The Digital Manufacturing and Design (DiManD)

Grant agreement No 814078– H2020-MSCA-ITN European Training Network Grant

Deliverable 3.2

CPS Architecture
Jun 2022

Lead parties for Deliverable : KTH Royal Institute of Technology
Deliverable due date : M38
Actual submission date : M39
Dissemination level : Public

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 814078

Table of Content

Acronyms

AADL Architecture Analysis and Design Language. 27

AAS Asset Administration Shell. 15

AGV Automated Guided Vehicle. 20, 21

AML Automation Markup Language. 14, 15

AutomationML Automation Markup Language. 27

BM Business Models. 25

BSo1 Batch Size of One. 25

CAN Controller Area Network. 27

CCD charge-coupled device. 27

CMOS complementary metal oxide semiconductor. 27

CPPS Cyber-Physical production systems. 8, 15, 16

CPS Cyber-Physical System. 6, 7, 10, 14, 15, 28, 30

HMI Human Machine Interface. 27

HTTP Hypertext Transfer Protocol. 24

IIoT Industrial Internet of Things. 27

IoT Internet of things. 27

IT Information Technology. 6

MA mechatronic agent. 16

MAPE-K Monitor-Analyze-Plan-Execute over a Knowledge. 8, 13, 18–24, 28, 30

MARTE Modeling and Analysis of Real-Time and Embedded systems. 27

MOST Media Oriented Systems Transport. 27

NFC Near Field Communication. 27

OKP One-of-a-Kind Production. 25

OPC UA Open Platform Communications United Architecture. 14, 24, 27

OT Operational technologies. 6

PCA Principal Component Analysis. 17

PnP Plug & Produce. 15, 25

PTO Plug to Order. 25

Rami 4.0 Reference Architectural Model Industrie 4.0. 6, 8, 11, 18, 19, 24, 25, 28, 30

RFID Radio Frequency Identification. 27

SI system Integrator. 19

SOA Service-oriented architecture. 6, 15, 24

UML Unified Modeling Language. 27

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 814078

Summary

This deliverable continues previous requirement analysis that an adaptable/evolvable/autonomous and sustainable production system may pose. In parallel to the autonomous computing definition, various self-x modules were defined as critical to form truly autonomous production systems -i.e., self-configuration, self-organisation, self-diagnosis,and self-learning. Subsequently, a guidance and common practices are presented for the implementation of Cyber-Physical production Systems in industrial scenarios following this modular construction of autonomous systems. In summary, this work focuses on the definition of a framework that lists sequential steps to create modular autonomy in manufacturing -self-configuration and self-diagnosis. Therefore, main enabling technologies of self-configuration and self-diagnosis behaviours, compliant with the concept of RAMI4.0, are defined as well.

Team involved in deliverable writing:

ESR 3: Angela Carrera, University of Mondragon
ESR 4: Ngoc Hien Nguyen, University of Mondragon
ESR 5: Miriam Ugarte, University of Mondragon
ESR 6: Trunal Patil, STIIMA
ESR 7: Jose Mulet, STIIMA
ESR 10: Luis Estrada-Jimenez UNINOVA

Supervisors : Prof. Antonion Maffei, KTH

Chapter 1

Introduction

The fourth industrial revolution arises as a new paradigm that promotes smart manufacturing processes through the interconnection of diverse devices and technologies such as embedded sensors, autonomous robots, cloud systems and so forth. The interconnection of these applications allows for real-time collection of production data from low level devices upwards to enterprise applications, to achieve the integration. Therefore, interoperability is a necessary requirement to synthesise software components, business processes and application solutions through a diversified heterogeneous and autonomous procedures, which represents a challenge due to the lack standardization.

The manufacturing sector includes different models for a different design such as computation, communication, control, protocol, and network design. The requirements are dependent on application aspects. In addition to that, organizations, industries, and researchers uses different languages during the development stage. Consequently, it leads to heterogeneity in the system and heterogeneity of design and tools which generates challenges in terms of overall performance.

As previously argued, not only the heterogeneity of components but the nature of technologies and communication standards used in the factory floor -or Operational technologies (OT) differs far from the technologies commonly used in Information Technology (IT) at enterprise level. For that purpose, the international standard ISA-95 provides a consistent terminology and consistent operations models for the development of interconnected manufacturing systems. The standard model defines different layers with various functions provided by separate service and data interface requirements. It also provides detailed elements and relationships of Service-oriented architecture (SOA) business processes modelling that enables manufacturing operations within and across production facilities. However, this architecture emphasises on the vertical integration at high manufacturing levels, but less on the horizontal interconnection of systems,

Not only the cross-layer connectivity is required but the semantic interoperability between components and systems, considering all development phases of the production system. For that purpose, the three-axes Rami 4.0 architecture [1, 2] provides a holistic overview of the IT and OT integration from various domains, covering the life cycle of production systems as well. Technical objects -entities or assets- are represented starting from their development, production and use -including maintenance and repair- up until their disposal, allowing a complete virtual representation.

The Cyber-Physical System (CPS) maturity model represents the information and knowledge processing defined according to general conditions (setting basic), the generation of information (creating transparency), processing information (creating understandings), linking information (improving decision making), and self-optimising by interacting each of the components of the CPS [3, 4]. Despite this general definition of virtual knowledge-creation processes sets initial steps for more intelligent capabilities, these are further defined as smart capabilities at different cognitive levels following the five-level CPS architecture model [5]. The five-stage sequential structure aims at the final value creation from the initial data acquisition using step-by-step guidelines for the creation and development of intelligent manufacturing systems.

Even though this models break down complex processes into easy-to-grasp-packages, they are still in a high level of abstraction. Thus, application specific guidelines can improve the industrial acceptance of smart manufacturing applications.

Thus, the main contribution of this report goes towards the development of a methodology that represents the shop-floor production life cycle, digging into deeper levels of autonomy (i.e. self-configuration and self-diagnosis) and at the same time emphasizing standardization and interoperability, clue aspects of the fourth industrial revolution.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 814078

Chapter 2

Methodology

2.1 Methodology description and research questions

The lack of understanding, guidance and common practices represent a critical road-block for the implementation and application of Cyber-Physical production systems (CPPS) in industrial scenarios. Several research efforts have pushed application specific concepts and ideas in this regard. However, there is still the need of creating common frameworks, specifying flows of information and technologies for its development.

