, the H_DA, is stored and handled by itself. Whenever a new physical component appears, a H_DA is launched with the name of the library that must be instantiated in order to allow the communication among the agent and the hardware, the description of the physical resource and the information of the H_GDA responsible for the sub-system where the H_DA is operating.

During the execution and with the experience acquired during that execution the H_DA keeps and updates a list of errors detected and the cures applied to solve that errors, if efficient. It allows the H_DA to, with the evolution of the system, avoid interactions with the higher level, and delays if the error could be quickly solved, improving the overall performance of the native system.

Appling the previously presented DPs, a cyber module can be described as presented in Figure 52.

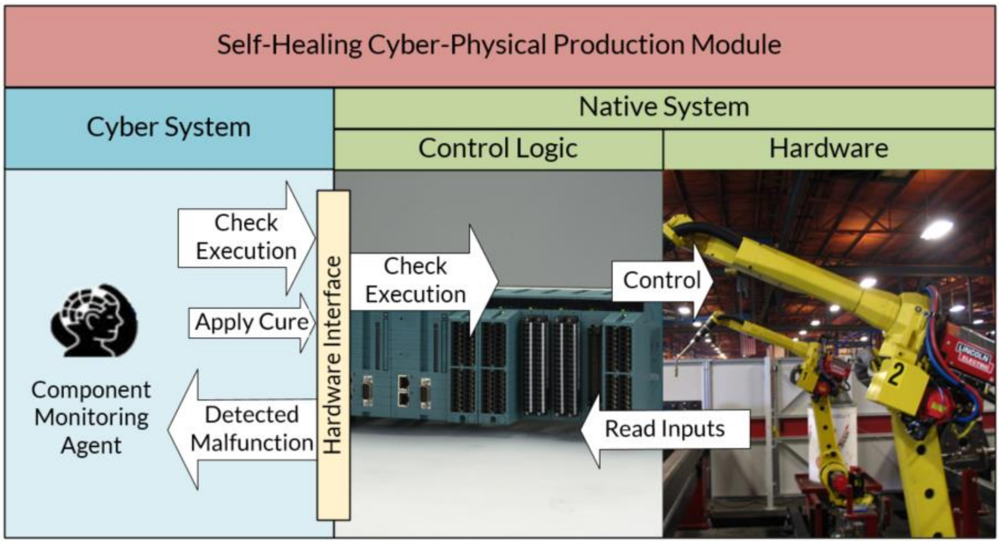


Figure 52 – Self-healing cyber-physical production module

The necessity for the native system run without the cyber system and the obligation to avoid dependencies forced the following DP (FR-04).

- DP-04 - *The proposed solution runs upon the existent native system and whenever the solution is removed or the connection from the native system to the cyber system and vice-versa, the existent native system stills running.*

The execution of the native system does not require the deployment of the cyber system. If the cyber system stops the native system must continue with the execution without any problem.

Depending on the definition of reconfiguration, the proposed solution can be defined as a reconfigurable cyber system or not. The cyber system does not reconfigure the behaviour of the system, but at the same time, the cyber system applies cures that can result in sub-routines capable of recovering from specific errors (FR-041).

- DP-041 - *The H_DA only can influence the native system's behaviour applying cures. The control of the process and hardware still being executed by the existing control logic.*

However, even keeping the possibility of reconfiguration, the proposed solution does not directly control the hardware. Hence, this FR can be considered fulfilled since the application of the cures is made through the reparametrisation within the control logic, which will result in different routines or parameters, executed temporarily, in order to re-cover from the detected malfunction.

The standard control logic performs the control independently of the existence of the cyber system. Since the proposed solution only changes some parameters, that will lead to the error recovery. However, the error recovery, from the control perspective is performed by the existent control logic, taking into account the new variables and param-eters' values (FR-042).

- DP-042 - *The H_DA applies the cure through a set of variables', and parameters' values change, without changing the existent control logic or directly controlling the hardware.*

To avoid the dependencies and also to avoid performance's degradation the H_DA and the native system communicates through a method that works in a transactional and atomic way. All the communications through this library are transactional and atomic.

Whenever a control step is performed and whenever necessary, the native system sends that information to the H_DA. That method is called, and atomically the information is sent to the H_DA without keeping a communication channel. Similarly, in the previous cases, whenever a H_DA intends to apply a cure, an atomic communication is done, where the H_DA sent to the native system the cure (parameters' values) (FR-043).

- DP-043 - *The communication between the H_DA and the native system occurs whenever one of the parts call a method defined within the standard interface to send information from one system to another.*

The communications between the two systems are fully asynchronous and do not depend on each other. That being said, when the H_DA is receiving the information from the native system, it can also apply a cure simultaneously (FR-044).

- DP-044 - *The methods defined in the standard interfaces are not blocking. Hence, the communication between the two systems can simultaneously occur.*

This characteristic guarantees that the delivering of information does not degrade the native system's performance and that the application of a cure does not unhallow the native system to send information to the cyber system.

In order to efficiently integrate the different physical resources and quickly implement the previously presented modules, the proposed cyber system offers a standard interface that must be used by the systems' integrators to integrate each module (FR-06).

- DP-06 - *The proposed self-healing system allows easy integration of new modules and components through the use of standard interfaces.*

As already mentioned, the communication between each physical agent and the hardware that the agent is abstracting occurs exclusively through the interface, called HAL (FR-061).

- DP-061 – *The communication between the H_DA and the physical resource occurs through the HAL. The HAL defines a set of methods that must be used by the H_DA to extract the execution's steps by the hardware and to apply the cures.*

Each H_DA instantiates a specific library that fulfils the HAL and allows the exchange of information between the physical agent and the hardware.

The HAL is equal for all the instantiated H_DAs. So, all the communications with the physical components use a similar HAL.

The following methods, defined in the HAL, must be implemented by the systems' integrators when a new resource needs to be integrated:

- intHWConnection(Component): This method must be called at the beginning of the execution to check if the communication with the hardware is possible and if it is working correctly;
- closeHWConnection(): This method must be called by the H_DA when the resource is unplugged;
- readState(): This method must be called by the H_DA to know the current value of the physical resource execution.
- sendState(Cure): This method must be called by the H_DA when an error is detected. This method receives as parameter the cure to be applied.

A scheme of the defined DPs can be found in Figure 53.

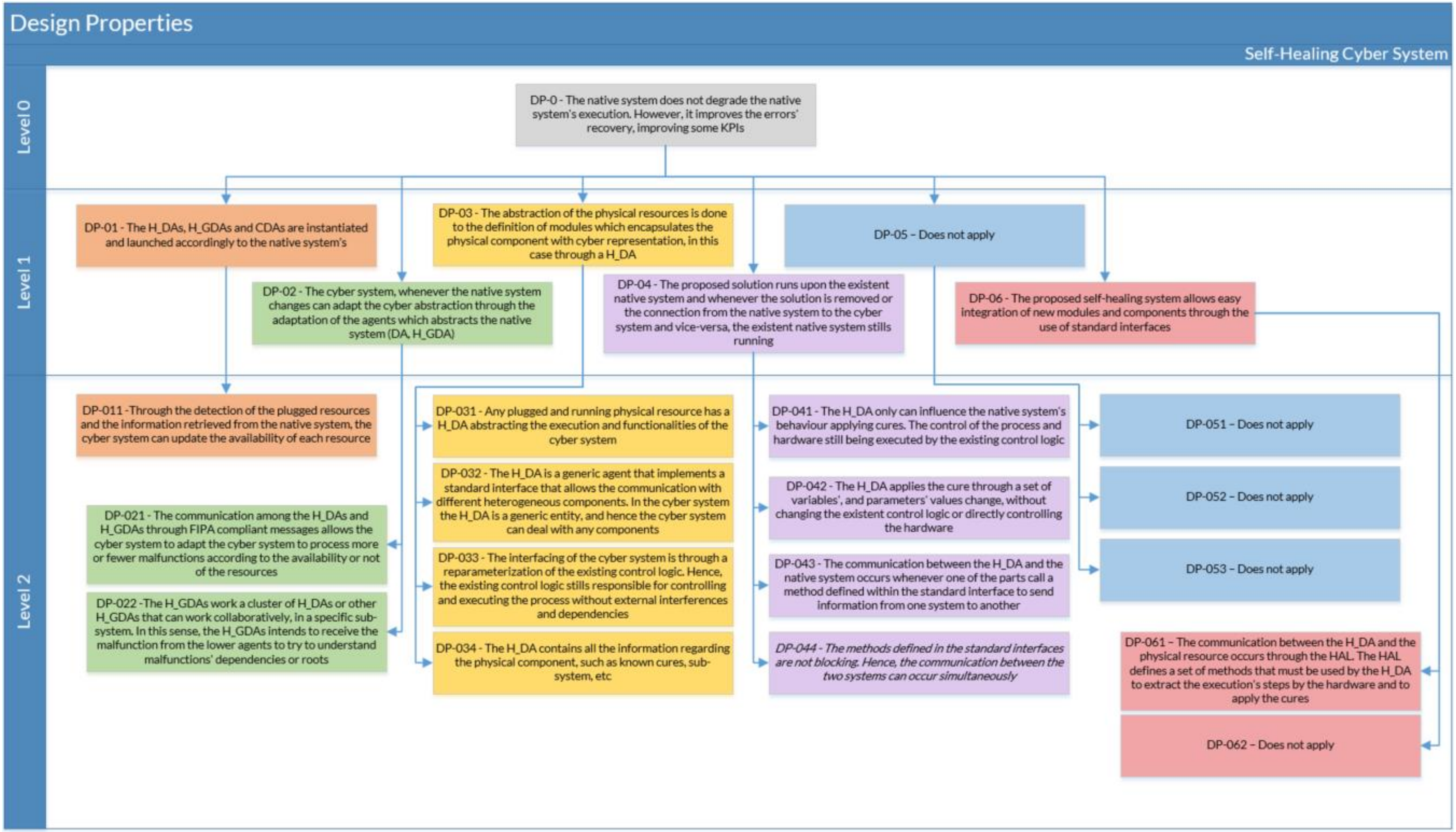
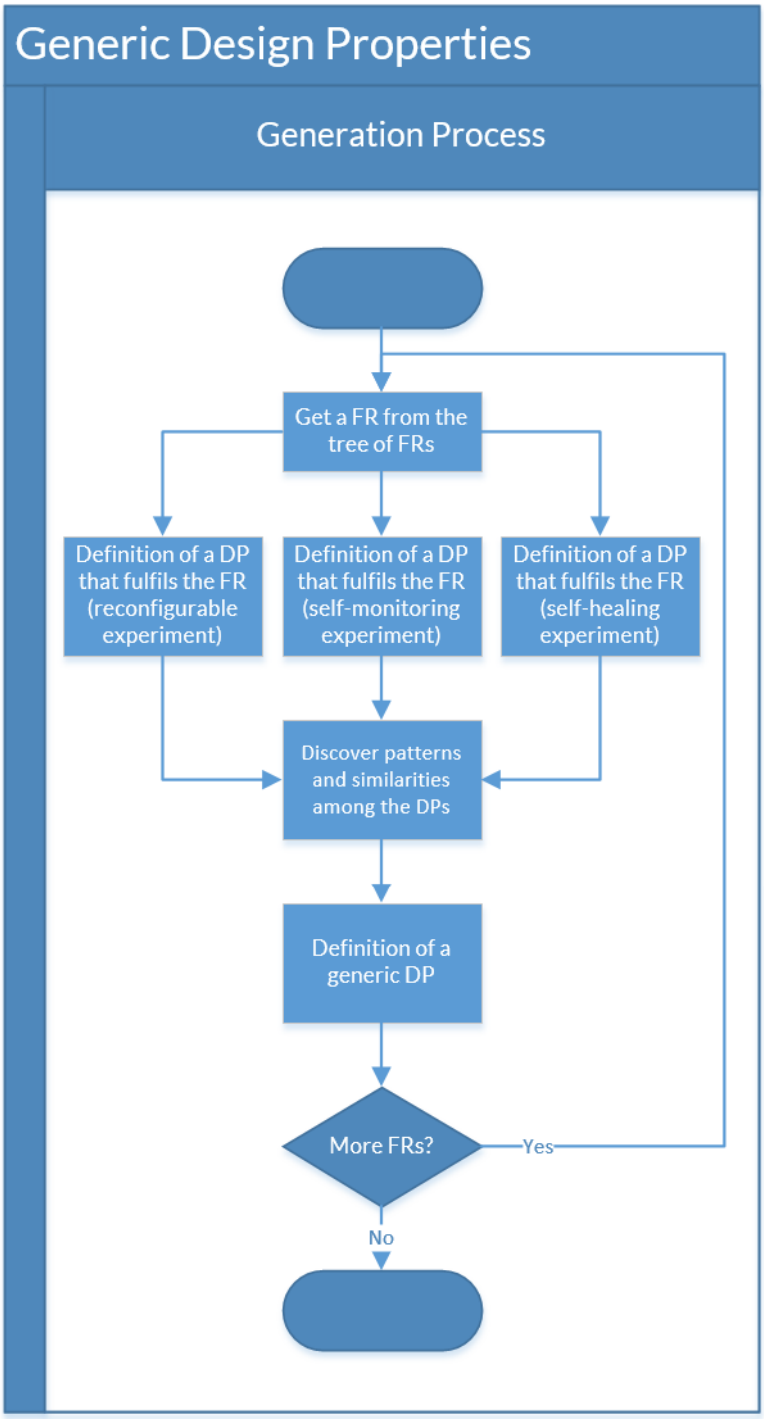


Figure 53 - Summary of the addressing of the FRs regarding the self-healing cyber system

4.1. Design Properties for Minimally Invasive Agent-based CPPS

After the design, development and assessment of the previously presented experiments, a tree of generic DPs, presented in Figure 56 were defined. The process used to generate the generic DPs is presented in Figure 54.



a

Figure 54 - Process used to generate the generic Design Properties

The process started with the elicitation of the FRs. After that, for each FR presented, it is proposed a DP which aims to fulfil the FR, presented in the previous sections.

After the definition of the DPs used to fulfil all the FR for each experiment, it is possible to analyse the DPs used for each experiment in each FR. Hence, exploring each DP applied in each use case it is possible to understand that exists similarities in the definition and application of the DPs to solve a specific FR. From that similarities, it is possible to extrapolate generic and common properties in the design of this kind of systems, generating a list of generic and common DPs for design minimally invasive agent based CPPS.

This list of generic DPs is one of the most important contributions of this research, allowing the definition of common guidelines which can be used by all the designers and developers during the proposition of less invasive solutions.

In figure , it is possible to see an example of how a generic DP is generated. The DP arises from the comparison and study of the application of specific DPs to fulfil the FR during the development of the particular experiments.

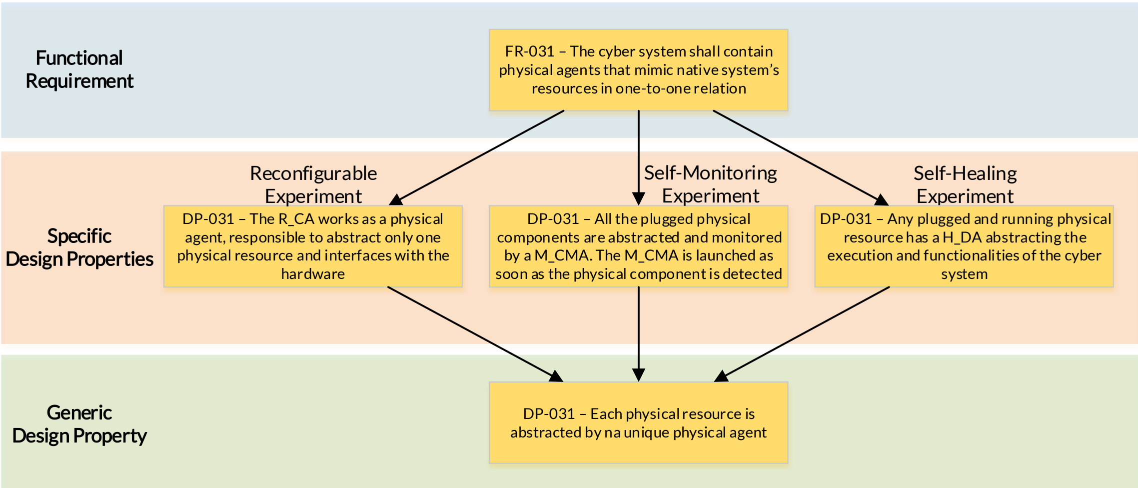


Figure 55 - Definition of the generic Design Properties from the specific Design Properties

As presented, to fulfil the FR-031, the three solutions proposed different approaches to deal with the same FR. Finding commonalities in the three specific DPs is possible to define a generic DP-031, designed to guide the development of a solution capable of delivering the FR-031.

The process proceeds until the formulation of all the generic DPs, one for each FR. Hence, this process concludes with a list of generic DPs that allows the implementation of a system capable of fulfiling all the FRs defined. The proposed tree matches the structure defined for the FRs, since the DPs intend to fulfil and respect the FRs previously elicitated.