In an attempt to fill this gap and trying to smooth the transition towards smart manufacturing, we propose two research questions that will guide hereafter the discussion and analysis of the report. Those are based on a preliminary collection of state of the art works based on **Self-configuration** and **Self-diagnosis** of smart manufacturing applications. Such self-x behaviours are chosen as they represent in our opinion a vast amount of interdependent applications. Guiding research questions are:

- **RQ1:** What sequential steps are required to fulfill self-x behaviours (self-diagnosis and self-configuration) in the context of smart manufacturing, being also compliant with the concept of autonomous computing (Monitor-Analyze-Plan-Execute over a Knowledge (MAPE-K) loop)?
- **RQ2:** What technologies and industrial standards should be considered for application specific self-x behaviours (self-diagnosis and self-configuration) in the context of smart manufacturing and compliant with the concept of Reference Architectural Model Industrie 4.0 (Rami 4.0)?

The Monitor-Analyze-Plan-Execute over a Knowledge (MAPE-K) and Reference Architectural Model Industrie 4.0 (Rami 4.0) frameworks are chosen as a baseline to start this formalization process. Special emphasis is made into covering some of the gaps that state of the art works have not covered in their implementation. Fig. 2.1 represent a sketch of the methodology followed.

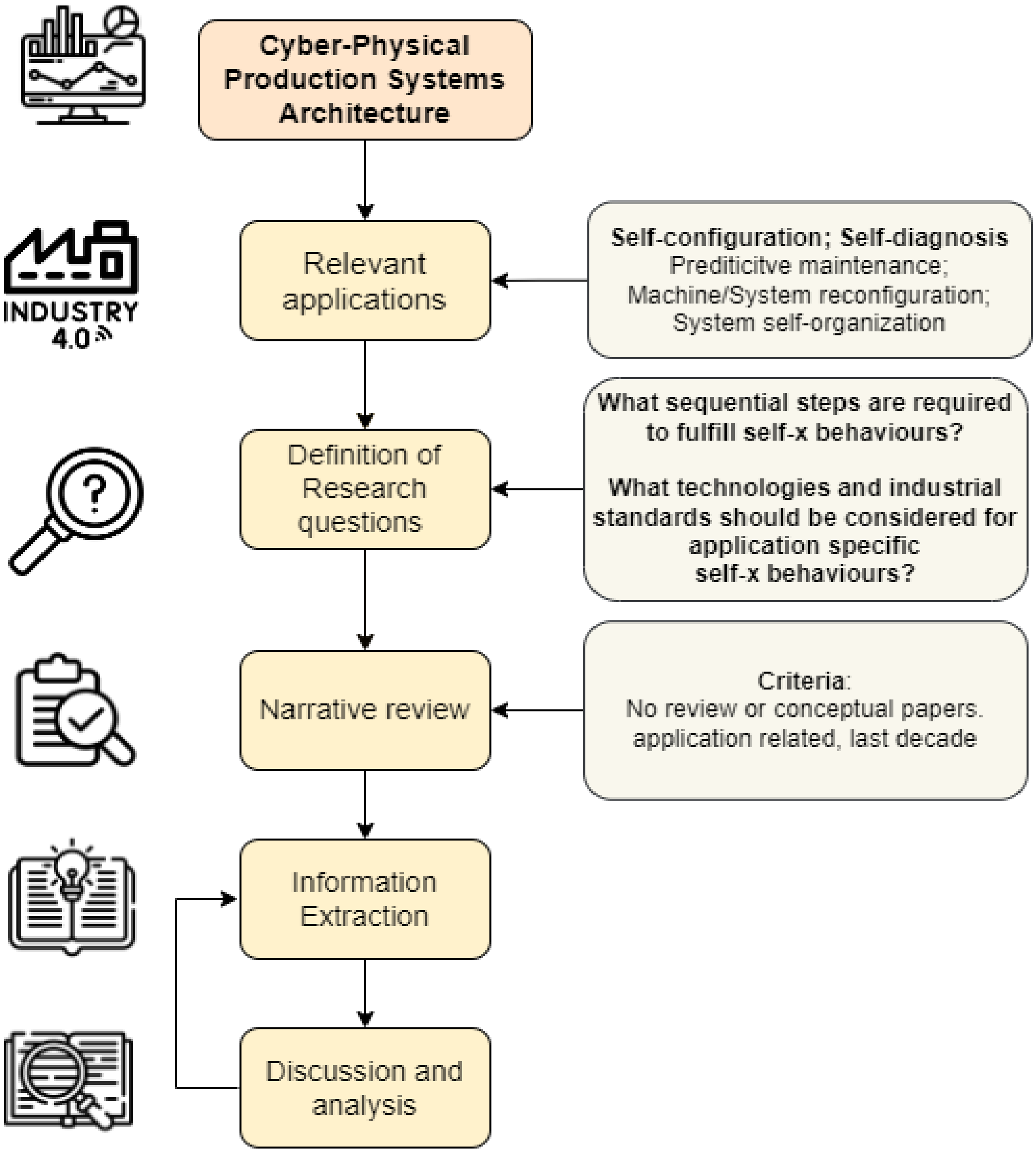

Figure 2.1: Methodology

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 814078

Chapter 3

Background and Related work

3.1 5C-CPPS

J. Lee et al. [5] proposed a 5-level architecture 3.1 as a guideline for implementing CPS based manufacturing systems. The architecture is composed of five layers, as shown in Fig. 3.1

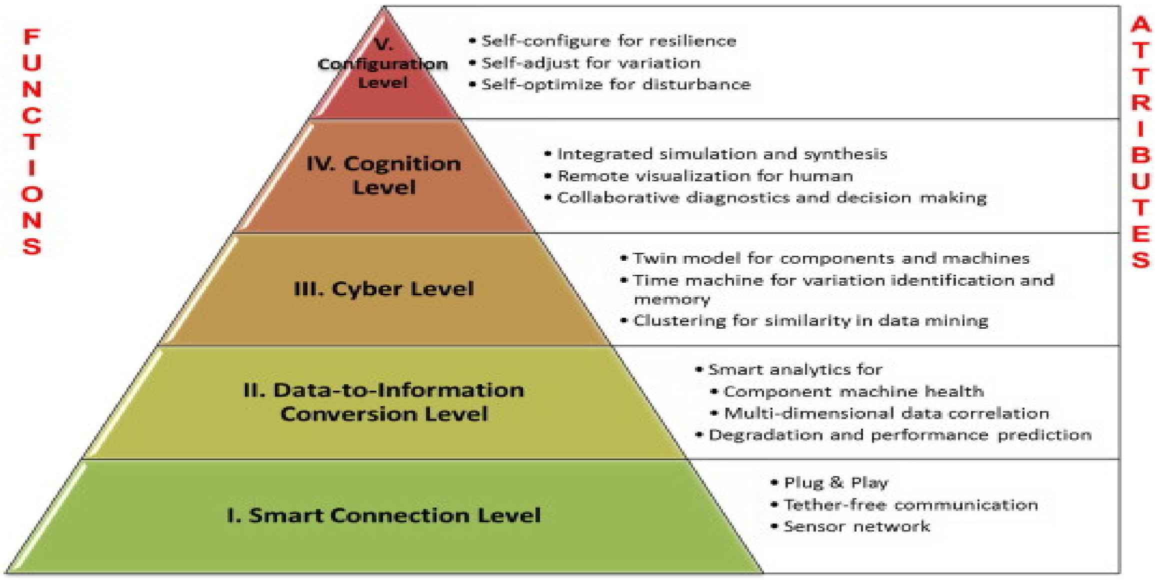

Figure 3.1: 5C-CPS architecture, from [ref]

- **Smart Connection Level**: This level proposes a tether free and seamless communication method for data acquisition. However, the data, the type of data and the sensor network must first be defined, to achieve an effective data acquisition.
- **Data-to-information Conversion Level**: This layer converts data to information to infer knowledge. It is mostly used for self-awareness such as health status management and prognosis of assets.

- **Cyber Level**: The cyber layer gathers all the data retrieved from the lower levels. Specific data analytics and self-comparison techniques can be applied to retrieve further knowledge about the performance of the assets.
- **Cognition Level**: This level transfers knowledge to the expert users for decision making and process optimisation. This requires data visualisation, integrated simulation and synthesis based on the knowledge gained previously.
- **Configuration Level**: The top level supervises the control of the assets, by configuring, adjusting and adapting them when necessary. This requires self-configuration and self-adaptation skills, so the system is able to react on the fly.

3.2 RAMI 4.0

Reference Architectural Model Industrie 4.0 (Rami 4.0) is a tri-dimensional model that describes how to face industry 4.0 in a systematic way. It ensures that all participants can communicate and understand to each other during the manufacturing life cycle[6]. See Fig. 3.2 to have a description of the tri-dimensional model. Specific benefits of Rami 4.0 are:

- It is a service oriented architecture.
- It includes IT elements during the whole manufacturing life cycle.
- Complex processes are reduced into specific layers that can include data privacy and security.

Three main dimensions of Rami 4.0 are: (1) The interoperability layer, (2) The industrie 4.0 plane and (3) The life cycle and value stream [6, 7]. In this report we are interested in the description in Rami 4.0 as a hierarchy of 6 layer (see Figure 3.3):

- **Business Layer**: Represents the business view on the information exchange related to the industrial processes. It can map regulatory and economic market structures. It can support the decisions of business executives in the decision making.
- **Function Layer**: Describes functions and services including their relation from an architectural point of view. Functions are independent from the physical assets (use case functionality).
- **Information Layer**: This layer describes the type of information to be used and exchanged between functions and services. It has information of the objects and their data models. They represent the semantic needed for functions and services to achieve communication exchange.
- **Communication Layer**: Description of protocols and mechanisms to achieve interoperability between the components in a specific use case, as well as function, and related information.
- **Integration Layer**: Provide all physical assets to create events in the form of asset administration shells. Also provides integration of network components like routers, switches, QR-codes, etc.
- **Asset Layer**: Includes systems, actors, applications as well as documents and people.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 814078

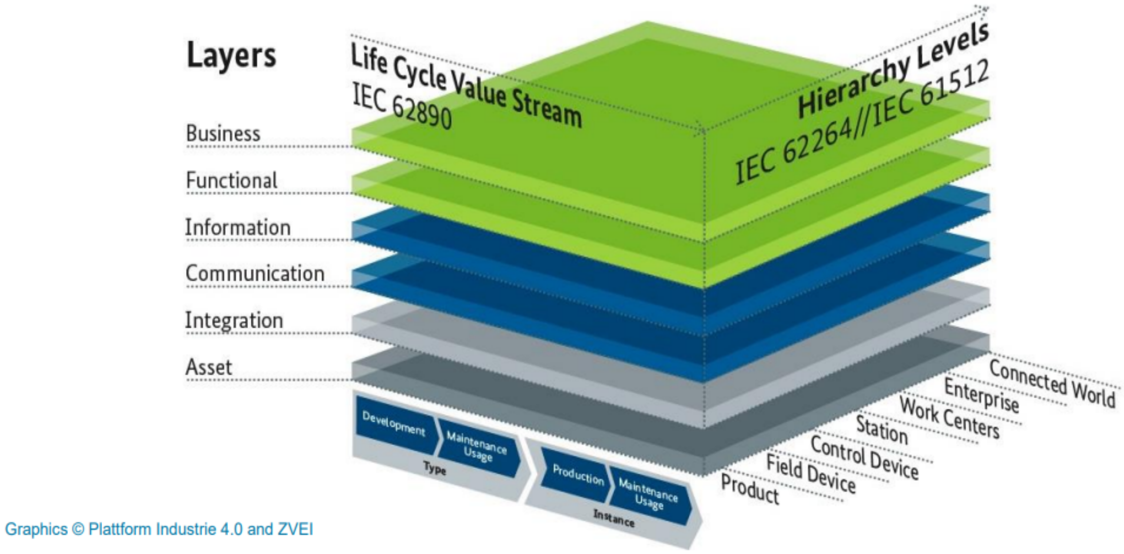

Figure 3.2: RAMI 4.0 model, from [6]

Figure 3.3: Interoperability dimension of Rami 4.0, from [6]

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 814078

3.3 MAPE-K loop in autonomous computing

The Monitor-Analyze-Plan-Execute over a Knowledge (MAPE-K) loop is designed to manage the adaption of autonomic systems [8, 9, 10]. These autonomic systems are computing systems that can manage themselves in accordance with high-level objectives from humans [11]. Fig. 3.4 shows a high-level model of an autonomic system whose a managed system is extended with a feedback loop composed of MAPE-K components.