The tree of generic DPs constitutes one of the main contributions of the proposed work, as presented in the first chapter. The following list of DPs aims to guide the researchers and developers during the design and development of minimally invasive solutions based on multi-agent technology.

Moreover, the following DPs can be useful when a solution needs to be evaluated and characterised as minimally invasive or not. Hence, in the next chapter, an evaluation model is presented. This model aims to use the following DPs and evaluate the agent-based solutions according to its applicability and invasiveness.

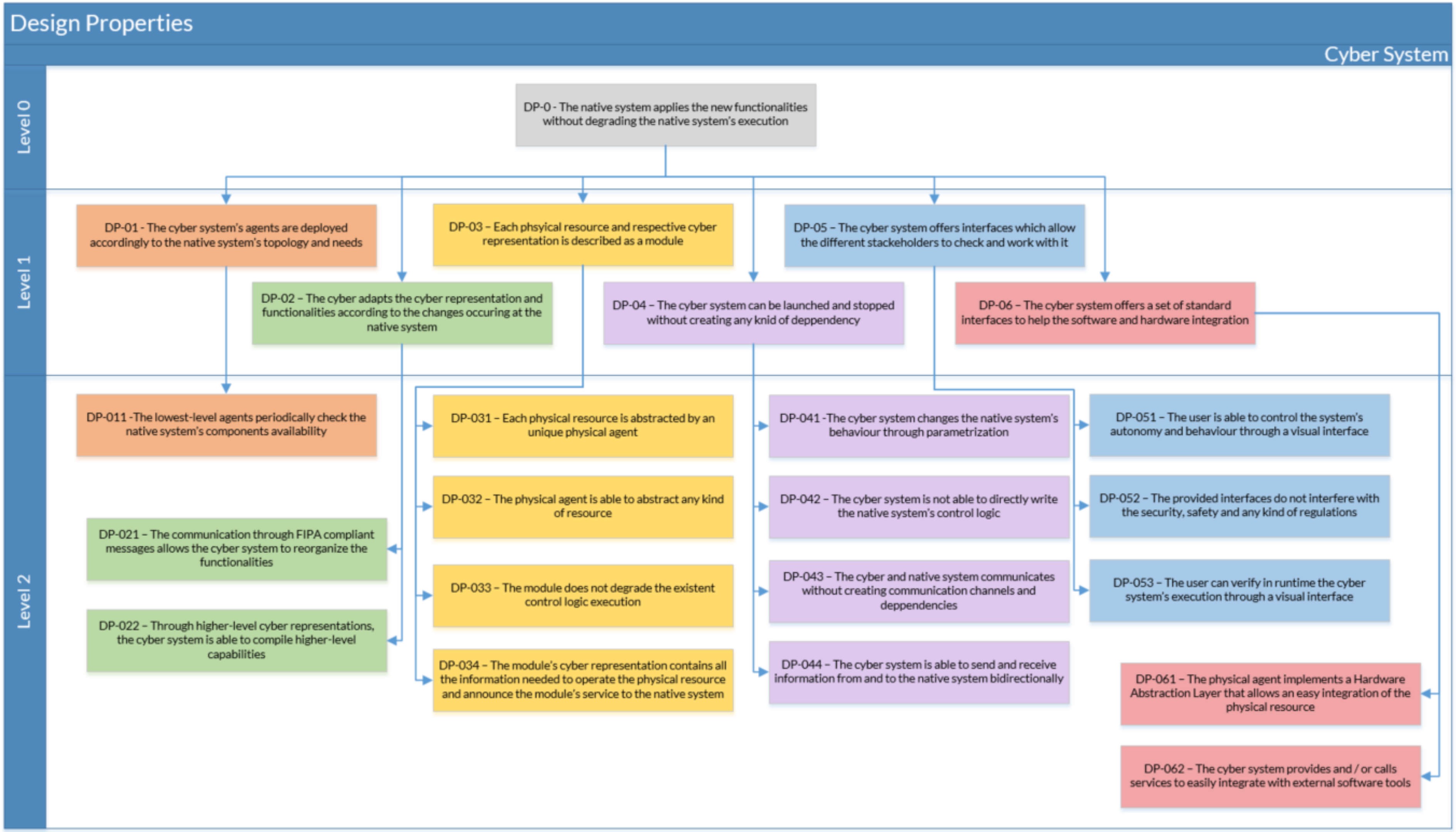


Figure 56 – Generic tree of DPs for minimally invasive agent-based CPPS

5. Evaluation Model

After the definition of the DPs to guide the development of the minimally invasive agent-based CPPS it is necessary to define a strategy and model to evaluate if the designs and developments were implemented efficiently using the previously defined DPs. Hence this section aims to present an evaluation model, it was designed to evaluate the application of the DPs separately and the solution as a whole.

5.1. Evaluation Factors

The previously defined DPs aim to guide the design and development of the systems. However, it is interesting to verify and evaluate the use of such DPs in order to rank the usage of each DP as well as evaluate the overall solution as minimally invasive.

Hence, the first evaluation step is the characterisation of each DP based on the following factors:

- DPApplication (DPApp): This factor indicates if the developed solution correctly addresses and follows a proposed DP. To evaluate a solution, each DP must be evaluated accordingly to the effective application of the DP. Hence, the first step to evaluate a solution it is ranking all the DPs accordingly to the proposed solution to address each one of them;

- Invasiveness Degree (InvD): Similarly to the previous factor, this factor applies for each DP. Since the objective is to develop minimally invasive solutions it is required not only the evaluation of the application of the DP but also if the proposed approach to follows the DP is considered invasive or not. This factor aims to evaluate if the proposed solution will cause unacceptable, or not, changes at the native system, evaluating hence if the solution offered to follow the DP is too invasive.

5.2. Quantitative Model

The Quantitative Model section presents the needed steps to create a quantitative method for evaluating the DPApp and the InvD factors. Since the evaluation of each parameter is subjective, a FIS is defined. To define this model the MATLAB Fuzzy Toolbox is used. The FIS is composed of three main features, the input, the output and the Inference Mechanism (IM) as presented in Figure 57.

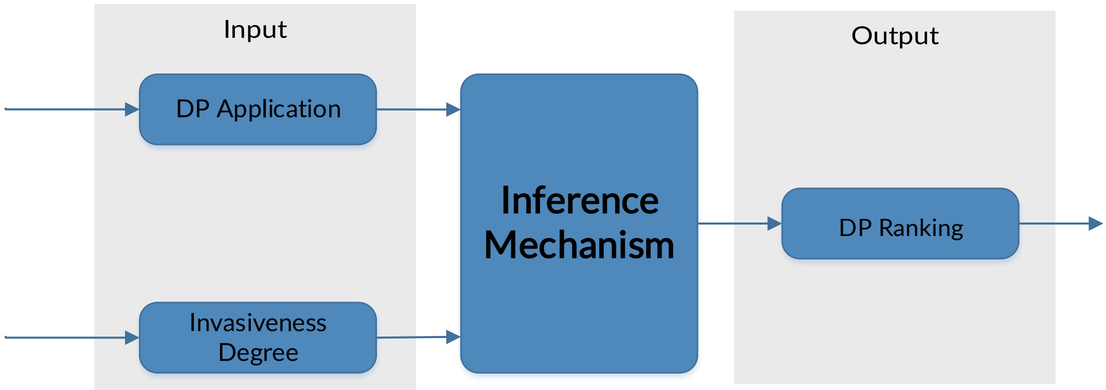


Figure 57 - FIS framework for the Ranking determination, per DP

The input of this framework is composed of DPApp and the InvD factors and the output is defined as DP Ranking, which gives the value of each DP evaluation. To define this model all the parameters need to be characterised. To do so, each factor evaluation is parameterised through the use of natural language. The categorisations for both factors are defined as:

- Very Low – VL;
- Low – L;
- Medium – M;
- High – H;
- Very High – VH.

This terminology is the commonly used in this type of evaluation (Markowski and Mannan, 2008).The categories were determined as five according to the need to create enough levels capable of defining all the known project since three were not enough and seven would increase the complexity level of the solution without offering new options. Although the input factors terminology is the same, the meaning for each one of them is different. So, to clarify this point each factor's category will be presented in different tables. The categories for the DPApp factors are depicted in Table 5.

Table 5 - Categories for the parameter DPApp

ID	Description
VL	The solution does not apply the DP
L	The solution does not apply the DP correctly
M	The solution applies the DP satisfactorily
H	The solution applies the DP very satisfactorily
VH	The solution applies the DP totally

For the InvD factor, Table 6 describes the meaning of each category.

Table 6 - Categories for the parameter InvD

ID	Description
VL	The DP application is not invasive
L	The DP application is slightly invasive
M	The DP application is invasive
H	The DP application is significantly invasive
VH	The DP application is extremely invasive

Regarding the FIS output, the used nomenclature is slightly different, only to better explain the meaning of the output of this model.

- Very Bad – VB;
- Bad – B;
- Acceptable – A;
- Good – G;
- Very Good – VG.

As explained in the for the input factors, the categories were determined as five according to the need to create enough levels capable of defining all the known project, since three were not enough and seven would increase the complexity level of the solution without offering new options. Although the input factors terminology is the same, the meaning for each one of them is different. The categories for the DP Ranking factors are depicted in Table 7.

Table 7 - Categories for the parameter DPR

ID	Description
VB	The DP application is very bad
B	The DP application is bad
A	The DP application is acceptable
G	The DP application is good
VG	The DP application is very good

With the categorisation of the inputs and output concluded, the creation of the fuzzy sets is the following steps. In accordance with Guyonnet et al., the definition of any distribution should be made based on the collection of statistical data and fit the theoretical distribution into a relative frequencies histogram.

However, for this specific case, there is not enough available data to use the statistical-based method approach. So, another method should be used to determine the most suitable shape to describe this type of systems.

The majority of the available literature content focus its attention in triangular shapes, due to its simplicity. However, when a human evaluation is required, the most used shape to characterise it is the Gaussian shape. This shape is commonly found in papers that used experts' opinions to evaluate a system (Markowski and Mannan, 2008). So, based on this the most suitable shape to be used in this particular case is the Gaussian.

Based on these, the membership functions must be defined for each of the five characteristics. To do so, formula required is given by equation 4.1.

$$\mu = e^{\left(\frac{-x(x-c)^2}{2\sigma^2}\right)} \tag{4.1}$$

The μ represents the category's Membership Function (MF), for every single category. After each MF is defined, the Figure 58 is obtained. The graphic of Figure 58 is a representative image for the two inputs, the DPApp and InvD, and the output, the DPR.

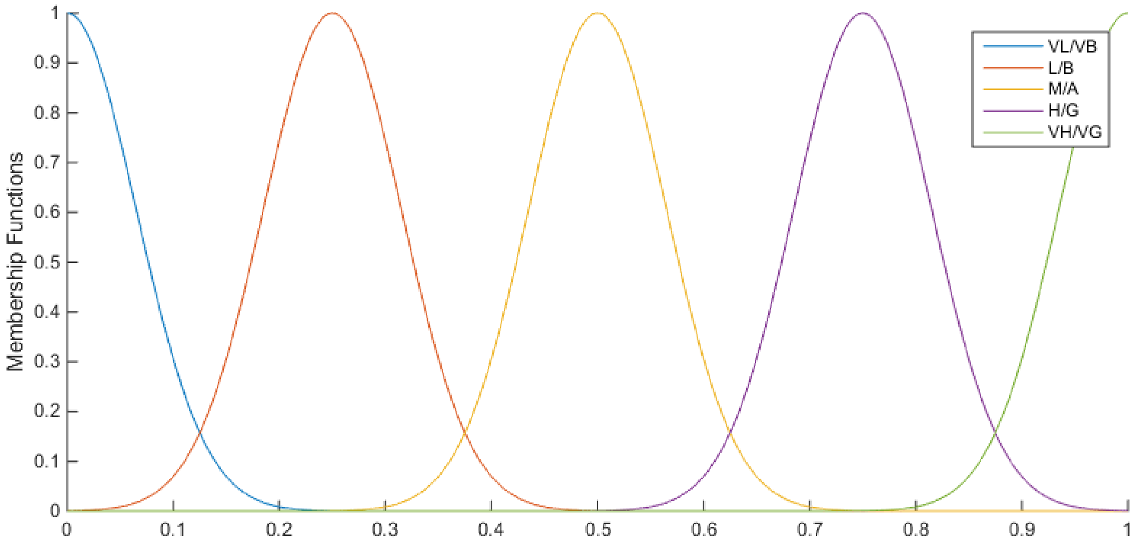


Figure 58 - Categories' MFs fuzzy set for input and output parameters

From left to right are represented the MF from VL/VB, L/B, M/A, H/G and VH/VG valid for the model's input/output. The following step of the model's definition is the IM definition. For this model, the IM used is the Mamdani's method, which is commonly used in the design of these models. Being the IM chosen the rules need to be set.

Each rule is defined as a combination of the two factors, DPApp and InvD by the parameters previously defined. The rules structure is defined as:

$$(\boldsymbol{DPApp} = Very\ Low) \wedge (\boldsymbol{InvD} = Very\ High) \rightarrow \boldsymbol{DPR} = Very\ Bad$$

Based on this example, the rules are defined, as shown in Table 8.

Table 8 - List of rules for the DP Evaluation Model

ID	Condition			DPR
	DPApp	Op.	InvD	
1	Very Low	and	Very High	Very Bad
2	Very High	and	Very Low	Very Good
3	High	and	Very Low	Good
4	Very High	and	Very High	Bad
5	Medium	and	Very High	Bad

Being everything defined, the model needs to be validated. The model's validation process is an iterative procedure, which means that if some of the three validation tests are not satisfactory, the model should be revised and some changes need to occur.

To guarantee the model's conformity, the first validation to be performed is named the extreme condition test. The purpose of this test is to force the model to extreme conditions and validate the results against the ideally expected. For this, the ideal results should be equal to the maximum/minimum/medium values of the output, but due to the iterations performed by the used software when the mathematical model is applied, which causes successive approximations, the tests' values should be close to the optimal ones. It was considered that if the values are not far from the optimal result more than 0.05 in a scale from 0 to 1.

The results of this validation are shown in Table 9.

Table 9 - Extreme conditions test

	DPApp	InvD	DPR
Minimum	0.0	1.0	0.0494
Medium	0.5	0.5	0.5000
Maximum	1.0	0.0	0.9510

As it is possible to analyse, the results for this test respect the restriction (Table 9).

The second performed test is the face validity. This test requires the graphical representation of the rules' application to be performed. This graphic is defined by a surface which indicates the system's behaviour. If the surface presents any irregularity as not being always increasing or decreasing trend the test indicates that the model is not valid.

The generated surface is presented in Figure 59.

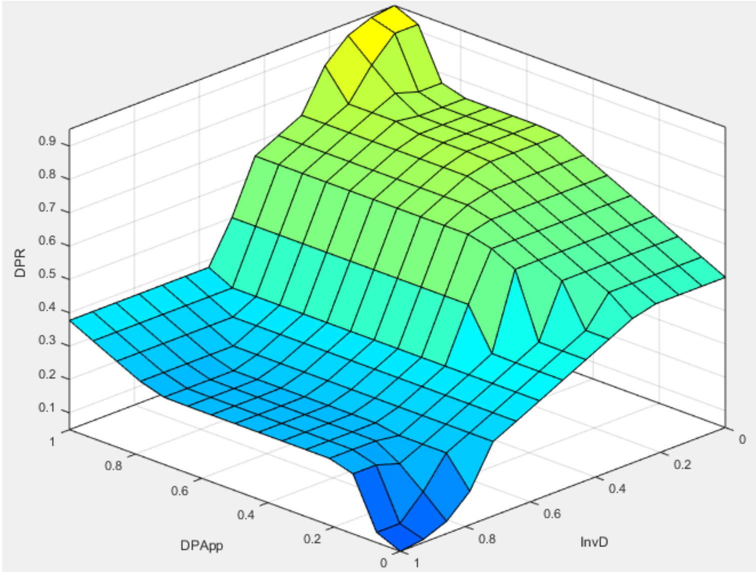


Figure 59 - Face validity

As it is possible to verify in Figure 59 the surface does not present any irregularity.

Finally, the third and last test is the behavioural analysis. To perform this test and verify the reasonability of the behaviour one input factor is kept constant, and the other one is smoothly increased. With this process it is possible to understand the behaviour of the system and discover possible unexpected trends that should be solved.

The behavioural analysis for the DPApp and InvD are shown in Figure 60 and Figure 61, respectively.

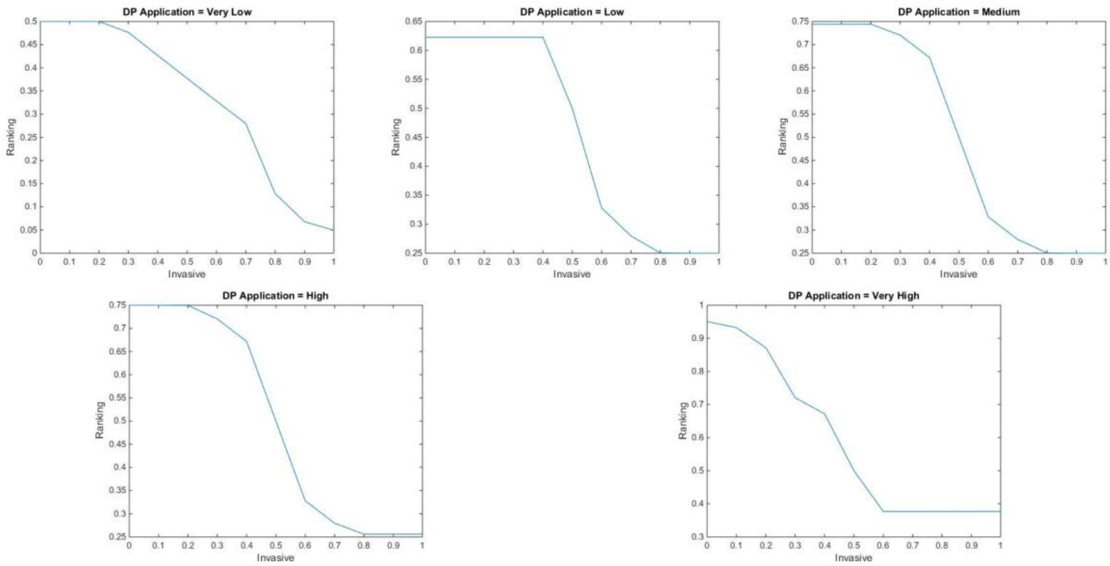


Figure 60 – Behavioural analysis with fixed DPApp

As it is possible to verify the behavioural analysis for the DPApp factor does not present any anomaly since is presents a decreasing trend for all the parameters.

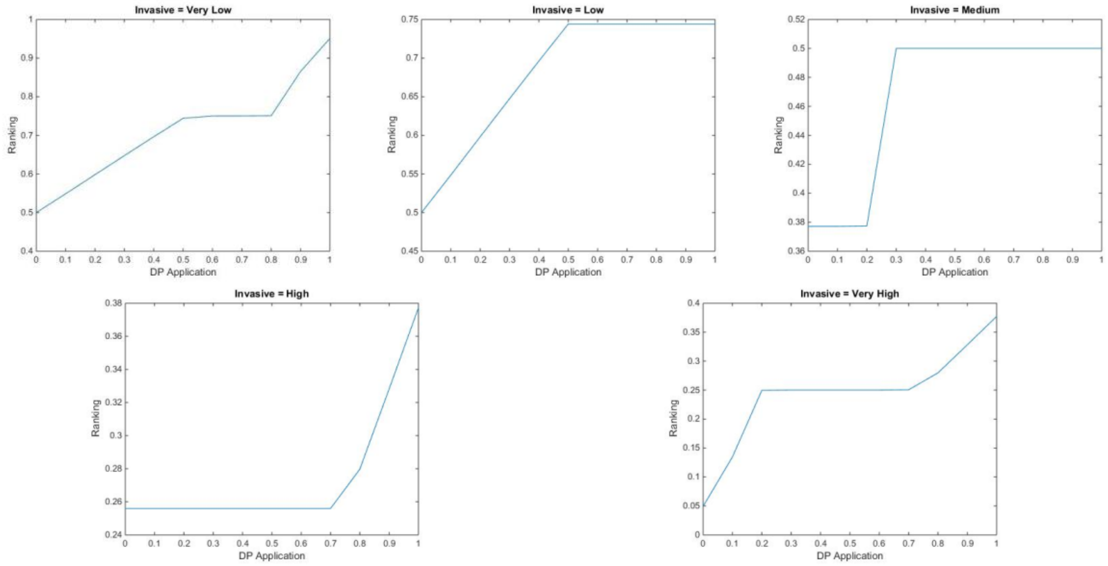


Figure 61 - Behavioural analysis with fixed InvD

The same it is possible to verify for the InvD factor. However, this behavioural factor analysis presents an increasing trend for all the parameters.

5.3. Overall Evaluation Score

The previously presented evaluation steps are able to evaluate the application of each DP for a specific solution. However, it still not able to evaluate a solution as a whole. Hence, the Overall Score presented in the following mathematical expression aims to evaluate the overall solution based on the results generated from the evaluation of each DP.

$$Overall\ Score = \frac{\sum_{n=1}^{n\ DPs}(DPR_n\ \times Relevance_n)}{\sum_{n=1}^{n\ DPs}(Relevance_n)}$$

The Overall Score mathematical expression is an application of a weighted arithmetic mean of the previously generated DPRs. The weights, here presented as Relevance allows the evaluators to rank the application of the different DPs accordingly to relevance for the proposed solution. For instance, a solution focused on monitoring can and should be evaluated differently from a solution focused on reconfiguration because the solution itself has different FRs.

The Relevance values must be defined in the interval between 0 and 1.

$$0 \leq Relevance \leq 1$$

5.4. Summary of the Evaluation Model

An overview of the entire model is presented in Figure 62. The model starts with the definition of the DPs to be evaluated. Afterwards, each DP is evaluated by an expert based on the factors previously presented (DPApp and InvD).

For each DP the FIS will generate a DPR which ranks the DPApp accordingly to the previously defined factors. Hence, each DP is already ranked and can be analysed separately.

The last step of the model aims to merge the generated DPRs and rank the entire solution using the Overal Score expression that calculates the weighted arithmetic mean with the DPRs. The Relevance of each DP defines the weight for the solution being evaluated.

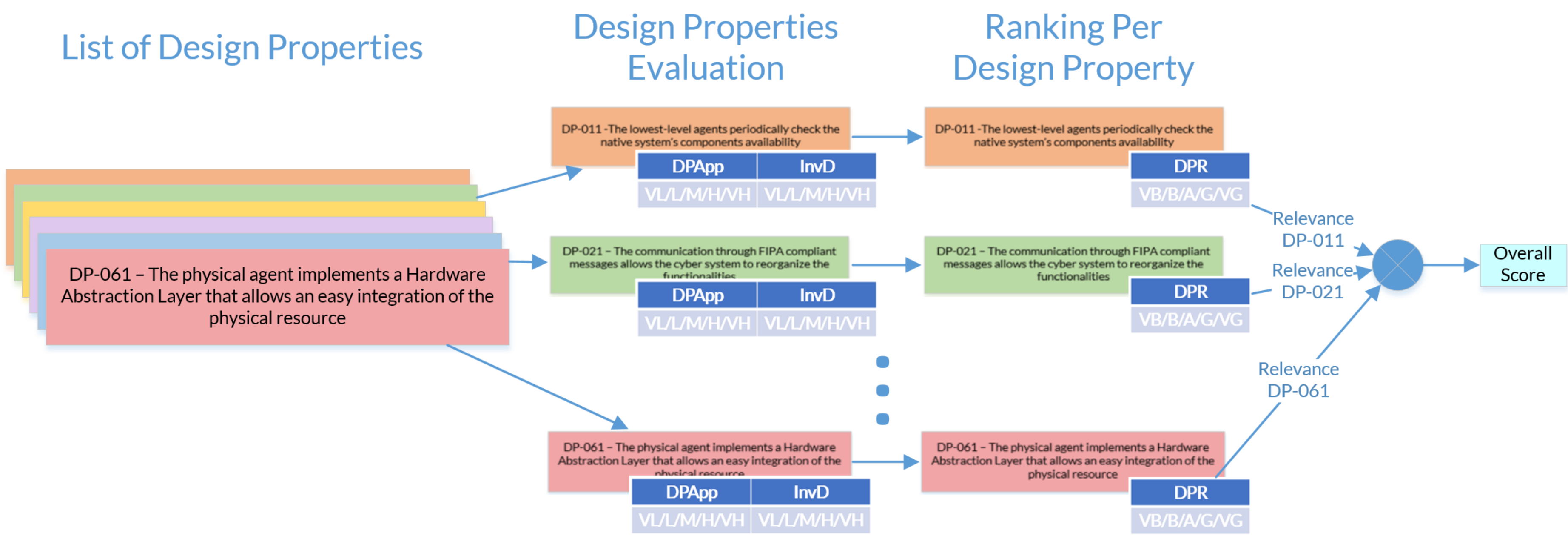


Figure 62 - Overall summary of the evaluation model

6. Validation / Discussion

This chapter presents the evaluation of five agent-based CPPS. Firstly, the three experiments presented in chapter four are evaluated. The evaluation of the experiments aims to rank and assess the experiments which were designed to be minimally invasive. After assessing the results of the known cases, two other test cases are analysed and discussed using the evaluation model.

It is essential to define which DPs must be used in the evaluation. The DPs from different levels are not independent, for instance DP-01 includes the application of DP-011 and DP-012. Hence, only the DPs created at level 2 must be used for the evaluation. The following list contains all the DPs which must be used to evaluate the solutions:

- DP-011 -*The lowest-level agents periodically check the native system's components availability*;
- DP-021 – *The communication through FIPA compliant messages allows the cyber system to reorganise the functionalities*;
- DP-022 – *Through higher-level cyber representations, the cyber system can compile higher-level capabilities*;
- DP-031 – *Each physical resource is abstracted by a unique physical agent*;
- DP-032 – *The physical agent can abstract any resource*;
- DP-033 – *The module does not degrade the existent control logic execution*;
- DP-034 – *The module's cyber representation contains all the information needed to operate the physical resource and announce the module's service to the native system*;
- DP-041 -*The cyber system changes the native system's behaviour through parametrisation*;

- DP-042 – *The cyber system is not able to directly write the native system's control logic*;
- DP-043 – *The cyber and native system communicates without creating communication channels and dependencies*;
- DP-044 – *The cyber system can send and receive information from and to the native system bidirectionally*;
- DP-051 – *The user can control the system's autonomy and behaviour through a visual interface*;
- DP-052 – *The provided interfaces do not interfere with the security, safety and any regulations*;
- DP-053 – *The user can verify at runtime the cyber system's execution through a visual interface*;
- DP-061 – *The physical agent implements a Hardware Abstraction Layer that allows easy integration of the physical resource*;
- DP-062 – *The cyber system provides and calls services to integrate with external software tools easily*.

The template which must be filled for the evaluation of each solution is presented in Table 10.

Table 10 - Solution evaluation's template

ID	DPApp	InvD	Relevance	DPR
DP-011				
DP-021				
DP-022				
DP-031				
DP-032				
DP-033				
DP-034				
DP-041				
DP-042				
DP-043				
DP-044				
DP-051				
DP-052				
DP-053				
DP-061				
DP-062				
	Overall Score			
	With Maximum Relevance (=1)			
	With the Assigned Relevance			

The green spaces must be filled using the methodology previously described, in order to describe the DPAapp, InvD and the Relevance for each use DP. Firstly, DPApp and InvD variables must be filled with the possible inputs (VL, L, M, H or VH). Posteriorly, the relevance column must be filled with a value between 0 and 1 for each DP. The FIS automatically generates the DPR values. The Overall Score aims to rank the overall solution based on the DPs evaluation. The first row presents the Overall Score without using the relevance. All values of the Relevance are 1, the maximum value. On the other row, the Overall Score presents the result taking in account the values assigned in the Relevance column.

In order to verify which are the most critical DPs for each solution, the DP positioning diagram was created. Each DP can be inserted in one of the three possible areas (Ok, Attention and Critical). Figure 63 presents the DP positioning diagram, with the different areas according to the DPApp and InvD evaluation.

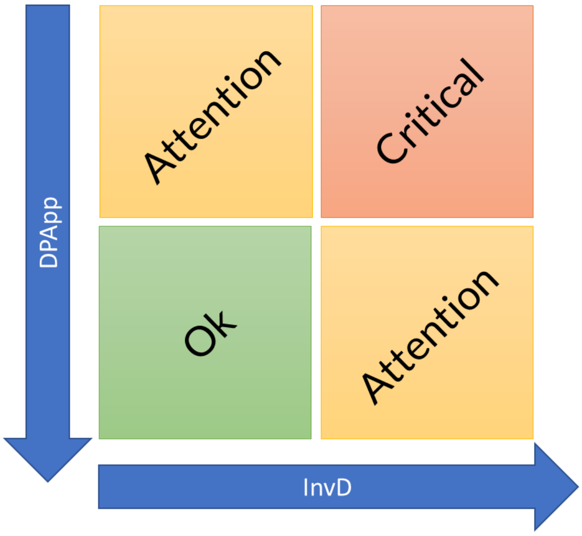


Figure 63 - DP positioning

The DPs are positioned according to the following rules, presented in Table 11.

Table 11 - DP positioning rules

DPApp	InvD	Position
Very High or High	Very Low or Low	Ok
Medium or Low or Very Low	Very Low or Low	Attention
Very High or High	Medium or High or Very High	Attention
Medium or Low or Very Low	Medium or High or Very High	Critical

With this diagram, it is possible to verify which DPs need to be handled in order to improve the application of the solution and reduce the degree of invasiveness.

6.1. Evaluation of the Experiments

6.1.1. Reconfigurable Agent-based CPPS

This experiment was designed under the FP7 PRIME project. The proposed experiment aims to deliver a reconfigurable CPPS capable of to reconfigure existent native system without requiring significant changes and adaptations.

Table 12 presents the evaluation of the first experiment, which it focuses on reconfiguration.

Table 12 - Reconfigurable experiment evaluation

ID	DPApp	InvD	Relevance	DPR
DP-011	High	Low	1	0,744074888
DP-021	Very High	Very Low	1	0,950626357
DP-022	Very High	Very Low	1	0,950626357
DP-031	Very High	Very Low	1	0,950626357
DP-032	Very High	Low	1	0,744074888
DP-033	Very High	Low	1	0,744074888
DP-034	High	Medium	1	0,5
DP-041	Very High	Low	1	0,744074888
DP-042	Very High	Low	1	0,744074888
DP-043	Very High	Very Low	1	0,950626357
DP-044	Very High	Medium	1	0,5
DP-051	High	Very Low	1	0,749988368
DP-052	High	Very Low	1	0,749988368
DP-053	Medium	Very Low	1	0,744074888
DP-061	Very High	Low	1	0,744074888
DP-062	High	Very Low	1	0,749988368
	Overall Score			
	With Maximum Relevance (=1)			0,76631217
	With the Assigned Relevance			0,76631217

As expected and it is possible to verify from an overall view of the table, it is possible to see that in most of the cases, the DPApp is High or Very High, and the InvD is Low or Very Low. Hence, the Overall Score of this solution, and as expected, is very high (0,766).

The solution proposed by this experiment is one of the experiments used to induce the DPs. Since this solution was developed to be a minimally invasive solution, the result demonstrates the intention of this design.

In Figure 64 shows that the implementation of three DPs must be improved. The improvement of these DPs will highly improve the overall solution. However, these three DPs are not considered critical.

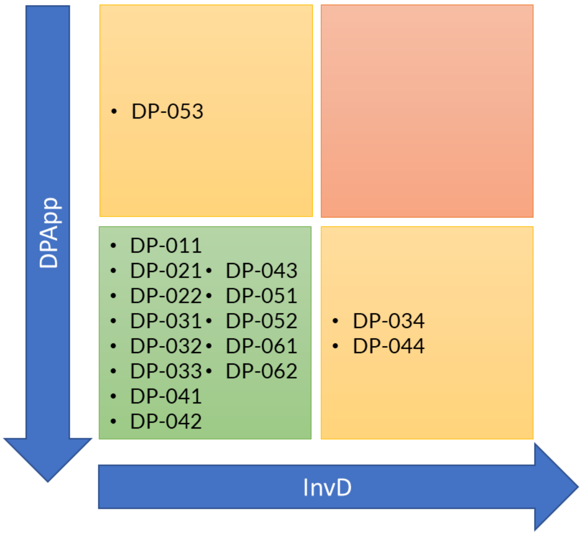


Figure 64 - DP positioning for the reconfigurable experiment

As it is possible to verify in the diagram, the DP-053 must be revised. This evaluation can be considered acceptable, but it is very close to an unacceptable application. In fact, the solution offers a visual interface which allows the operator to consult the information regarding the cyber system execution, as presented in Figure 65.