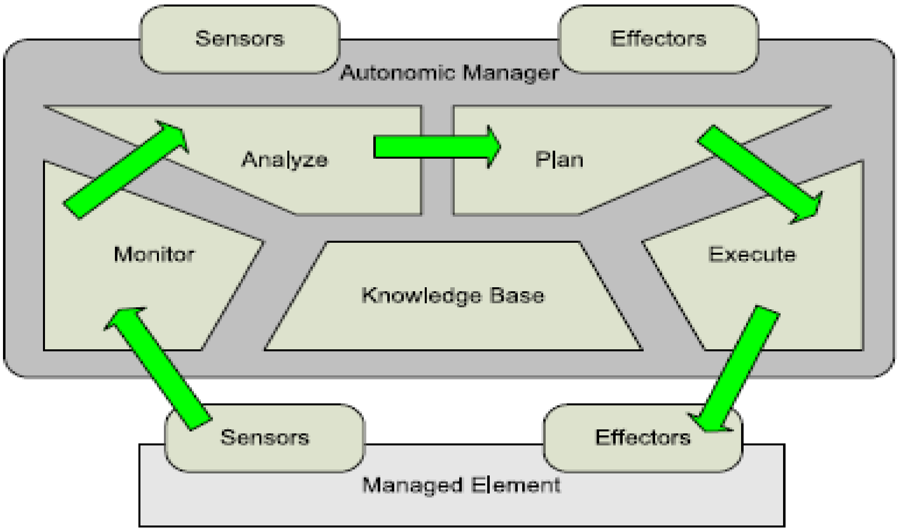

Figure 3.4: MAPE-K loop, from [12]

An autonomic system consists of a managed element and an autonomic manager with a feedback control loop at its core [12]. The autonomic manager, also called as a controller, is composed of two manageability interfaces: sensors and effectors; the MAPE-K engine consisting of a monitor, an analyzer, a planner, and an executor with a common knowledge base. The Monitor acquires information from managed resources and detects changing conditions that trigger adaptions. This is enabled by captured data with contextual information through the accumulated sensors, resulting in relevant events stored in the Knowledge base for future reference. The Analyze component senses the Knowledge base to define the requirements for adaptation of the managed autonomic system with respect to the system goals. If adaption requirements are triggered, the Plan component makes up a plan consisting of adaptation actions in the managed process through its effectors. These adaption actions are executed by the Execute component adapting the managed autonomic system as needed [8, 12].

The autonomic system is formed by the control loop and can be connected with other autonomic systems (e.g., self-configuration, self-diagnosis). This means that multiple MAPE-K loops can be deployed in the managed systems and the the MAPE-K loops' components can be decentralized to maintain a large, complex structure [10].

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 814078

3.4 Smart manufacturing applications

Smart manufacturing covers a broad scope of applications in the field of production engineering that tries to integrate new technological enablers i.e. cloud computing, collaborative robotics, internet of things [13] with current production requirements i.e. high individualization of products or short time to market. These new requirements entail new business models i.e. engineer to order and compel companies and shop floors to have higher levels of autonomy (autonomous manufacturing automation). In this context self-x behaviours [14] are driving such research endeavour and can be considered a starting point for novel research and development in smart manufacturing applications.

3.4.1 Self-configuration

Definitions: There is a large body of existing work in manufacturing systems that can be quickly adapted to changing requirements, known as flexible, agile or re-configurable manufacturing systems. In section 3.1, the 5-level architecture of CPS was presented, the top-level represents the configuration aspect that deals with developing systems where each machine has the capability to decide its own parameters and operations to achieve a *self-configuration* [15]. However, the definition and scope of *self-configuration* is still not well established.

Self-configuration (sometimes referred as reconfiguration) and adaptive coordination (sometimes referred to as adaptation) refers to the spectrum of changes that a system makes to itself in response to occurrences in its environment and internally [16]. For [15], self-configuration is is the ability of system to change its configuration (i.e., the connection between different system modules, parameters, and calibration) in order to improve or restore system functionality in response to actions". In [17], the author highlights the *connection* aspect, by stating that "Self-configuration refers to the capability of automatically identifying manufacturing things and connecting them into a feasible manufacturing line." By contrast, [18] refers to the capabilities of manufacturing nodes by stating that "self-configuration is defined as the preparation required to realize the core technical functionalities of a smart manufacturing technology. This involves the generation of inputs required by the corresponding engineering application without user intervention." Likewise [19], stated that manufacturing systems should have a set of available skill capabilities used to configure an assembly system by comparing a set of process/skill requirements to the available skills. Hence, this match process would support the real-time resource (skill) allocation. This makes it evident that a process model will need to include a skill and requirement concepts.

Another significant aspect is the association in the literature of 'self-configuration' with 'plug-and-produce' concept, which purpose is to quickly enable (dis)connect components of a manufacturing system with little or no reprogramming and reconfiguration of the remaining system. Since the domain of manufacturing equipment is large, complex, and lacks interface standardization, modular hardware architecture is often introduced to encapsulate components as modules with well-defined interfaces [20]. Similarly, [17] also emphasizes the importance of standard hardware and software interfaces, so manufacturing networks can be dynamically re-established via adding or deleting a manufacturing thing (or node) or re-configuring the links (or edge) between connected manufacturing nodes. Generally, self-configuration can be observed within smart factories that have been set up in accordance with standards such as ISA-95, Automation Markup Language (AML), and Open Platform Communications United Architecture (OPC UA) [18].

Although differences of opinion still exist, there appears to be some agreement that 'self-configuration'

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 814078

refers to the identification of the availability of resources and connectivity of several modules according to their skills and capabilities to fulfil a set of requirements. As for the benefits, the ability of self-configuration will allow increasing the use of CPS and in the end, represents an improvement in productivity and cost-reduction [21].

Relevant state of the art:

A large amount of research has been carried out in the field of self-configuration of manufacturing systems. Products with different requirements need different robot and machine configurations; thus, it is a key trend to generate adaptable production systems. [15] Presents a framework for manufacturing self-configuration. It is focused on machine parameter reconfiguration for testing equipment. Once a new part is introduced for testing, various agents interact: querying, executing, analyzing and comparing machine settings from a cloud base pipeline. In [22] authors present the concept of skill self-configuration in the context of Evolvable assembly systems. The process starts with specifications of the workflow of the product, the dynamic assignment of skills to an agent requirement, finally the process configuration is carried out by the deployment of composite skills. Sanderson et al [21] present a function behaviour structure methodology for adaptable production systems. It is complemented by an ontological model to support the adaptability and configuration of the production system. Ontologies for self-configuration have been also used with multi-agent modelling to support semantics and cloud based decision for supervisory control as shown in [23]. In [24], the self-configuration of a plug and produce system is proposed based on a service oriented workflow manager.

For adaptive manufacturing control, three "self-x" functions perform collectively to achieve optimal manufacturing operations and system performance. Being self-configuration one of the capabilities. In [17] and [25] a reconfiguration management module is introduced as an enhancement for CPPS to address the need to identify the demand and plan for reconfiguration. The demand for reconfiguration is triggered for changing requirements that affect the target production, alternative configurations are generated at machine level and system level as well as the optimization of production parameters. Finally, the new configurations are evaluated to select a new configuration. The implemented reconfiguration management relies on a two-stage genetic algorithm for the generation of configurations. [26] designed an analytical target cascade method for configuring manufacturing resources. These algorithms can set up manufacturing items in a suitable manufacturing route based on pre-established rules. However, they cannot self-learn and deal with uncertain manufacturing environments. Using a multi-agent based approach, [27] proposed a plug and produce architecture for configuration and reconfiguration of assembly systems, the methodology is based on a fitness function to select the fittest capability among a set of capabilities that can accomplish a given task, so in that way allocate tasks to resources. On the other hand, [18] proposed a digital twin autonomous framework for self-configuration, using Asset Administration Shell (AAS) as a digital representation of manufacturing assets and capabilities. The framework has a component manager capable of generate a service composition among heteregoneous components using SOA. Product and plant requirements are represented using the standard AML.