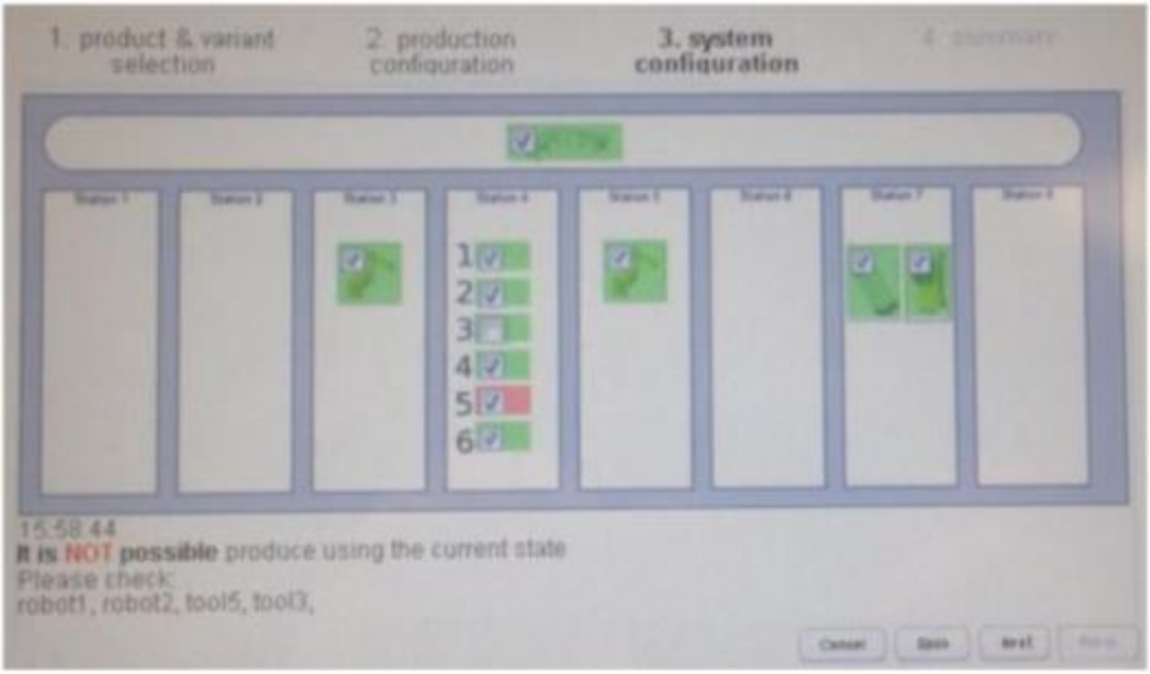


Figure 65 – Visual interface developed for the reconfigurable experiment

The visual interface shows the physical resources available and not available (on green and red respectively). Based on that information, the operator can choose which resources must be used by the native system, selecting the checkbox on the left of each resource. Moreover, on the bottom, it is shown if it is possible or not to produce the desired product, based on the resources available and selected. If not, the system presents a list of possible physical resources which could be added or selected in order to allow the production of the desired product.

Although the main functionalities of the cyber system are controlled and checked by the operator, this visual tool can be improved with the addition of information regarding the internal state of the cyber system, for instance, the available CSks.

Regarding DP-034 and DP-044 the DPApp is High and Very High respectively. However, regarding the InvD, it is considered Medium in both cases. Hence, it is recommendable to revise the implementation of these two DPs in order to reduce the degree of invasiveness of both.

In order to integrate a new physical resource, it is necessary to code a blueprint file based on XML (DP-034). That blueprint file must describe the skills offered by the resource, the physical location and the library that must be instantiated and used by the physical agent to communicate and reconfigure the hardware. This blueprint file is posteriorly copied to the hardware where the physical agent will run, and it is consulted by the physical agent immediately after its starting the execution.

Although it is a functional solution, it is necessary to know the structure of the blueprint file and how to work with XML files. It must be interesting to develop a tool capable of generating the blueprint files dynamically, based on the information inserted by the operators or systems integrators.

The DP-044 focuses the bidirectional communication between the physical agent and the physical resource. It is necessary, for each component, that the system's integrator code the mechanisms to receive and apply the parametrisations as well as the mechanisms to retrieve the states of the native system. This step implies a considerable integration effort that must be performed whenever a new component is integrated.

This challenge needs to partially handled by the cyber system and partially handled by the native system. The effort necessary to integrate new resources could be drastically reduced if some well-known protocol was used to deliver the information from one point to another. For instance, OPC-UA.

6.1.2. Self-Monitoring Agent-based CPPS

This experiment was designed under the FP7 PRIME project. The objective of this agent-based cyber is to deliver a framework, capable of retrieving raw data from the existent native system and pre-process this raw data in order to generate new and relevant knowledge about the execution of the native system. Afterwards, the cyber system can also deliver this data to external software tools.

The evaluation of the self-monitoring experiment is described in Table 13.

Table 13 – Self-monitoring experiment evaluation

ID	DPApp	InvD	Relevance	DPR
DP-011	Very High	Very Low	1	0,950626357
DP-021	High	Very Low	1	0,749988368
DP-022	Very High	Very Low	1	0,950626357
DP-031	Very High	Low	1	0,744074888
DP-032	Very High	Low	1	0,744074888
DP-033	Very High	Very Low	1	0,950626357
DP-034	Very High	Very Low	1	0,950626357
DP-041	Very Low	Very Low	0	0,5
DP-042	Very High	Very Low	1	0,950626357
DP-043	Very High	Very Low	1	0,950626357
DP-044	Very High	Low	1	0,744074888
DP-051	Very Low	Very Low	0	0,5
DP-052	Very High	Very Low	1	0,950626357
DP-053	Very High	Very Low	1	0,950626357
DP-061	Very High	Very Low	1	0,950626357
DP-062	Very High	Very Low	1	0,950626357
	Overall Score			
	With Maximum Relevance (=1)		0,84339938	
	With the Assigned Relevance		0,89245643	

Two DPs, which they can be easily identified, have low DPRs. The evaluation of the other DPs is very high. Consequently, the Overall Score of this solution is very high also. This solution results in the best application of the DPs. Hence, it is the best-ranked solution among the evaluated cases. An overview of the evaluation of the experiments and test cases is presented at the end of this chapter. The two DPs identified as not implemented as desired are also easily identified in Figure 66.

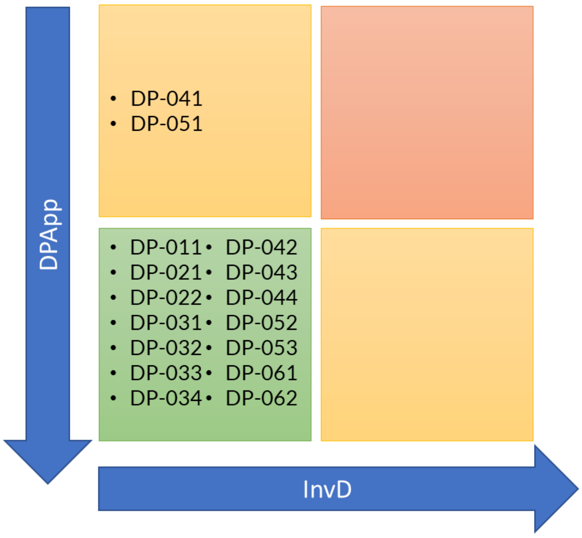


Figure 66 - DP positioning for the self-monitoring experiment

Since the objective of the solution was the extraction of data from the current system and does not change or influence the normal execution of the native system, the relevance for DP-041 and DP-051 was defined as 0. In fact, this solution proposed extracts the data from the native system and pre-process it without applying any reconfiguration, reparameterization or control. Since DP-041 is related to the ability to change the behaviour of the native system, these DP is not applied to this solution. Similarly, the DP-051 is not applicable because if the system does not change its behaviour, the operator cannot control this process. In Figure 67 a frame of the visual interface developed is shown.

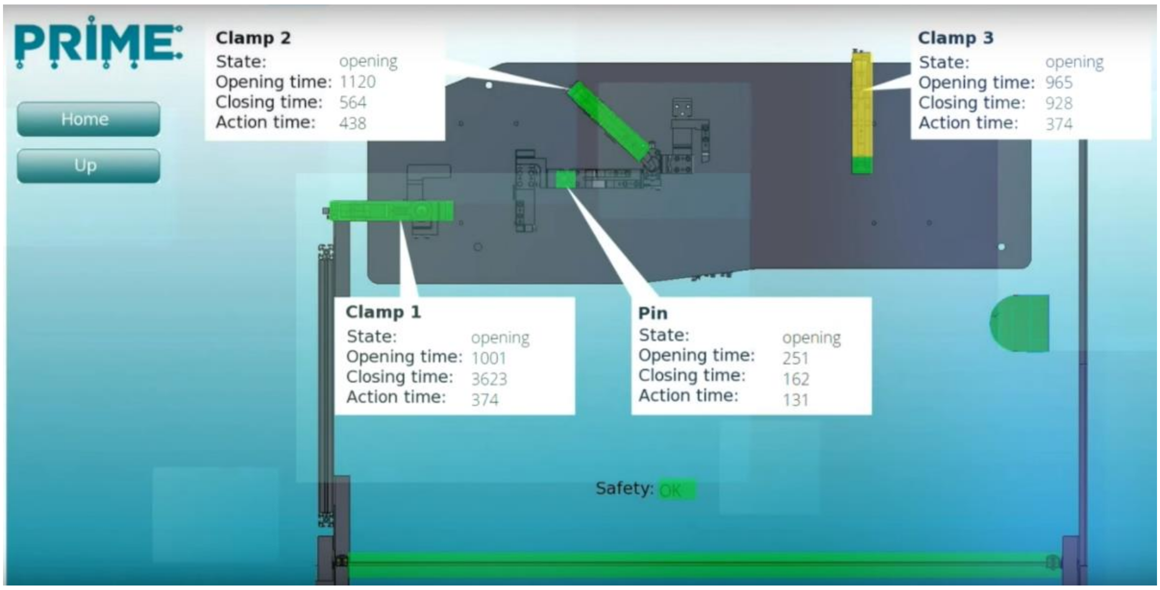


Figure 67 - Visual interface developed for the self-monitoring experiment

The relevance and the application of these two DPs are deeply related to the objectives of this solution. Since the solution was designed not to control or reconfigure the behaviour of the system, and these DPs are related to that functionalities, these DPs are not applied in this case. Although, since the proposed solution aims to retrieve and analyse data to generate new knowledge and if possible to trigger maintenance or optimisations, it could be interesting to apply these DPs and develop a solution capable of optimising itself based on the infrastructure already designed to retrieve and process the data. For instance, in this experiment, it was tested the triggering of alerts indicating that a specific component was not working properly, as presented in Figure 68.

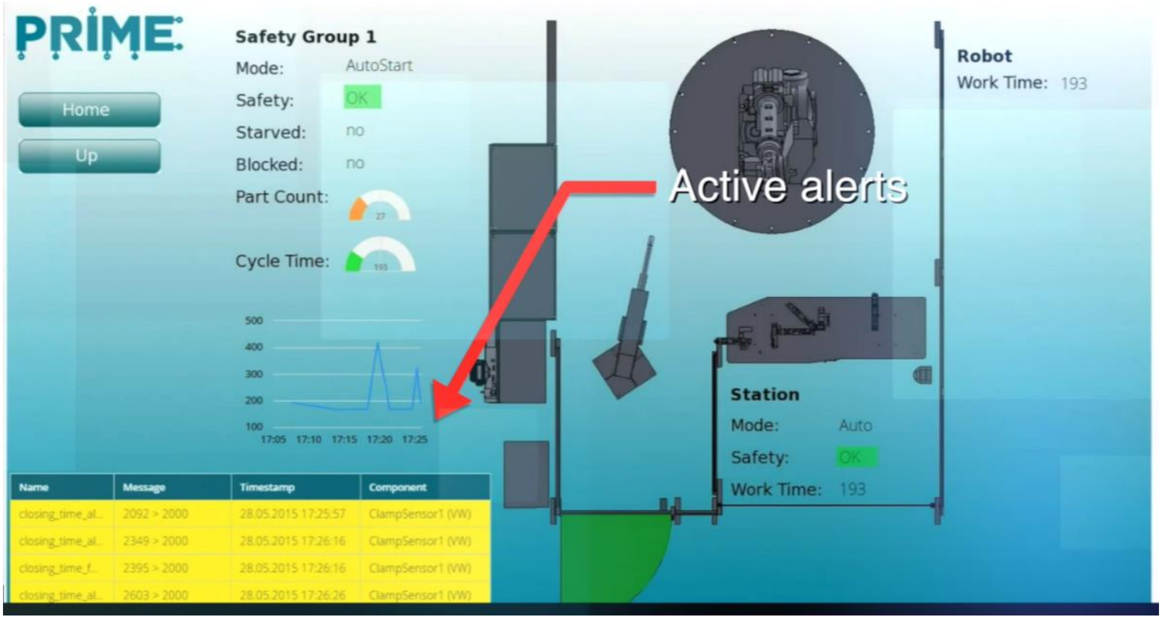


Figure 68 - Alerts triggered by the self-monitoring cyber system

In this case, it is expected that the maintenance time solves the problem and clean the alert from the visual interface. Although, it could be interesting to deliver a reconfiguration mechanism capable of reallocating the production and reduce the use of this station.

6.1.3. Self-Healing Agent-based CPPS

This experiment was designed in a research proposed during a Masters. This cyber system aims to implement an intelligent and evolvable self-healing system based on AIS algorithms.

The evaluation of the last experiment, which it focuses self-healing and error recovery is presented in Table 14.

Table 14 - Self-healing experiment evaluation

ID	DPApp	InvD	Relevance	DPR
DP-011	High	Low	1	0,744074888
DP-021	Very High	Very Low	1	0,950626357
DP-022	Very High	Very Low	1	0,749988368
DP-031	Very High	Very Low	1	0,744074888
DP-032	Very High	Low	1	0,744074888
DP-033	Very High	Low	1	0,744074888
DP-034	High	Medium	1	0,749988368
DP-041	Very High	Low	1	0,744074888
DP-042	Very High	Low	1	0,950626357
DP-043	Very High	Very Low	1	0,744074888
DP-044	Very High	Medium	1	0,5
DP-051	High	Very Low	1	0,5
DP-052	High	Very Low	1	0,5
DP-053	Medium	Very Low	1	0,5
DP-061	Very High	Low	1	0,744074888
DP-062	High	Very Low	1	0,5
	Overall Score			
	With Maximum Relevance (=1)		0,694359604	
	With the Assigned Relevance		0,694359604	

The Overall Score is not so high as the previous cases. It is possible to verify that some DPs are not followed or if followed the solution is considered invasive.

Figure 69 shows the positioning of each DP according to the evaluation of the DPApp and InvD.How it is possible to verify, the application of five DPs must be improved in order to increase the Overall Score of the solution proposed.

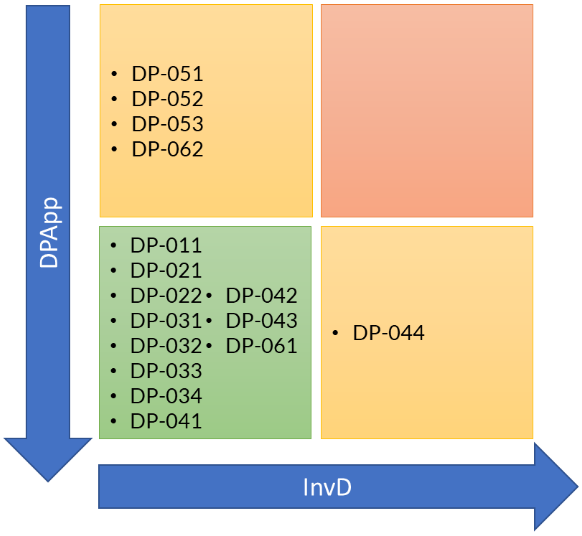


Figure 69 - DP positioning for the self-healing case

The application of the DP-051, DP-052 and DP-053 are Very Low because no visualisation tool was developed or integrated with this cyber system, and since these DPs focus on the development of tools to guide and help the humans, the evaluation of these DPs is very poor. In order to increase the usefulness of this solution and as presented by the DPs, it is essential for the operator to know and see the current state of the system and which changes the cyber system is applying in the production line. In this case, the system could be improved through the development of a visual interface where the operator could see which malfunctions were detected. Hence, the operator could check if the cure was applied correctly, and in some cases, the problems could be related to situations easily solved by the human.

Similarly to the reconfigurable experiment, this experiment could benefit if some well-known communication protocol was used to deliver the information between the two systems (cyber and native system). Although the DP-044 was fulfilled, it is required a quite considerable integration effort to do it. Hence, it could be useful to develop a strategy and solution to reduce this effort.

6.2. Test Cases

In order to validate and use the methodology with other solutions, two external developments were used to assess the usage of the evaluation model.

6.2.1. openMOS

openMOS is a European project funded by the European Commission under the EU Framework Programme for Research and Innovation Horizon 2020 (2014 - 2020) within the FoF – Technologies for Factories of the Future initiative.

The idea of the openMOS project, shown in Figure 70, is to integrate well established plug-and-produce system concepts from many years of research in this field, into industrial relevant technology platforms which have emerged in recent years. The openMOS consortium has been carefully chosen to bring together representative industrial and technology development organisations to demonstrate the technical and commercial viability of plug-and-produce in different industrial sectors.