[28] approaches the concept of Plug & Produce (PnP) by dividing the runtime process in three phases: i) the first (Plug) phase includes the connection of components; ii) the second phase (Plan) describes an iterative strategy for integration, localization of components and computation of robot motions and collision-free paths; and iii) the final phase (produce) executes automatically the program without human presence. The interaction of human operators is mainly studied in the second phase. This approach uses the ability of the system to manage the representation (self-description) of the components (model of the structure, functional role and geometrical information) as input for the planification phase. [29] presents i) a unified abstraction to describe the skills and requirements,

ii) generation of resource-specific action in formal execution considering the high-level descriptions, and iii) the runtime orchestration by exposing the actions as services. The Smart Manufacturing & reconfigurable Technologies (SMART) demonstrator [30] is a complete integrated environment. The agent control layer allows the workstation to be dynamically added, removed and controlled at runtime with minimal human intervention, based on a DDS communication specification where agents subscribe and publish to inform on relevant topics. The Instantly Deployable Evolvable Assembly System (IDEAS) [19] aims at proving that agent production devices with fine granularity enables a highly responsive and adaptable reconfigurable production system. The first self-reconfiguring system is based on mechatronic agent (MA) with standardised interfaces and build-in control capabilities which allow quickly connections and dynamic configuration. The idea extends from physical equipment modularity to functional capabilities needed to execute assembly processes. [31] stresses the importance of the representation that an automation agent has about itself in the same symbolic way as its environment; i.e. a symbolic representation containing the relationships of this agent with other entities as well as with its environment. In this sense, its own word model contains a representation of itself (model reflective). This is different from a pure software (SW)-centric approach since this approach includes a hardware (HW) "embodiment" in the form of a mechatronic component, similarly to the one presented in the IDEAS project.

Current work contribution: Previous literature suggests that self-configuration int the context of adaptable manufacturing systems is still an open research gap, specially in the context of CPPS, where the plethora of emerging technologies can support new methodologies for self-configuration. In particular, the generalization of this adaptive behaviour to various contexts seems to be still a major issue. Other factors and implications for future research to consider are:

- Automatic software configuration, and management of internal configurations of machines.
- Lack of generalization of approaches for a whole family of products.
- Knowledge based approaches to support the self-configuration process based on experience.
- The standardised representation of this knowledge.
- Further analysis of the human implications in a self-configuration process.
- The development of different assessment methodologies for the evaluation and characterization of reconfigurable systems. These tools might help monitoring performances to face unpredictable behaviours and to point into future research lines.

3.4.2 Self-diagnosis

Definitions: In general terms, Self-diagnosis is the characteristics of diagnosing, or identifying, conditions in oneself [32] . In manufacturing oriented, the Self-diagnosis characteristics of a system means the ability of a system to identify and solve the error by using data of manufacturing systems to give useful feedback, allowing smart decisions for efficient productivity, and maintainability. In conclusion and simple words self-diagnosis characteristics means the ability of a manufacturing system to detect and analyse an error or malfunction within itself. General objectives of such an approach are identifications of the anomalies, breakdowns, and fault causes in the systems [33]. Benefits of Self-diagnosis characteristics in manufacturing are as,

- Reducing accidental maintenance,
- Managing the difficulty in the manufacturing's.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 814078

- Efficient process and product quality
- Avoiding the failures in the operations and High production capacity
- Providing decision support [34]

Relevant state of the art: In the last decade, many researchers worked on diagnosis field. They proposed different methods like reasoning model, fuzzy logic, and neural networks. For example, [35] used symptom-based and functional-based reasoning modeled by agents. [36] focused on deep learning algorithms for self-diagnosing the total risk of autonomous vehicles. [37] combined statistical methods with domain-knowledge to extract features from sensor data, and applied supervised machine learning algorithms for predictive maintenance. Supervised learning however relies on classification of data based on previous knowledge. Thus, [38] proposed a semi-supervised deep learning approach to train the model for fault diagnosis.

Conventional diagnostic methods have aimed to model the subsystems with little integration. J.Lee et al. designed watchdog agent, that accumulate information's from sensor to identify the process failures [39]. Similarly, R. Yu et al. proposed POMAES (problem oriented multi agent based e service system) for the collaborative maintenance [40]. Furthermore, J.Göhringer designed the internet based diagnosis in the assembly systems which is also capable of remote monitoring [41]. To solve the issues in identify errors and defects in manufacturing systems different approach found which is listed below. In the another study [42], reported the combined combining HMM (Hidden Markov model) and Principal Component Analysis (PCA) for defect diagnosis. Principal Component Analysis was applied to for feature extraction, whereas Hidden Markov model was used to differentiate different process operating parameters. In [43], proposed multiple agent diagnostic method to detect and remove errors in manufacturing systems. This method designed to endeavour the imitate human behaviour using multiple fuzzy intelligent agents that check the issues from different views. This multiple agent diagnostic system was installed in CIM (computer integrated manufacturing) which includes of two conveyor which are used to transfer pallets, robots, and stations to simulate system. The work from MAGIC EU project [44], in which author proposed the diagnostics tools that used signal and casual based diagnostic agents. The signal-based diagnostic agent to execute classical signal based identification, whereas causal based diagnostic agent to emulate the man behaviour. Similarly, another study reported the multi agent system to achieve in diagnostic features in the power field with the help of PEDA (Protection Engineering Diagnostic Agents) system [45]. European project I-RAMP3 (Intelligent Network Devices for fast Ramp up) worked in the direction to change manufacturing system into an agent similar system with the help of NETwork Enabled-Device (NETDEV) and sensor information to achieve self-diagnosable characteristics, monitoring, and control in the systems [46]. However, there is still some major challenges to achieve self-diagnostics characteristics, which includes capturing the domain knowledge, control error propagation, efficient models for data processing, planning and execution task [47].