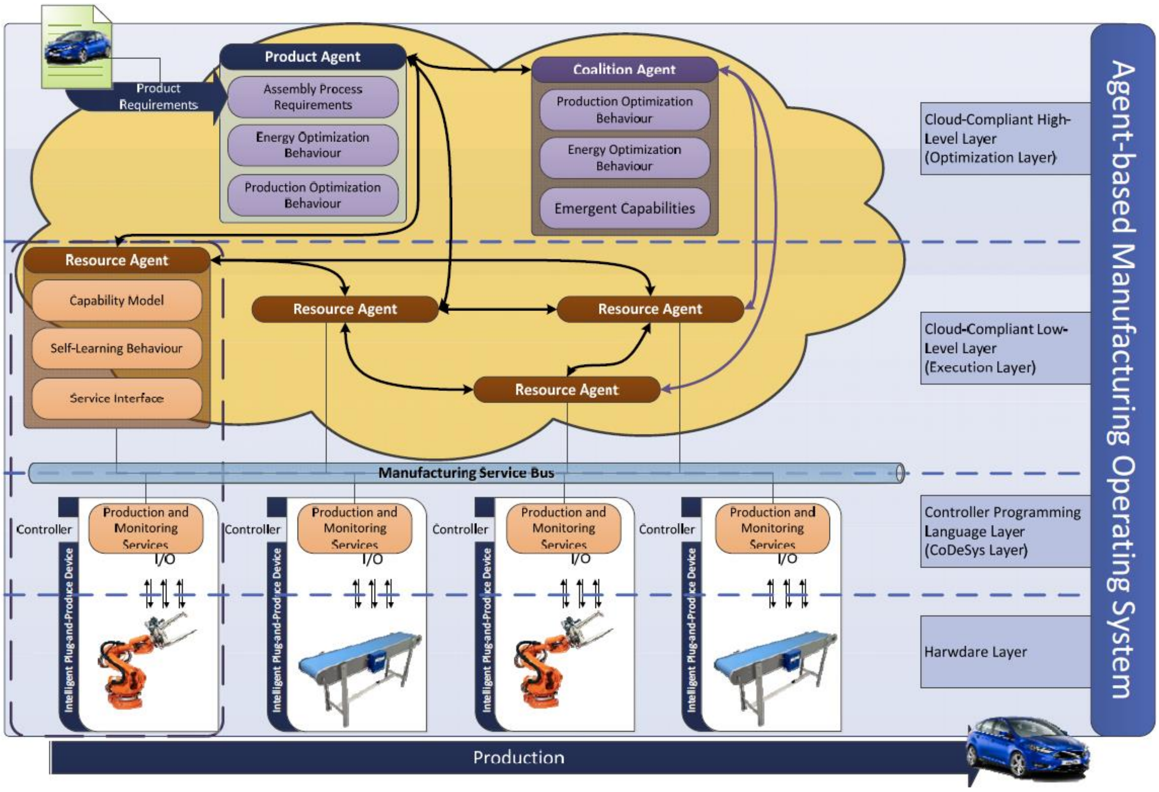


Figure 70 - openMOS concept overview, adapted from (openMOS, 2016)

The innovation targeted by the openMOS project is a common, openly accessible plug-and-produce system platform which allows all stakeholders in the automation system value chain to come together and jointly develop and exploit solutions.

The proposed multi-agent system must be able to deal with different pluggable production modules and deliver optimised solutions to deal with the plugged resources and the products to be produced. To deal with this complex environment the following multi-agent environment has been proposed (openMOS, 2017):

- Product Agent: This agent abstracts one product being produced. It is responsible for guaranteeing that the product executes all the required production steps according to the needed parameters and requirements.
- Resource Agent: All the resources which could add value to the product, performing production tasks, are abstracted by one Resource Agent. Since this agent abstracts and interface with one physical resource, it is considered a physical agent. However, in the openMOS case exists one more physical agent type;
- Transport Agent: The Transport Agent is also considered a physical agent. All the physical components capable of handle the products along the production system are abstracted as Transport Agents;
- Cloud Interface Agent: The Cloud Interface Agent was designed to allow the communication, through a service bus approach, between the service bus and the multi-agent environment. The only exception is when the service bus sends execution information to the respective Resource Agents, in order to avoid bottlenecks and delays;
- Deployment Agent: The Deployment Agent aims to allow dynamic and easy deployment of all the agents. This agent receives requests from the Cloud Interface Agent to deploy or remove agents according to the native system's topology;
- Production Optimizer Agent: The Production Optimizer Agent is a higher-level agent capable of managing all the cyber level information and optimise the production according to the available resources. To do that, it uses an algorithm defined for the optimisation.

Figure 71 describes the UML class diagram for the openMOS's CPPS.

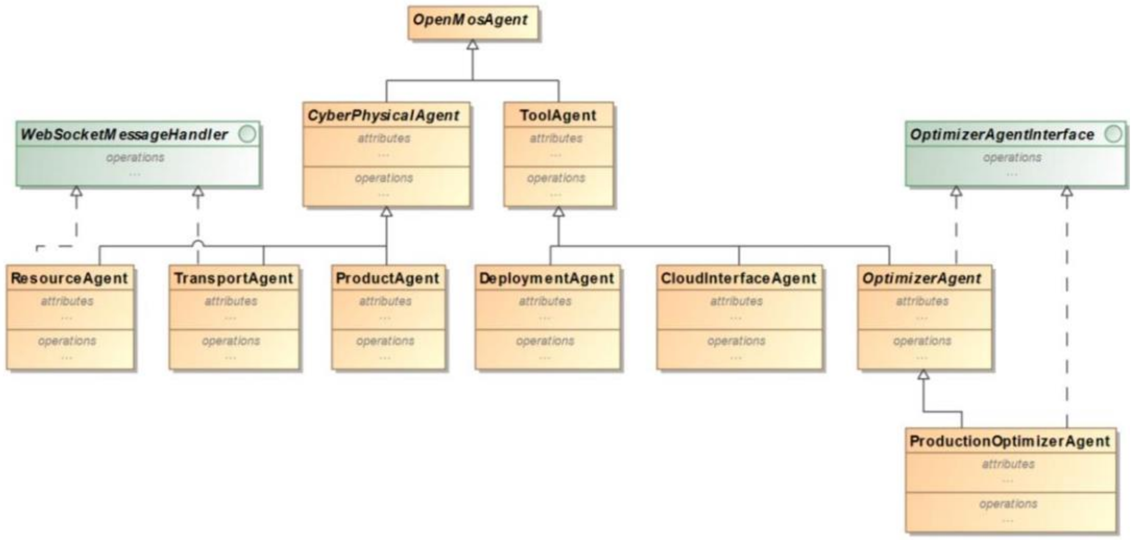


Figure 71 - Class diagram for the openMOS CPPS (openMOS, 2017)

Through a mechanism for periodically check if the physical resource is connected, based on heartbeat, the Resource Agent can easily represent if the resource is available or not (DP-011).

The cloud of agents delivered by the openMOS project is entirely FIPA compliant (DP-021) and it is responsible of abstracting all the available resources and the products being produced. With this information, the cyber system can compile the available capabilities and optimise the usage of the physical resources according to the characteristics of the products (DP-022).

Each physical resource is abstracted by one Resource Agent (DP-031) that contains all the information necessary to describe the physical resource within the cyber system (DP-034). The communication between the cyber system and the native system occurs through the Manufacturing Service Bus (MSB), developed under the openMOS project. Hence, the cyber system can abstract heterogeneous resources (DP-032), and since the communication with the physical resource is performed by an adapter which connects the MSB and the physical resource, the cyber system does not degrade the execution of the native system (DP-033).

Although the openMOS cyber system does not directly control the native system, it changes the behaviour of the system through the application of new recipes. Those recipes will obligate the adaptors to operate and control the execution of the native system. This solution is not very invasive, however, forces a considerable effort from the integration point of view. Hence, the DP-041 and DP-042 cannot be considered acceptable due to the effort necessary to adapt the control logic, in order to receive the orders from the adaptors, and the effort necessary to develop a specific adaptor, with all the logic, for each physical resource connected.

All the communications, without exceptions, between the cyber system and the native system, occur through the MSB. Hence, in order to deliver a flexible and reliable solution, the MSB and the cyber system provide a set of services which allow the communication between both. This solution fulfils the DP-043 and DP-044.

The human operator can verify the current state of the cyber system and partially the native system through a visual interface deployed in the cloud (DP-053). The proposed visual interface allows the operator to trigger the production of new products and helps during the addition of new physical resources (DP-051). Moreover, the visual interface does not interfere with the existent security, safety or other regulations (DP-052).

The cyber system, based on the approach defined for the openMOS platform, implies the implementation of complex adaptors for each physical resource. The adaptors are responsible for receiving the recipes triggered by the MAS and controlling the execution of the physical resource based on the recipes received (DP-061). This solution involves a massive effort regarding the integration between the native and cyber system. On the other hand, the integration with external software tools is quite easy due to the services available in the MSB (DP-062).

After analysing the openMOS solution and specifically the design and development of the agent-based cloud environment, Table 15 was filled.

Table 15 - openMOS evaluation

ID	DPApp	InvD	Relevance	DPR
DP-011	Very High	Low	1	0,744074888
DP-021	Very High	Very Low	1	0,950626357
DP-022	High	Very Low	1	0,749988368
DP-031	Very High	Low	1	0,744074888
DP-032	Very High	Very Low	1	0,950626357
DP-033	Very High	Low	1	0,744074888
DP-034	Very High	Low	1	0,744074888
DP-041	Medium	High	1	0,255925112
DP-042	Medium	Medium	1	0,5
DP-043	Very High	Very Low	1	0,950626357
DP-044	Very High	Very Low	1	0,950626357
DP-051	High	Very Low	1	0,749988368
DP-052	High	Low	1	0,744074888
DP-053	High	Very Low	1	0,749988368
DP-061	Medium	Medium	1	0,5
DP-062	Very High	Very Low	1	0,950626357
	Overall Score			
	Maximum Relevance		0,748712277	
	Assigned Relevance		0,748712277	

The openMOS solution, compared to the previous evaluations, obtained a quite high Overall Score. These results demonstrate that the proposed solution, in general, applies the defined DPs and the application of the DPs is not very invasive. The previously presented table demonstrates that only three DPs are not highly ranked.

Figure 72 summarises that most of the DPs are well followed. However, in one of the cases, it is recommended to be aware of the current implementation and if possible improve it. For the other two cases is recommended to reformulate the application of the DPs strongly.

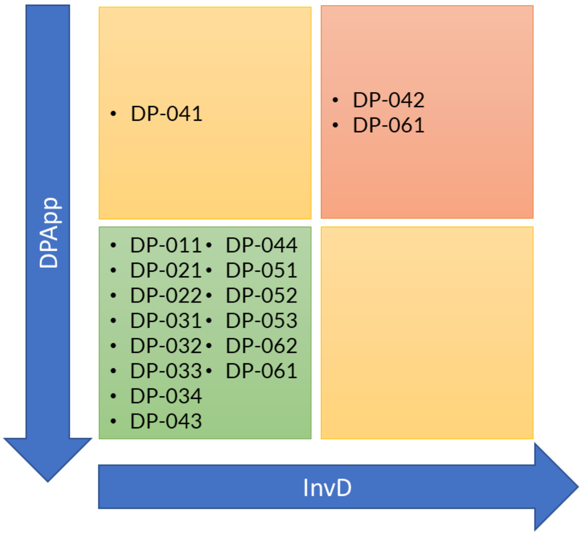


Figure 72 - DP positioning for the openMOS

Except for DP-041, DP-042 and DP-061, all the other DPs were considerably well implemented with a reduced InvD.

6.2.2. GO0D MAN

GO0D MAN is a European project funded by the European Commission under the EU Framework Programme for Research and Innovation Horizon 2020 (2014 - 2020) within the FoF – Technologies for Factories of the Future initiative.

The idea of the GO0D MAN project is to develop a cyber environment capable of to integrate the process and quality control in multi-stage manufacturing to achieve zero defect. To do so, an agent-based CPPS is proposed to integrate the shop-floor components with higher-level functionalities, such as services to provide big data analytics and knowledge management, as presented in Figure 73.

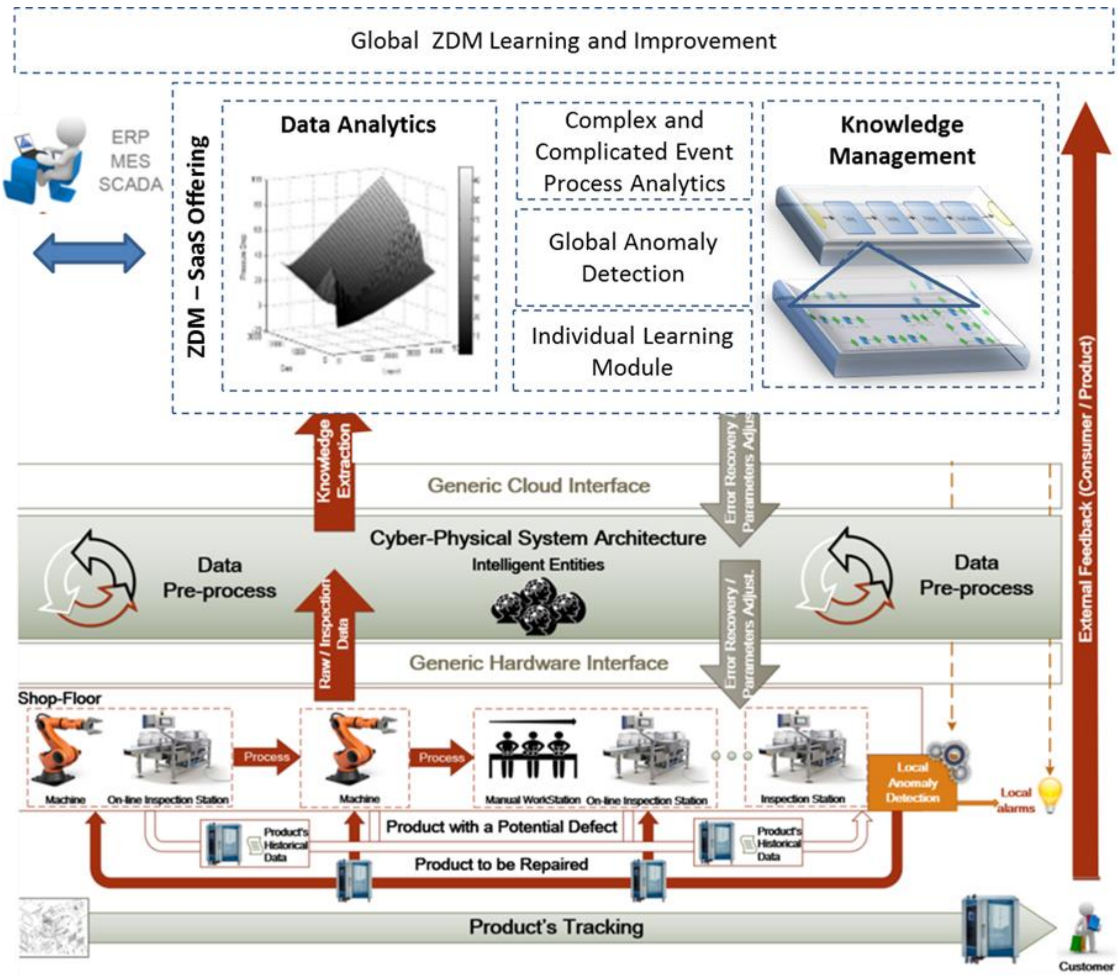


Figure 73 - GO0D MAN overall concept (GO0DMAN project, 2017a)

Since multi-stage production needs to deal with several interactions and dependencies from the different stations, it is imperative, from a quality inspection point of view, to correlate and analyse the data that could be extracted from the several stations and components operating at the shop-floor. Hence, the agent-based CPPS aims to verify the execution process and extract data from the production stations as well as the information from the quality inspection tools already running at the shop-floor. The quality inspection can be already running at in the existent native system or they can be added to the existent production line.

The proposed agent-based CPPS is composed of four generic agents (GO0DMAN project, 2017b):

- Product Agent: This agent is responsible for abstracting a physical product which is being produced. Hence, whenever a new product enters the production line, a new Product Agent is launched in order to abstract that product. This agent aims to control the product execution as well as collect all the information related to it, such as quality inspections, production steps, an so forth;

- Product Type Agent: Whenever a product variant or type can be produced in a specific production line, a new Product Type Agent must be launched to represent that possibility and create a template of a product which can be produced. This agent will also collect all the information from the Product Agents that follows the Product Type Agent template in order to have information about the production of each variant and possible problems;

- Resource Agent: The Resource Agent can be described as the physical agent of the GO0D MAN MAS. Whenever a new physical resource is running on the production line and it must be abstracted by an agent, a new Resource Agent is launched to do it. Since the resources running at the shop-floor can execute different tasks and that tasks can be performed by different types of actors, the Resource Agents, when launched, can be specified as Machine Agents, Quality Control Agents or Operator Agents;

- Independent Meta Agent: This agent has two main objectives, the first one is to guarantee the communication with external tools. In this case this agent constantly communicates with services provided by the data analytics and knowledge management tools. Moreover, at the CPPS level, the Independent Meta Agent works as a higher-level entity capable of optimising the performance of the CPPS. This agent, when necessary, triggers lower-level optimisations without depending from the higher-level tools. It receives the information from the other agents and tries to discover possible optimisations and improvements that could be applied to the other agents, mainly to the Resource Agents.

Figure 74 describes the UML class diagram for the GO0D MAN’s CPPS.