Recent studies also focuses on self-diagnotic capabilities that are integrated to intelligent agents, which is enabled by data-driven techniques. Specifically, [48] portrayed intelligent systems (e.g., software agents or mobile robots) that interact with their environment and may suffer from undesired environmental changes, faulty perception, or internal faults. These intelligent systems are required to be embedded with the self-diagnotic capabilities to help them detect different situations and react in a smart way. The authors made the use of a declarative programming paradigm (ASP) based on the logic programs and their answer sets to build a strong-fault models integrated to the intelligent systems. To build the environment for self-diagnosis, [49] presented architectural instance for industrial-analytics Big Data for the use case of fault diagnosis. This architecture begins with the manufacturing resources data captured from heterogeneous source stream. The data go through the

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 814078

process of data characteristics extraction, which is processed by procumpute views: OLAP model and data mining techniques, and learning model with Naive Bayes algorithm. Instead of proposing the architecture development of self-diagnosis, [50] presented an overview of mature fault identification and diagnosis techniques in a wide range of literature review. The authors found that the wide use of Model-Based and alternative Data-Driven AI techniques across the aerospace, automotive and industrial control domains reflects the complexity of the current industrial CPS application space. In line with [50], [51] proposed a framework for the fault diagnosis knowledge-base based on semi-supervised multi-spatial manifold clustering method to achieve the automatic evolution of the knowledge-base for the diversity of faults. To enhance the reliability and accuracy of self-diagnosis in manufacturing systems, [52] addressed the problem related to the lack of accuracy or fluctuation of data, affecting the correct classification rate of faults. The authors presented a new hybrid fault diagnosis approach that combines learning algorithm and knowledge base (Fuzzy rules) to handle ambiguous and even erroneous information.

Self-diagnosis mainly relies on sensor and manufacturing data to detect or predict malfunctions. Hence, [53] focused on self-aware sensor networks capable of self-diagnosing and self-configuring in case of breakdown. They proposed an agent-like system for collaborative communication and self-aware capabilities based on sensor nodes. Similarly, [50] showed a use case that enables the self-diagnosis with the system of sensors and actuators where the sensors detect the environment (e.g, walls or other nearby obstacles) and actuators (e.g., movement arm controlled by a motor) carry out actions based on the commands of diagnostics system. Therefore, they must be integrated in a self-diagnosis system to enable the intelligent agents react in a smart way.

Current work contribution: The previous studies indicates that self fault identification in the manufacturing systems is still in infancy. In specific, still manufacturing system are not able to detect and correct the fault within itself. In this study, in order to achieve self-diagnosis characteristics the necessary sequential steps and technology are mapped according to MAPE-K and Rami 4.0 architecture layers respectively.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 814078

Chapter 4

Results and Discussion

This section aims to provide a detailed description of both: functions and services require for self-configuration and self-diagnosis, considering the MAPE-K and Rami 4.0 frameworks. The generalization of those behaviour are based on a review of relevant state of the art in the context of smart manufacturing related applications already presented in Chapter 3 of this report.

4.1 Formalization of the self-x behaviours (MAPE-K framework)

4.1.1 Self-configuration

In the context of the current report, manufacturing self-configuration is based mainly in the context of intelligent product driven manufacturing, i.e. consideration that the product has the knowledge of which operations need to be done [54]. From this point we divide such self-configuration process considering the 4 stages of the MAPE-K loop, as follows:

- *Knowledge:* This stage may include manufacturing storing information relevant for the managed system, i.e. its environment, own description and status.

  General equipment description, equipment suppliers' data, and system Integrator's data are useful for defining high level properties of assets [22]. Asset's context information includes physical event context (internal status) and social interaction context (environment status) [55], which set a situation model [31]. At lower operational level, information about components' functions or activities they can perform (skills), and failures that could affect their behaviours are also stored [19, 31]. These behaviours or skills will form the basis for the monitoring and evaluation of adaptation functions or mechanisms [21]. This information of assets is often stored in a centralised knowledge base by semantic descriptions [29], but may limit system's requirements as scalability or modularity.

  Manufacturing activities, process and operation information, as well as recipes [21] conforms manufacturing planning at operational level, setting the activity models [18, 28].

  A set of performance indicators, as well as an historical of fault models, are also stored in the knowledge stage in order to evaluate the performance of the system that might trigger the autonomous adaptation mechanism. These indicator may also be useful in helping during the selection between different adaptation strategies.

- *Monitoring:* The *monitoring* stage is in charge of collecting data from all managed resources and storing it into *knowledge* data repositories (when required). In this context, manufacturing resources are all the elements required in the manufacturing process, whether they are humans, products, materials, software or machines. Raw data from devices or machines; requirements in the form of production orders delivered by users and customers are captured and sent to the components on the *monitoring* stage.

  These components will be modelled with descriptive attributes to be autonomous entities, but also interact with other components. The first component are *machining resources* (i.e. motors, grippers, robots) that include the run-time conditions of machines and skills [23, 19, 22]; The *production* component monitors new production requirements collected from customer and users orders into the manufacturing systems. Lastly, *transport entities* monitor the requirements for transportation from the current location to destination units (i.e. conveyors, Automated Guided Vehicle (AGV) etc.).

- *Analyzing:* In the *analyzing* stage, information collected from the previous phase is analyzed to detect if changes or adaptations are needed [10]. The information collected from *production component*, allows analyzing if production requirements have changed and require a new adaptation. Then, by analyzing the data from *machining components* it is possible to analyze the feasibility of adaptation. For this, the production requirements are analyzed and validated against machining skills and *recipe files* of production processes kept into *knowledge* repositories [21]. Furthermore, the information of *machining components* allows analyzing performance requirements to achieve and maintain system quality. These kinds of requirements can be analyzed in the form of KPIs related to a specific run-time condition or machine skill [21, 29].

- *Planning:* The planing phase in the MAPE-K model describes the actions needed to self-configure a system. Planning activities to recover the manufacturing system from an undesired state, identifying the necessary steps, creation of the plan and a launching a signal that triggers the execution of the plan (in this case for self-configuration).

  The planing phase stars by localizing physically and logically the modules necessary to perform a manufacturing operation e.g location, availability, network infrastructure.

  Depending on its availability and physical localization, these components can be grouped considering the required action e.g. gripper + axis can form a robotic arm. Also, a Robot on top of an AGV can form a mobile resource for intralogistics operations. The grouping is necessary to understand which manufacturing modules will be used and how they will the interact with each other.

  Considering additional production requirements, different task can be prioritized. This is useful to optimize the time of operations or optimize energy consumption. At the same time and considering additional production requirements, different task can be prioritize. In case of more than one resource/set of resources available, a balancing work load distribution is also possible. From this criteria, a workflow of operations can be generated (i.e. sequence). Those usually describe sequence of steps to manufacture a product.

  The last part of the planning phase should consider the generation of skills/services to be carried out by each resource. Some resources may have functionalities which result of a composition of various skills/services e.g. a pick and place operation.