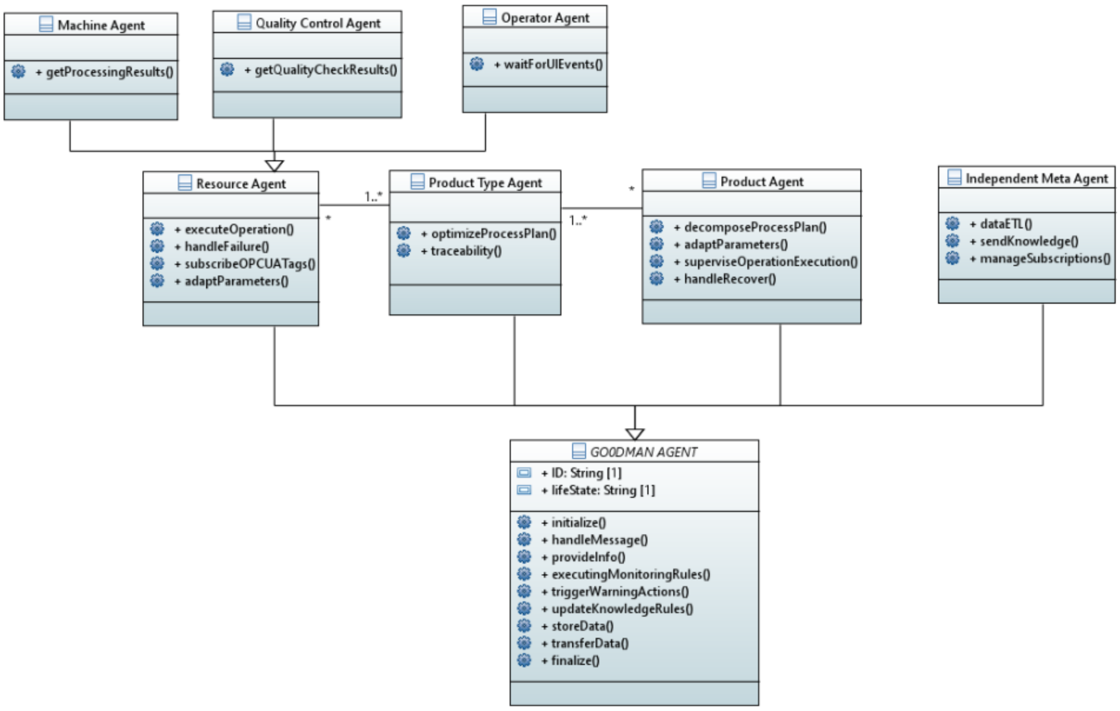


Figure 74 - Class diagram for the GO0D MAN CPPS (GO0DMAN project, 2017b)

The GO0D MAN platform automatically knows the availability of the physical resources due to the abstraction of the available resources through Resource Agents (DP-011). Hence, whenever a new component is added, the cyber system updates the capabilities offered by the system, combining the available capabilities (DP-022). All the communications among the agents (to update the capabilities, trigger executions, and so forth) are FIPA compliant (DP-021).

This solution aims to deliver a modular CPPS. To do so, each physical resource is abstracted by a unique Resource Agent (DP-031). Hence, the Resource Agent has been designed in a generic way capable of dealing with heterogeneous components (DP-032). However, the fact that the Resource Agent can have different specialisations (Operator Agent, Quality Control Agent and Machine Agent) forces the systems' integrators to have knowledge about the cyber system, which can constitute a barrier to the implementation of these systems. In order to allow the integration of these heterogeneous components, each Resource Agent is responsible for managing all the information necessary for the abstraction of the physical resource (DP-034), such as the physical location, capabilities, and so forth.

Since the module does not control the execution of the control logic directly (DP-042), the module does not degrade the execution of the physical resource (DP-033).

Regarding the DP-041, that stresses the reparameterization of the process without interfering with the existent control logic. For the GO0D MAN case, the physical agents trigger the execution of the tasks accordingly to the product process, this forces the native system to have the cyber system always running, in order to trigger the execution, even without directly writing and reading the IOs (DP-042).

The solution proposed by the GO0D MAN project uses the OPC-UA protocol. Hence, the DP-043 and DP-044 are fulfilled since the OPC-UA guarantees that the communication can occur in parallel and without permanent dependencies.

The GO0D MAN CPPS has a virtual interface, which can be used by the operator to visualise which components are available and which products and types of products are being produced (DP-053). Moreover, whenever the operator desires to launch a new product or type of products, this interface can be used for that purpose also (DP-051). The proposed visual interface does not influence any of the existent regulations (DP-052).

The GO0D MAN CPPS was designed with a set of services which allows the integration with external software tools (DP-062). Indeed, the CPPS was developed in order to interface an external tool capable of analysing big amounts of data and generate suggestions to be applied by the CPPS.

Regarding DP-061, a HAL must be proposed in order to define the boundary between the specific and the generic code. The Resource Agents instantiated in each solution are coded to interface with specific physical resources. This avoids problems with the development of libraries responsible for this integration. However, the solution is not easy to be integrated by systems integrators with knowledge in the platform and agent technology.

After analysing the GO0D MAN overall solution implementation and specifically the design and development of the agent-based CPPS, Table 16 was filled.

Table 16 – GO0D MAN evaluation

ID	DPApp	InvD	Relevance	DPR
DP-011	Very High	Low	1	0,744074888
DP-021	Very High	Very Low	1	0,950626357
DP-022	Very High	Very Low	1	0,950626357
DP-031	High	Low	1	0,744074888
DP-032	Medium	Medium	1	0,5
DP-033	Very High	Low	1	0,744074888
DP-034	Very High	Low	1	0,744074888
DP-041	Medium	Medium	1	0,5
DP-042	Very High	Low	1	0,744074888
DP-043	Very High	Very Low	1	0,950626357
DP-044	Very High	Very Low	1	0,950626357
DP-051	Very High	Very Low	1	0,950626357
DP-052	Very High	Very Low	1	0,950626357
DP-053	Very High	Very Low	1	0,950626357
DP-061	Very Low	Very Low	1	0,5
DP-062	Very High	Very Low	1	0,950626357
	Overall Score			
	Maximum Relevance		0,801586581	
	Assigned Relevance		0,801586581	

Comparing these results with the other results obtained so far, it is possible to verify that the GO0D MAN solution obtains a good Overall Score. However, it is possible to verify that in the application of few DPs it still being possible to improve the DPApp and reduce the InvD of such application, as shown in Figure 75.

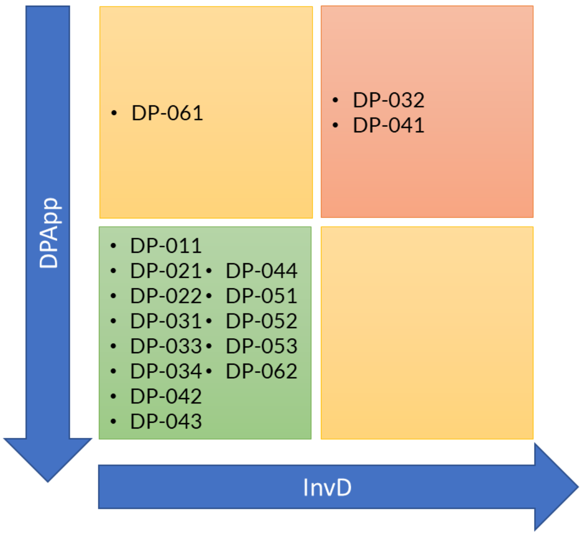


Figure 75 - DP positioning for the GO0D MAN

The diagram shows that DP-061, DP-032 and DP-041 must be revised in order to improve the overall solution. However, in two of them is recommended to revise as soon as possible, since the DPApp and also the InvD can be improved.

6.3. Discussion

With the evaluation of the experiments used to design the minimally invasive solutions, it is possible to verify which kind of Overall Scores can be defined as references for the proposed evaluation model.

Table 17 presents the results obtained for all the experiments and test cases. It is interesting to verify that the results obtained for the two test cases are not very different from the results obtained for the experiments developed during this research.

Table 17 - Solutions' evaluation overview

Experiments	Reconfigurable	Maximum Relevance	0,766312172
		With Relevance	0,766312172
	Self-Monitoring	Maximum Relevance	0,84339938
		With Relevance	0,892456434
	Self-Healing	Maximum Relevance	0,694359604
		With Relevance	0,694359604
Test Cases	openMOS	Maximum Relevance	0,748712277
		With Relevance	0,748712277
	GO0D MAN	Maximum Relevance	0,801586581
		With Relevance	0,801586581

This suggests, and since the two proposed test cases aim to deliver minimally invasive agent-based CPPS, that the proposed evaluation model can be used to evaluate such systems. However, it must be increased the number of test cases to be evaluated as well as the number of researchers, developers and systems integrators involved in the process of evaluating each test case. This will improve the evaluation of each solution.

Through the evaluation of the application of each DP, it is possible to verify that there exists some DPs that are not well used in several solutions. For instance, regarding DP-041, it is possible to verify, in Figure 76, that the application of this DP has a low ranking for three of the five cases (Self-monitoring, openMOS and GO0D MAN).

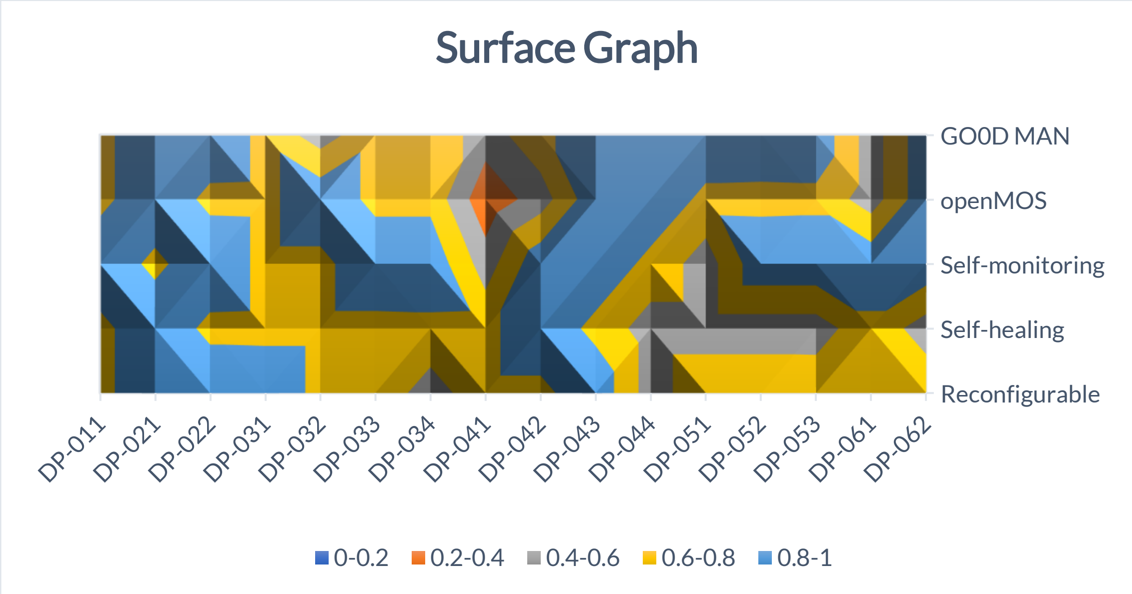


Figure 76 - DP rankings' surface graph

This situation can suggest that the application of this DP is one of the main challenges of developing these solutions.

In Figure 77, it is possible to verify that other DPs, as DP-044 or DP-061 are also challenging for most of the systems. However, in these cases, it is not so explicit as in the DP-041 case.

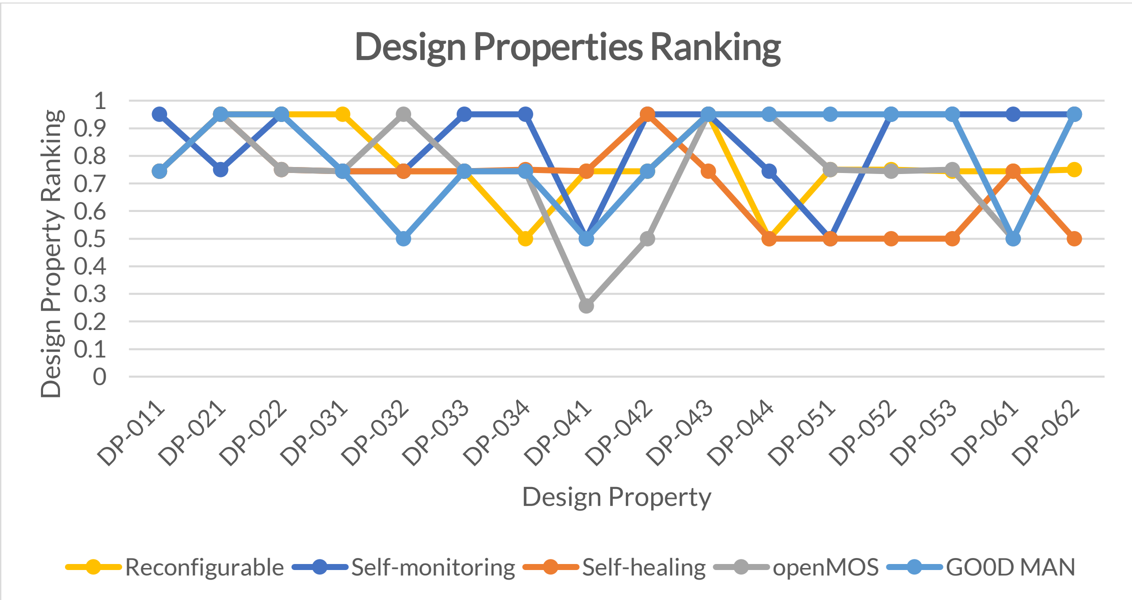


Figure 77 - Design Properties Ranking

Supporting the previous figures, the histogram, presented in Figure 78, helps to understand which DPs have the worst evaluation. In the histogram, it is possible to verify a sum of the DPRs for each DP as well as the DPRs for each case.

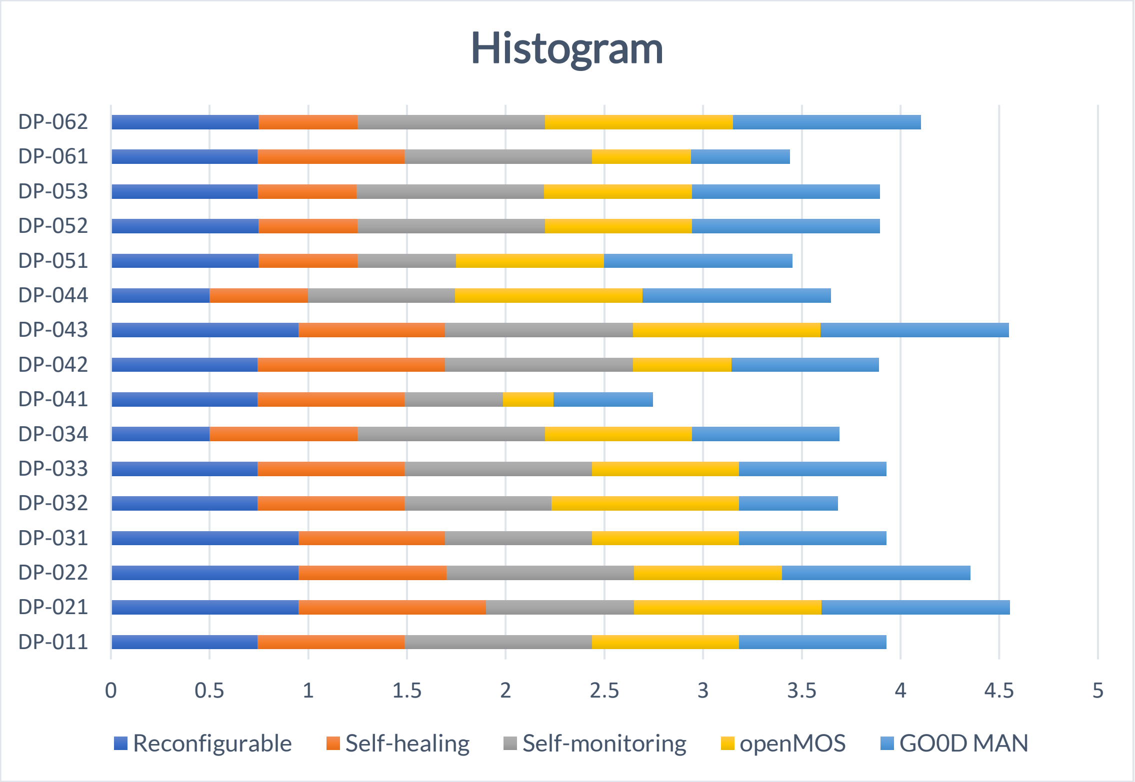


Figure 78 - DP rankings' histogram

As verified in the experiments and test cases, DP-041 and DP-44 are, in general, the most difficult DPs to be followed. However, it is possible to verify that some more DPs must be kept in mind, such as DP-061, DP-051 or DP-032.

7. Conclusions and Future Work

The last chapter summarises the work developed in this research and the obtained results. It verifies the contributions proposed at the beginning of the document. Moreover, it is presented the areas that the author believes that can be improved guiding the future activities. In the end, it is presented a list of the scientific contributions written during this work.

7.1. Summary of the Main Contributions

The proposed work aims to increase the adoption of the agent-based CPPS in real production environments through the design of generic FRs and DPs which should guide the development of such systems.

Hence, the document presented a deep literature review on this topic in order to verify which are the emergent trends regarding the development of agent-based systems and more specifically of CPPS managed by multi-agent environments. Analysing the state of the art, it was possible to verify which are the main barriers regarding the adoption of such systems (**C1**).

The definition of a minimally invasive solution was not consensual. Hence, a definition has been proposed and used to guide the research (**C2**). Afterwards, an interactive process took place in order to collect as much as possible useful information from possible consumers of the technology (European projects' test cases). That collected information combined with the literature review generated the FRs (**C3**). Two trees of FRs, one focused on the requirements which must be fulfilled by the newly developed software and another one with the requirements regarding the characteristics of the existent native system were proposed.

Based on the trees with the FRs, three experiments were designed (**C4**). These experiments were designed to deliver advanced functionalities as well as fulfil the generic requirements previously defined. For each experiment was designed a tree with the DPs used to guide the design and development of each solution. Merging the DPs used for the design of the three experiments, a tree of generic DPs was created (**C5**) This list of generic properties is one of the main contributions of the proposed research.

Although the proposed DPs emerged from an inductive process, it was not clear how to evaluate the applications. To overcome this gap, an evaluation model was defined in order to evaluate each solution qualitatively (**C6**). The proposed FIS based model can be used to evaluate different cases, and if necessary, the model can be updated in order to evaluate other DPs as input. To demonstrate the proposed model, two external cases were evaluated and compared with the proposed experiments.

It is interesting to verify, based on the results retrieved from the five evaluations, that most of the DPs are usually well applied, but for some of them it is not so common.

7.2. Confirmation of the Research Hypothesis

The challenges found at the beginning of the research were formalised as:

Q1: Which are the main design properties that should be considered when designing a minimally invasive agent based cyber-physical production system using traditional manufacturing environments with discrete control?

Q2: How it is possible to evaluate the design and development of a minimally invasive agent-based cyber-physical production system?

The hypothesis formulated to potential answer the proposed research questions are:

H1: If the developed agent-based cyber-physical production systems aimed to deliver new functionalities to the existent manufacturing systems with discrete control without requiring significant changes and adaptations share common properties and characteristics, it is possible to provide a list of design properties that must be followed in the development of such systems.

H2: I t is possible to evaluate the design and development of minimally invasive agent-based cyber-physical production system through the definition of an evaluation model capable of evaluating the implementation of each design property separately, and posteriorly evaluate the implementation based on the usage of all design properties. This model can be achieved through the development of a fuzzy inference system-based model.

The results demonstrated in section six indicate that a formulation of DPs for the design of minimally invasive agent-based CPPS can be a promising approach to guide the development of such systems and increase the adoption of these solutions. The fact that the two external cases (openMOS and GO0D MAN), with the objective of design minimally invasive agent-based CPPS, obtained equivalent scores, using the proposed evaluation model, indicates that at least in this two cases, most of the DPs were applied to develop such systems.

Regarding H1, it is possible to say that with the defined list of DPs it is possible to guide the development of less invasive agent-based CPPS systems, capable of bringing emergent functionalities, making usage of the new trends such as AI, BD processing, among others. The solutions delivered use a multi-agent approach and without requiring expertise in the technologies used in the solution.

Regarding H2 and keeping in mind that three experiments and two test cases were tested in section six, it is possible to say that the proposed model is capable of to evaluate different solutions. The evaluation model based on a FIS is capable of to evaluating different solutions according to correct application of the DPs and if the solutions proposed to apply the DPs are considered too invasive or not. The evaluation model was used to evaluate the following systems:

- Reconfiguration experiment: The system is able to reconfigure the control logic without requiring new hardware and a significant effort to adapt the existent control logic;
- Self-monitoring: The system is able to extract raw data, pre-process it and send that data to external tools performing advanced data analytics. This solution does not require any change on the existent control logic;
- Self-healing: The system is able to detect malfunctions and apply routines to overcome that states. The integration only requires a system's integrator without any expertise in agent-based technology;
- openMOS: The system is able to deal with modular production systems, highly flexible environments, with the ability to plug and unplug components at runtime. The agent-based performs production optimisation using the existent modules and applies recipes that are handled with existent technology, without requiring any agent expertise;

- GO0D MAN: The system aims to extract production process and quality check information. Afterwards, the MAS pre-processes that data and shares with higher level tools to try to discover quality issues. Whenever a quality issue is detected, the cyber system suggests possible root causes for the problem and apply if possible new parametrisations to the native system in order to solve the problem.

The different applications and solutions presented during this work indicate that the proposed evaluation model is generic and capable of evaluating different agent-based CPPS designed with slightly different objectives. This shows the scalability of the proposed approach beyond the application scenarios that motivated it.

7.3. Open Points and Future Activities

Despite the results obtained in this research and the amount of the experiments and test cases analysed, it is still worth to explore and refine some aspects of the work proposed in this document:

- Refinement and update of the FRs: Through a constant update regarding the literature review and the attendance of research activities such as research projects, it is possible to refine and improve the existent list of FRs;

- More experiments: The development of new experiments, with specific DPs for the experiments, will contribute to the refinement of the existent generic ones. The accuracy of the list of DPs depends on the amount of the experiments since the experiments are the main source of information to create and refine the generic DPs;

- Test cases: Beside the development of new experiments, it is important to analyse more external test cases. This will accurate the evaluation model and if possible extract relevant information also for the FRs and DPs improvement;

- Apply the evaluation model for other purposes: Since the evaluation model was designed to work with any kind of DPs. Could be interesting to verify the application of the model within a different list of DPs, in an optimal case, a list designed by external researchers and industrial oriented people;

- Evaluation of the test cases by external people: One of the most important steps in the future is the necessity to evaluate the same five cases, evaluated during this work, by external experts in order to get a bigger picture about the applications and verify the results with different points of view;

- Rules refinement: Similarly to the previous point, it is recommended to refine and revise the FIS's rules with a group of external experts.

7.4. Scientific contributions and peer validations

Based on the presented work, several publications were prepared:

7.4.1. Related to Section 3

International Journals:

- Ribeiro, Luis; Rocha, Andre; Veiga, Angelo; Barata, José; Collaborative routing of products using a self-organizing mechatronic agent framework—A simulation study, Computers in Industry,68,27-39,2015, Elsevier.

Conferences with peer revision:

- Ribeiro, Luis; Rocha, Andre; Barata, Jose; A product handling technical architecture for multiagent-based mechatronic systems, IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society,4342-4347,2012, IEEE;

- Ribeiro, Luis; Rocha, Andre; Barata, Jose; A study of JADE's messaging RTT performance using distinct message exchange patterns, "Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE",7410-7415,2013, IEEE.

Book chapters:

- Rocha, Andre; Ribeiro, Luis; Barata, José; A multi agent architecture to support self-organizing material handling, "Doctoral Conference on Computing, Electrical and Industrial Systems", 93-100,2014, "Springer, Berlin, Heidelberg";

- Peixoto, João Alvarez; Oliveira, José Antonio Barata; Rocha, André Dionisio; Pereira, Carlos Eduardo; The migration from conventional manufacturing systems for multi-agent paradigm: The first step, "Doctoral Conference on Computing, Electrical and Industrial Systems", 111-118,2015, "Springer, Cham";

- Di Orio, Giovanni; Barata, Diogo; Rocha, André; Barata, José; A cloud-based infrastructure to support manufacturing resources composition, "Doctoral Conference on Computing, Electrical and Industrial Systems", 82-89, 2015, "Springer, Cham".

7.4.2. Related to Section 4

International Journals:

- Antzoulatos, Nikolas; Castro, Elkin; de Silva, Lavindra; Rocha, André Dionisio; Ratchev, Svetan; Barata, José; A multi-agent framework for capability-based reconfiguration of industrial assembly systems,International Journal of Production Research, 55, 10, 2950-2960, 2017, Taylor & Francis;

- Rocha, Andre Dionisio; Lima-Monteiro, Pedro; Parreira-Rocha, Mafalda; Barata, Jose; Artificial immune systems based multi-agent architecture to perform distributed diagnosis, Journal of Intelligent Manufacturing, 1-13, 2017, Springer US.

Conferences with peer revision:

- Rocha, Andre; Di Orio, Giovanni; Barata, Jose; Antzoulatos, Nikolas; Castro, Elkin; Scrimieri, Daniele; Ratchev, Svetan; Ribeiro, Luis; ,An agent based framework to support plug and produce, "Industrial Informatics (INDIN), 2014 12th IEEE International Conference on", 504-510, 2014, IEEE;

- Di Orio, Giovanni; Rocha, André; Ribeiro, Luís; Barata, José; The prime semantic language: Plug and produce in standard-based manufacturing production systems, The International Conference on Flexible Automation and Intelligent Manufacturing (FAIM 2015), 8, 2015;

- Antzoulatos, Nikolas; Rocha, André; Castro, Elkin; de Silva, Lavindra; Santos, Tiago; Ratchev, Svetan; Barata, José; Towards a Capability-based

Framework for Reconfiguring Industrial Production Systems, IFAC-PapersOnLine, 48, 3, 2077-2082, 2015, Elsevier;

- Rocha, Andre Dionisio; Peres, Ricardo; Barata, Jose; An agent based monitoring architecture for plug and produce based manufacturing systems, "Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on", 1318-1323, 2015, IEEE;

- Rocha, Andre Dionisio; Monteiro, Pedro Lima; Barata, Jose; An artificial immune systems based architecture to support diagnoses in evolvable production systems using genetic algorithms as an evolution enabler, Flex. Autom. Intell. Manuf, 25, 2015;

- Rocha, André Dionisio; Peres, Ricardo Silva; Flores, Luis; Barata, Jose; A multiagent based knowledge extraction framework to support plug and produce capabilities in manufacturing monitoring systems, "Mechatronics and its Applications (ISMA), 2015 10th International Symposium on", 1-5, 2015, IEEE;

- Santos, Tiago; Ribeiro, Luis; Rocha, Andre Dionisio; Barata, Jose; A system reconfiguration architecture for hybrid automation systems based in agents and programmable logic controllers, "Industrial Informatics (INDIN), 2016 IEEE 14th International Conference on", 98-103, 2016, IEEE.

Book chapters:

- Rocha, André Dionísio; Barata, Diogo; Di Orio, Giovanni; Santos, Tiago; Barata, José; ,Prime as a generic agent based framework to support pluggability and reconfigurability using different technologies, "Doctoral Conference on Computing, Electrical and Industrial Systems", 101-110, 2015, "Springer, Cham";

- Rocha, Andre Dionisio; Caetano, Pedro; Oliveira, Jose Barata; A Generic Reconfigurable and Pluggable Material Handling System Based on Genetic Algorithm, Service Orientation in Holonic and Multi-Agent Manufacturing, 103-113, 2017, "Springer, Cham";

- Peres, Ricardo Silva; Rocha, Andre Dionisio; Coelho, Andre; Oliveira, Jose Barata; ,"A Highly Flexible, Distributed Data Analysis Framework for

Industry 4.0 Manufacturing Systems", Service Orientation in Holonic and Multi-Agent Manufacturing, 373-381, 2017, "Springer, Cham";

- Peres, Ricardo Silva; Rocha, Andre Dionisio; Barata, Jose; Dynamic Simulation for MAS-Based Data Acquisition and Pre-processing in Manufacturing Using V-REP, "Doctoral Conference on Computing, Electrical and Industrial Systems", 125-134, 2017, "Springer, Cham";

- Lima-Monteiro, Pedro; Parreira-Rocha, Mafalda; Rocha, André Dionisio; Oliveira, Jose Barata; ,Big Data Analysis to Ease Interconnectivity in Industry 4.0—A Smart Factory Perspective, Service Orientation in Holonic and Multi-Agent Manufacturing, 237-245, 2017, "Springer, Cham".

7.4.3. Related to Section 5

Conferences with peer revision:

- Peres, Ricardo Silva; Parreira-Rocha, Mafalda; Rocha, Andre Dionisio; Barbosa, José; Leitão, Paulo; Barata, José; ,Selection of a data exchange format for industry 4.0 manufacturing systems, "Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE", 5723-5728, 2016, IEEE.

Book chapters:

- Lima-Monteiro, Pedro; Parreira-Rocha, Mafalda; Rocha, André Dionisio; Oliveira, Jose Barata; Big Data Analysis to Ease Interconnectivity in Industry 4.0—A Smart Factory Perspective, Service Orientation in Holonic and Multi-Agent Manufacturing, 237-245, 2017, "Springer, Cham".

8. References

Adolphs, P., Bedenbender, H., Dirzus, D., Ehlich, M., Epple, U., Hankel, M., Heidel, R., Hoffmeister, M., Huhle, H., Kärcher, B., 2015. Status Report-Reference Architecture Model Industrie 4.0 (RAMI4. 0). VDI-Ver. Dtsch. Ingenieure EV ZVEI-Ger. Electr. Electron. Manuf. Assoc. Tech Rep.

Amin, S.M., Wollenberg, B.F., 2005. Toward a smart grid: power delivery for the 21st century. IEEE Power Energy Mag. 3, 34–41. https://doi.org/10.1109/MPAE.2005.1507024

Andreev, M., Ivaschenko, A., Skobelev, P., Tsarev, A., 2010. A Multi-Agent Platform Design for Adaptive Networks of Intelligent Production Schedulers. IFAC Proc. Vol., 10th IFAC Workshop on Intelligent Manufacturing Systems 43, 78–83. https://doi.org/10.3182/20100701-2-PT-4011.00015

Antzoulatos, N., Castro, E., Scrimieri, D., Ratchev, S., 2014. A multi-agent architecture for plug and produce on an industrial assembly platform. Prod. Eng. 8, 773–781. https://doi.org/10.1007/s11740-014-0571-x

Babiceanu, R.F., Seker, R., 2016. Big Data and virtualization for manufacturing cyber-physical systems: A survey of the current status and future outlook. Comput. Ind. 81, 128–137. https://doi.org/10.1016/j.compind.2016.02.004

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

Bellifemine, F., Poggi, A., Rimassa, G., 1999. JADE–A FIPA-compliant agent framework, in: Proceedings of PAAM. London, p. 33.

Blanc, P., Demongodin, I., Castagna, P., 2008. A holonic approach for manufacturing execution system design: An industrial application. Eng. Appl. Artif. Intell. 21, 315–330. https://doi.org/10.1016/j.engappai.2008.01.007

Bruckner, S., Wyns, J., Peeterst, P., Kollingbaum, M., 1998. Designing agents for manufacturing control. Presented at the Proc. Artificial Intelligence and Manufacturing Research Planning Workshop—State of the Art & State of the Practice, AAAI Press, Albuquerque, pp. 40–46.

Bussmann, S., Schild, K., 2001. An agent-based approach to the control of flexible production systems, in: ETFA 2001. 8th International Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No.01TH8597). Presented at the ETFA 2001. 8th International Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No.01TH8597), pp. 481–488 vol.2. https://doi.org/10.1109/ETFA.2001.997722

Camarinha-Matos, L.M., Vieira, W., 1999. Intelligent mobile agents in elderly care. Robot. Auton. Syst. 27, 59–75. https://doi.org/10.1016/S0921-8890(98)00083-9

Cândido, G., Barata, J., 2007. A Multiagent Control System for Shop Floor Assembly, in: Holonic and Multi-Agent Systems for Manufacturing. Springer, Berlin, Heidelberg, pp. 293–302. https://doi.org/10.1007/978-3-540-74481-8_28

Consortium, I.I., 2015. Industrial internet reference architecture. Ind. Internet Consort. Tech Rep June.

Cowling, P.I., Ouelhadj, D., Petrovic, S., 2003. A multi-agent architecture for dynamic scheduling of steel hot rolling. J. Intell. Manuf. 14, 457–470. https://doi.org/10.1023/A:1025701325275

Danny, P., Ferreira, P., Lohse, N., Guedes, M., 2017. An AutomationML model for plug-and-produce assembly systems. © IEEE. https://doi.org/10.1109/INDIN.2017.8104883

de Oliveira, J.A.B., 2003. Coalition Based Approach for Shop Floor Agility–A Multiagent Approach. Universidade Nova de Lisboa.

Deloitte, 2014. Industry 4.0–challenges and solutions for the digital transformation and use of exponential technologies. White Pap.