- *Executing:* This phase executes the self-configuration actions of the plans that have been generated, adding resources or releasing necessary behaviours.

The self-configuration actions are dependent on the level of flexibility or automation that a manufacturing process have. Thus, the execution can be totally self-managed or may need manual assistance. For example, in plug and produce systems even if the software can be manually updated, pluggable modules should be still manually placed. Certain shop floors with high level of flexibility can use AGVs or cranes for material/tool handing and routing.

The execution stage under the scope of self-configuration in manufacturing is in charge of executing agent behaviours in manufacturing resources or providing the necessary services required in commercial boards and low level control devices.

Fig. 4.1 presents a first sketch of a conceptual framework of the self-configuring behaviour using the MAPE-K model as well as main components and its interaction.

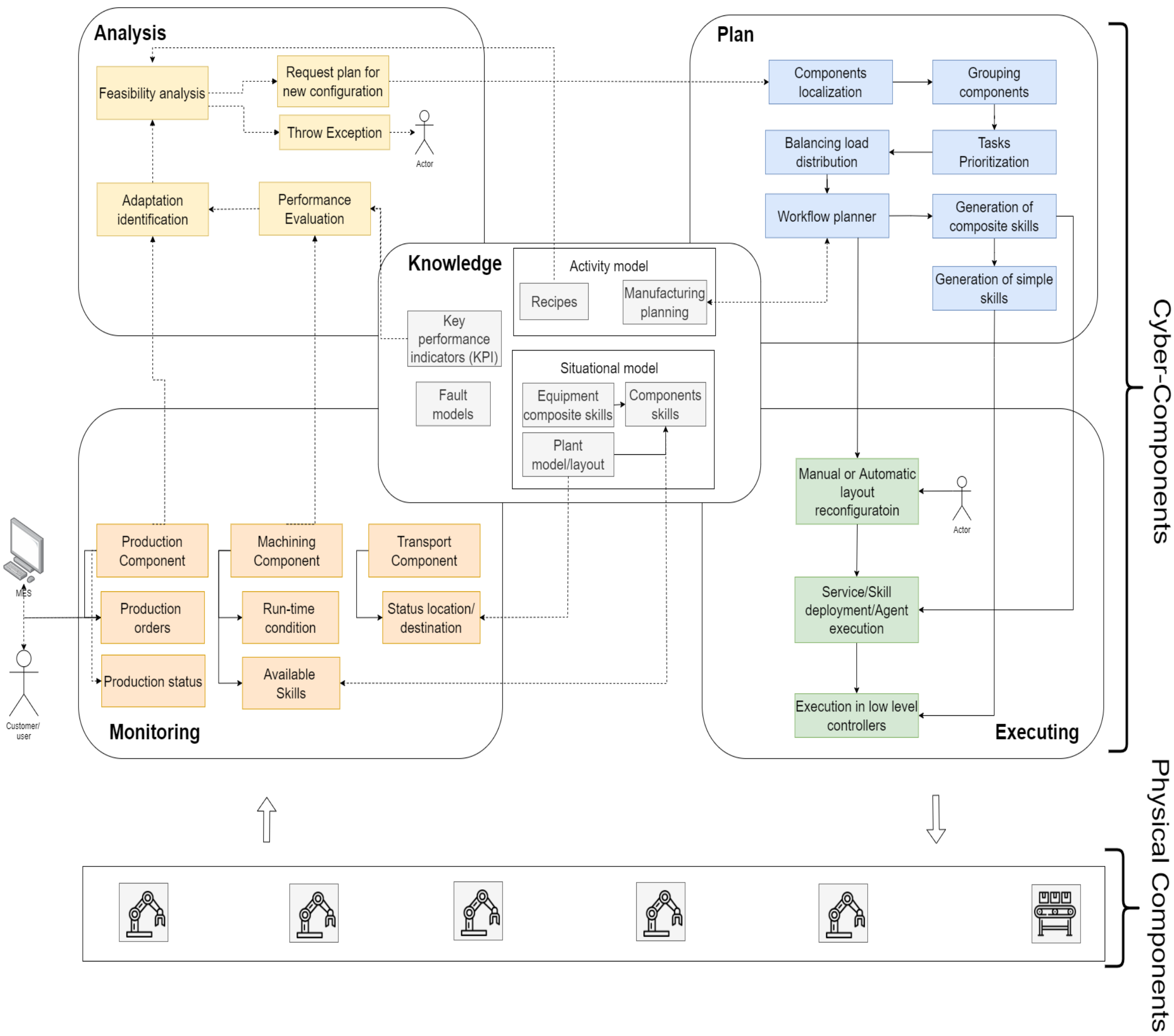

Figure 4.1: Conceptual framework for self-configuring behaviour using the MAPE-K model

4.1.2 Self-diagnosis

In this section, the sequential steps are mentioned to achieve Self-diagnosis characteristics in manufacturing line on the basis MAPE-K loop, which are divided in knowledge, monitor, analyze, plan, and execution step.

- *Knowledge:* The fundamental functions of self-diagnosis is detect and monitor abnormality during all operational stages of an agent [56, 53]. This is realized by the collected operational data [57, 50, 48]: environment characteristics, episodic experiences, risk information, policies for other components in the MAPE-K loop, models for fault detection and correction, data input variables and fault diagnosis. The data enable the agent to make the autonomous adjustment of policies and symptoms to react the environment and operational changes [58, 56]. The models for these autonomous adjustments are also diverse, such as combined learning algorithm and knowledge base (neuro-fuzzy approach and probabilistic model) [52].

- *Monitoring:* The responsibility of the monitor stage is to collect online/inline raw data from sensors. The raw data includes, data collected from monitoring equipment (machines and devices from manufacturing system), which describes the input from the environment (camera and wheel encoder) [49], attribute, knowledge element, state change, operating procedure, and feedback [51]. After the data collection, the data pre-processing involves a data filtration and data normalisation. The assumptions used in data pre-processing are retrieved based on the knowledge sections historical data. After the pre-processing of data, necessary sensor data are transferred to the analyse stage [59].

  In summary, the monitor section includes the substeps: collection of data, data pre-processing (filtration and data normalisation) and updates knowledge section, and transfer important data to analysis section for fault identification.

- *Analyzing:* The pre-processed data received from the monitor section are analysed in the analysis section in order to identification of the faults and anomaly [53]. Based on depending types of data and topology of the sensor networks, different types of fault classification algorithms and models can be used in this phase. At the identification stage, the sensor current data symptoms can be compared with value forecasted by fault model [60]. The fault model is stored in the knowledge repositories and based on the historical data. In this way, fault can be isolated by examining the data and then fault can be classified with the help of the knowledge base [60, 52, 61].