Di Orio, G., Cândido, G., Barata, J., 2015. The Adapter module: A building block for Self-Learning Production Systems. Robot. Comput.-Integr. Manuf., Sustaining Resilience in Today's Demanding Environments 36, 25–35. https://doi.org/10.1016/j.rcim.2014.12.007

Dias-Ferreira, J., Ribeiro, L., Akillioglu, H., Neves, P., Onori, M., 2016. BIOSOARM: a bio-inspired self-organising architecture for manufacturing cyber-physical shopfloors. J. Intell. Manuf. 1–24. https://doi.org/10.1007/s10845-016-1258-2

Djurdjanovic, D., Lee, J., Ni, J., 2003. Watchdog Agent—an infotronics-based prognostics approach for product performance degradation assessment and prediction. Adv. Eng. Inform. 17, 109–125. https://doi.org/10.1016/j.aei.2004.07.005

Farhangi, H., 2010. The path of the smart grid. IEEE Power Energy Mag. 8, 18–28. https://doi.org/10.1109/MPE.2009.934876

Farid, A.M., Ribeiro, L., 2015. An axiomatic design of a multiagent reconfigurable mechatronic system architecture. IEEE Trans. Ind. Inform. 11, 1142–1155. https://doi.org/10.1109/TII.2015.2470528

Fletcher, M., McFarlane, D., Thorne, A., Jarvis, D., Lucas, A., 2003. Evaluating a Holonic Packing Cell, in: Holonic and Multi-Agent Systems for Manufacturing. Presented at the International Conference on Industrial Applications of Holonic and Multi-Agent Systems, Springer, Berlin, Heidelberg, pp. 246–257. https://doi.org/10.1007/978-3-540-45185-3_23

Fox, M.S., Chionglo, J.F., Barbuceanu, M., 1993. The integrated supply chain management system. Technical report, Department of Industrial Engineering, University of Toronto.

Galaske, N., Anderl, R., 2016. Disruption Management for Resilient Processes in Cyber-physical Production Systems. Procedia CIRP, 26th CIRP Design Conference 50, 442–447. https://doi.org/10.1016/j.procir.2016.04.144

Gilchrist, A., 2016. Introducing Industry 4.0, in: Industry 4.0. Apress, pp. 195–215. https://doi.org/10.1007/978-1-4842-2047-4_13

GO0DMAN project, 2017a. Deliverable D1.2, ZDM Management Methodology.

GO0DMAN project, 2017b. Deliverable D2.1, Multi-Agent Architecture Specification.

Goddard, W., Melville, S., 2004. Research Methodology: An Introduction. Juta and Company Ltd.

Hehenberger, P., Vogel-Heuser, B., Bradley, D., Eynard, B., Tomiyama, T., Achiche, S., 2016. Design, modelling, simulation and integration of cyber physical systems: Methods and applications. Comput. Ind. 82, 273–289. https://doi.org/10.1016/j.compind.2016.05.006

Hu, F., Lu, Y., Vasilakos, A.V., Hao, Q., Ma, R., Patil, Y., Zhang, T., Lu, J., Li, X., Xiong, N.N., 2016. Robust Cyber–Physical Systems: Concept, models, and implementation. Future Gener. Comput. Syst. 56, 449–475. https://doi.org/10.1016/j.future.2015.06.006

Iung, B., 2003. From remote maintenance to MAS-based e-maintenance of an industrial process. J. Intell. Manuf. 14, 59–82. https://doi.org/10.1023/A:1022239329544

Jacobi, S., Madrigal-Mora, C., León-Soto, E., Fischer, K., 2005. AgentSteel: An Agent-based Online System for the Planning and Observation of Steel Production, in: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS '05. ACM, New York, NY, USA, pp. 114–119. https://doi.org/10.1145/1082473.1082813

Jarvis, D.H., Jarvis, J.H., 2003. Holonic Diagnosis for an Automotive Assembly Line, in: Deen, S.M. (Ed.), Agent-Based Manufacturing, Advanced Information Processing. Springer Berlin Heidelberg, pp. 193–206. https://doi.org/10.1007/978-3-662-05624-0_9

Kagermann, H., Helbig, J., Hellinger, A., Wahlster, W., 2013. Recommendations for Implementing the strategic initiative INDUSTRIE 4.0: securing the future of German manufacturing industry; final report of the Industrie 4.0 working group. Forschungsunion.

Krumeich, J., Jacobi, S., Werth, D., Loos, P., 2014. Big Data Analytics for Predictive Manufacturing Control - A Case Study from Process Industry, in: 2014 IEEE International Congress on Big Data. pp. 530–537. https://doi.org/10.1109/BigData.Congress.2014.83

Lastra, J.L.M., Torres, E.L., Colombo, A.W., 2005. A 3D Visualization and Simulation Framework for Intelligent Physical Agents, in: Holonic and Multi-Agent Systems for Manufacturing. Presented at the International Conference on Industrial Applications of Holonic and Multi-Agent Systems, Springer, Berlin, Heidelberg, pp. 23–38. https://doi.org/10.1007/11537847_3

Lee, E.A., 2008. Cyber Physical Systems: Design Challenges, in: 2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC). Presented at the 2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC), pp. 363–369. https://doi.org/10.1109/ISORC.2008.25

Lee, J., Bagheri, B., Kao, H.-A., 2015. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. Manuf. Lett. 3, 18–23. https://doi.org/10.1016/j.mfglet.2014.12.001

Lee, J., Lapira, E., Bagheri, B., Kao, H., 2013. Recent advances and trends in predictive manufacturing systems in big data environment. Manuf. Lett. 1, 38–41. https://doi.org/10.1016/j.mfglet.2013.09.005

Leitão, P., 2009. Agent-based distributed manufacturing control: A state-of-the-art survey. Eng. Appl. Artif. Intell., Distributed Control of Production Systems 22, 979–991. https://doi.org/10.1016/j.engappai.2008.09.005

Leitão, P., Barbosa, J., Pereira, A., Barata, J., Colombo, A.W., 2016a. 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., Colombo, A.W., Karnouskos, S., 2016b. Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges. Comput. Ind., Emerging ICT concepts for smart, safe and sustainable industrial systems 81, 11–25. https://doi.org/10.1016/j.compind.2015.08.004

Leitao, P., Karnouskos, S., Ribeiro, L., Lee, J., Strasser, T., Colombo, A.W., 2016. Smart Agents in Industrial Cyber–Physical Systems. Proc. IEEE 104, 1086–1101. https://doi.org/10.1109/JPROC.2016.2521931

Leitão, P., Restivo, F., 2008. A holonic approach to dynamic manufacturing scheduling. Robot. Comput.-Integr. Manuf., BASYS '06: Balanced Automation Systems 7th IFIP International Conference on Information Technology for Balanced Automation Systems in Manufacturing and Services 24, 625–634. https://doi.org/10.1016/j.rcim.2007.09.005

Leitão, P., Rodrigues, N., Turrin, C., Pagani, A., 2015. Multiagent System Integrating Process and Quality Control in a Factory Producing Laundry Washing Machines. IEEE Trans. Ind. Inform. 11, 879–886. https://doi.org/10.1109/TII.2015.2431232

Lüder, A., Peschke, J., Sauter, T., Deter, S., Diep, D., 2004. Distributed intelligence for plant automation based on multi-agent systems: the PABADIS approach. Prod. Plan. Control 15, 201–212. https://doi.org/10.1080/09537280410001667484

Mankins, J.C., 1995. Technology readiness levels.

Mařík, V., Lažanský, J., 2007. Industrial applications of agent technologies. Control Eng. Pract., Special Issue on Manufacturing Plant Control: Challenges and IssuesINCOM 200411th IFAC INCOM'04 Symposium on Information Control Problems in Manufacturing 15, 1364–1380. https://doi.org/10.1016/j.conengprac.2006.10.001

Marik, V., McFarlane, D., 2005. Industrial adoption of agent-based technologies. IEEE Intell. Syst. 20, 27–35. https://doi.org/10.1109/MIS.2005.11

Mařík, V., Vrba, P., Hall, K.H., Maturana, F.P., 2005. Rockwell Automation Agents for Manufacturing, in: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS '05. ACM, New York, NY, USA, pp. 107–113. https://doi.org/10.1145/1082473.1082812

Markowski, A.S., Mannan, M.S., 2008. Fuzzy risk matrix. J. Hazard. Mater. 159, 152–157. https://doi.org/10.1016/j.jhazmat.2008.03.055

Marseu, E., Kolberg, D., Birtel, M., Zühlke, D., 2016. Interdisciplinary Engineering Methodology for changeable Cyber-Physical Production Systems. IFAC-Pap., 12th IFAC Workshop on Intelligent Manufacturing Systems IMS 2016 49, 85–90. https://doi.org/10.1016/j.ifacol.2016.12.166

Mobley, R.K., 2002. An Introduction to Predictive Maintenance. Butterworth-Heinemann.

Mönch, L., Stehli, M., Zimmermann, J., 2003. FABMAS: An Agent-Based System for Production Control of Semiconductor Manufacturing Processes, in: Holonic and Multi-Agent Systems for Manufacturing. Presented at the International Conference on Industrial Applications of Holonic and Multi-Agent Systems, Springer, Berlin, Heidelberg, pp. 258–267. https://doi.org/10.1007/978-3-540-45185-3_24

Monostori, L., 2014. Cyber-physical Production Systems: Roots, Expectations and R&D Challenges. Procedia CIRP, Variety Management in Manufacturing 17, 9–13. https://doi.org/10.1016/j.procir.2014.03.115

Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W., Ueda, K., 2016. Cyber-physical systems in manufacturing.

CIRP Ann. - Manuf. Technol. 65, 621–641. https://doi.org/10.1016/j.cirp.2016.06.005

Monostori, L., Váncza, J., Kumara, S.R.T., 2006. Agent-Based Systems for Manufacturing. CIRP Ann. - Manuf. Technol. 55, 697–720. https://doi.org/10.1016/j.cirp.2006.10.004

Onori, M., Barata, J., Frei, R., 2006. Evolvable Assembly Systems Basic Principles, in: Information Technology For Balanced Manufacturing Systems, IFIP International Federation for Information Processing. Springer US, pp. 317–328. https://doi.org/10.1007/978-0-387-36594-7_34

Onori, M., Lohse, N., Barata, J., Hanisch, C., 2012. The IDEAS project: plug & produce at shop-floor level. Assem. Autom. 32, 124–134. https://doi.org/10.1108/01445151211212280

openMOS, 2017. Deliverable: D4.1, Open System Platform.

openMOS, 2016. Deliverable: D2.1, Plug & Produce Automation Device Specification.

Pach, C., Berger, T., Bonte, T., Trentesaux, D., 2014. ORCA-FMS: a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling. Comput. Ind. 65, 706–720. https://doi.org/10.1016/j.compind.2014.02.005

Pechoucek, M., Riha, A., Vokrinek, J., Marik, V., Prazma, V., 2002. ExPlanTech: Applying multi-agent systems in production planning. Int. J. Prod. Res. 40, 3681–3692. https://doi.org/10.1080/00207540210140086

Peres, R.S., Parreira-Rocha, M., Rocha, A.D., Barbosa, J., Leitão, P., Barata, J., 2016. Selection of a data exchange format for industry 4.0 manufacturing systems, 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. 5723–5728. https://doi.org/10.1109/IECON.2016.7793750

Prieto-Sandoval, V., Jaca, C., Ormazabal, M., 2017. Towards a consensus on the circular economy. J. Clean. Prod. https://doi.org/10.1016/j.jclepro.2017.12.224

Psaier, H., Dustdar, S., 2010. A survey on self-healing systems: approaches and systems. Computing 91, 43–73. https://doi.org/10.1007/s00607-010-0107-y

Ribeiro, L., 2017. Cyber-physical production systems' design challenges, in: 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE). Presented at the 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), pp. 1189–1194. https://doi.org/10.1109/ISIE.2017.8001414

Ribeiro, L., Barata, J., Onori, M., Hanisch, C., Hoos, J., Rosa, R., 2011. Self-organization in automation - the IDEAS pre-demonstrator, in: IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society. pp. 2752–2757. https://doi.org/10.1109/IECON.2011.6119747

Ribeiro, L., Rocha, A., Veiga, A., Barata, J., 2015. Collaborative routing of products using a self-organizing mechatronic agent framework—A simulation study. Comput. Ind. 68, 27–39. https://doi.org/10.1016/j.compind.2014.12.003

Rocha, A., Di Orio, G., Barata, J., Antzoulatos, N., Castro, E., Scrimieri, D., Ratchev, S., Ribeiro, L., 2014. An agent based framework to support plug and produce, in: Industrial Informatics (INDIN), 2014 12th IEEE International Conference on. IEEE, pp. 504–510. https://doi.org/10.1109/INDIN.2014.6945565

Rocha, A.D., Peres, R., Barata, J., 2015. An agent based monitoring architecture for plug and produce based manufacturing systems, in: 2015 IEEE 13th International Conference on Industrial Informatics (INDIN). IEEE, pp. 1318–1323. https://doi.org/10.1109/INDIN.2015.7281926

Rocha, A.D.B. da S., 2013. An agent based architecture for material handling systems. Faculdade de Ciências e Tecnologia.

Sauer, O., 2004. Modern production monitoring in automotive plants, in: Proceedings of the FISITA 2004 World Automotive Congress, Barcelona, Spain. pp. 23–27.

Scholten, B., 2007. The Road to Integration: A Guide to Applying the ISA-95 Standard in Manufacturing. ISA.

Schuh, G., Potente, T., Wesch-Potente, C., Weber, A.R., Prote, J.-P., 2014. Collaboration Mechanisms to Increase Productivity in the Context of Industrie 4.0. Procedia CIRP, 2nd CIRP Robust Manufacturing Conference (RoMac 2014) 19, 51–56. https://doi.org/10.1016/j.procir.2014.05.016

Shpilevoy, V., Shishov, A., Skobelev, P., Kolbova, E., Kazanskaia, D., Shepilov, Y., Tsarev, A., 2013. Multi-agent system “Smart Factory” for real-time workshop management in aircraft jet engines production. IFAC Proc. Vol. 46, 204–209. https://doi.org/10.3182/20130522-3-BR-4036.00025

Shrouf, F., Ordieres, J., Miragliotta, G., 2014. Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm, in: 2014 IEEE International Conference on Industrial Engineering and Engineering Management. pp. 697–701. https://doi.org/10.1109/IEEM.2014.7058728

Shrouf, Fadi, Ordieres-Meré, J., García-Sánchez, A., Ortega-Mier, M., 2014. Optimizing the production scheduling of a single machine to minimize total energy consumption costs. J. Clean. Prod. 67, 197–207. https://doi.org/10.1016/j.jclepro.2013.12.024

Snowden, D., 2002. Complex acts of knowing: paradox and descriptive self-awareness. J. Knowl. Manag. 6, 100–111. https://doi.org/10.1108/13673270210424639

Stock, T., Seliger, G., 2016. Opportunities of Sustainable Manufacturing in Industry 4.0. Procedia CIRP, 13th Global Conference on Sustainable Manufacturing – Decoupling Growth from Resource Use 40, 536–541. https://doi.org/10.1016/j.procir.2016.01.129

Thiede, S., Juraschek, M., Herrmann, C., 2016. Implementing Cyber-physical Production Systems in Learning Factories. Procedia CIRP, 6th CIRP Conference on Learning Factories 54, 7–12. https://doi.org/10.1016/j.procir.2016.04.098

Ueda, K., 1992. A Concept for Bionic Manufacturing Systems Based on DNA-type Information, in: Proceedings of the IFIP TC5 / WG5.3 Eight International PROLAMAT Conference on Human Aspects in Computer Integrated Manufacturing. North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands, pp. 853–863.

Ueda, K., Vaario, J., Ohkura, K., 1997. Modelling of Biological Manufacturing Systems for Dynamic Reconfiguration. CIRP Ann. - Manuf. Technol. 46, 343–346. https://doi.org/10.1016/S0007-8506(07)60839-7

Utz, W., Lee, M., 2017. Industrial Business Process Management Using Adonis Towards a Modular Business Process Modelling Method for Zero-Defect-Manufacturing, in: Industrial Engineering, Management Science and Application (ICIMSA), 2017 International Conference on. IEEE, pp. 1–5. https://doi.org/10.1109/ICIMSA.2017.7985590

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

Van Dyke Parunak, H., Baker, A.D., Clark, S.J., 2001. The AARIA agent architecture: From manufacturing requirements to agent-based system design. Integr. Comput.-Aided Eng. 8, 45–58.

Vrba, P., 2003. MAST: manufacturing agent simulation tool, in: EFTA 2003. 2003 IEEE Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No.03TH8696). Presented at the EFTA 2003. 2003 IEEE Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No.03TH8696), pp. 282–287 vol.1. https://doi.org/10.1109/ETFA.2003.1247718

Wada, H., Sakuraba, Y., Negishi, M., 1998. Machinery control system using autonomous agents. Yokogawa Electr. Corp. Tokyo Jpn.

Wang, S., Wan, J., Zhang, D., Li, D., Zhang, C., 2016. Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination. Comput. Netw., Industrial Technologies and Applications for the Internet of Things 101, 158–168. https://doi.org/10.1016/j.comnet.2015.12.017

Yue, X., Cai, H., Yan, H., Zou, C., Zhou, K., 2015. Cloud-assisted industrial cyber-physical systems: An insight. Microprocess. Microsyst. 39, 1262–1270. https://doi.org/10.1016/j.micpro.2015.08.013

Zuehlke, D., 2010. SmartFactory—Towards a factory-of-things. Annu. Rev. Control 34, 129–138. https://doi.org/10.1016/j.arcontrol.2010.02.008