  After identification of faults, the analyse section transfers the fault data to the knowledge components and fault correction requests to the plan section for preparing the recovery plan.

- *Planning:* The planning phase is responsible of making some adaptation plans based on previously identified failure data. There is some overlap with Analysis phase as planning might also involve some further fault assessment to locate the root cause of the failure, or do some diagnosis analysis. For example, [36] diagnoses the risk of the vehicle condition to foresee the risk implied when driving autonomous vehicles.

  As outcome of the diagnosis and fault analysis, some policies are set to make necessary adjustments on the plan [56]. These policies are also stored on the Knowledge databases for continuous feedback.

  The planning process then makes process or system recovery plans according to the policies, knowledge databases and diagnosis results [56, 48, 57]. Other contingency actions might be less disruptive and propose a set of adoption sequences [58, 49] for later execution.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 814078

- *Executing:* This step executes the previously generated contingency and system recovery plans. Some of the contingency actions directly have impact on the software itself, which might imply the termination of an existing task [53]. In other cases, fault rectification and correction is applied directly on the actuators in the field [49, 50].

  Execution sometimes might require some human intervention. Hence, execution results and failures are reported to the operator for better decision making or monitoring purposes [51]. Fault execution results are also reported back to increase the knowledge base [37, 59].

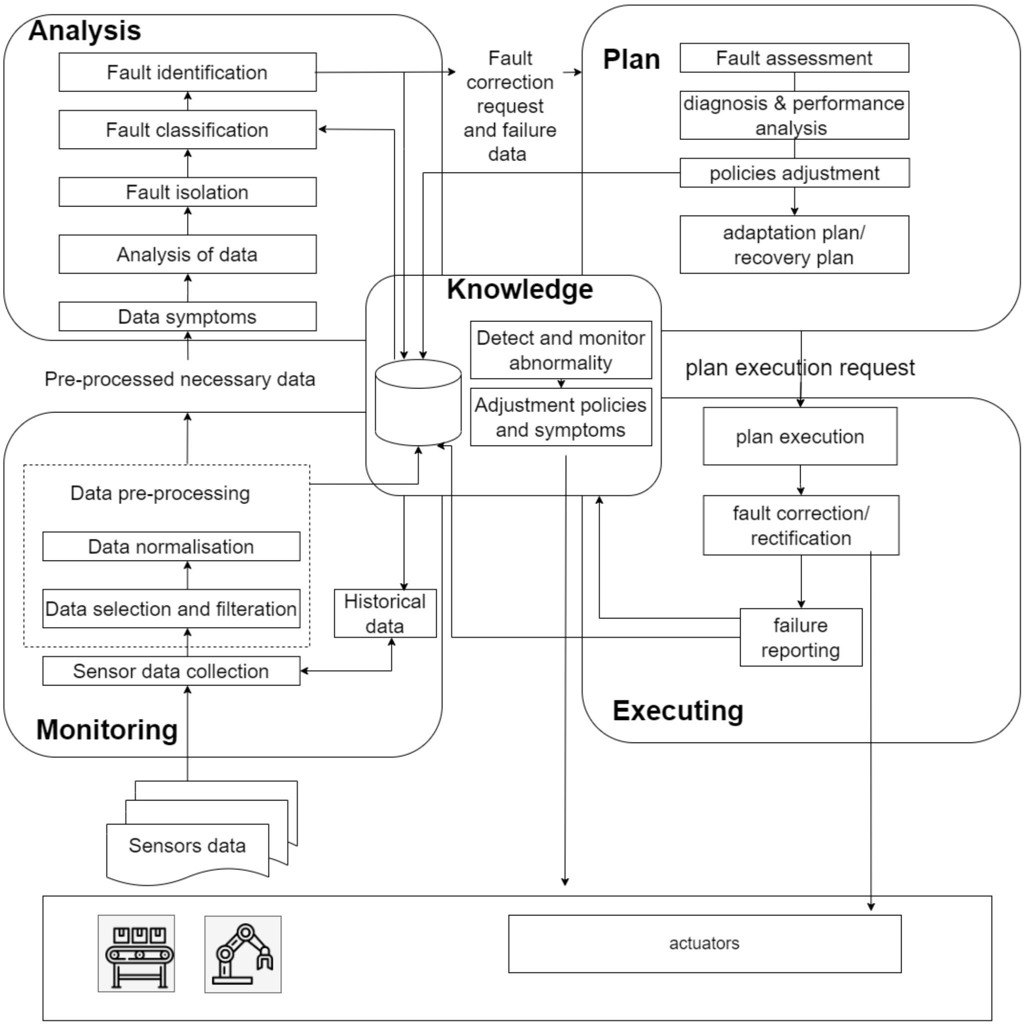

Figure 4.2: Conceptual framework for self-diagnosing behaviour using the MAPE-K model

Fig. 4.2 presents a first sketch of a conceptual framework of the self-diagnosis behaviour using the MAPE-K model as well as main components and its interactions.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 814078

4.2 Technologies and standards required/ Mapping with RAMI 4.0

4.2.1 Self-configuration

A smart self-configuring system represented in the layers of the Rami 4.0 framework consists of building a generic set of concepts for the interconnection or most of the function of a self-configuring system. From the high level (business layer to the shop-floor). Below a generic description of the building blocks are described.

- *Asset layer layer:* In the asset layer, we can find the physical components that constitute a self-configuring system i.e.: milling machines, tools, robotic systems, controllers, transport systems, pallets, people. Description of the asset layer constitutes also main elements of the monitoring stage of the MAPE-K loop, as shown in Fig. 4.1 i.e. production components, machining components and transport components.

- *Integration layer:* By means of the integration layer, physical assets are connected to the digital world. This layer represents a middle-ware that eases the readability and standard at both sides (physical and digital). It has elements such as RFID, NFC, QR codes, wired connection and gateways. The representation of physical resources as multi-agent systems i.e. product, resources, transport or the encapsulation of the functionalities of the assets as services. Various types of sensors such as energy meters, temperature, vibration and others can also be used as integration of physical devices and physical variables.

- *Communication layer:* The communication methods through the layers are crucial to obtain data from assets and deliver functional services necessary to fulfill business requirements. A Service-oriented architecture (SOA) allows for system components to be reusable and interoperable via system interfaces. In Fig. 4.4, the communication is presented in a bidirectional view from business layer to asset and vice versa.

  Consumers communicate to the system through the use of multi-platform applications such as web or mobile apps. These type of applications use the secured Hypertext Transfer Protocol (HTTP) based communication protocol.

  Data communication from assets to service oriented layer is supported by data transmission protocols like OPC UA , MQTTT, MT Connect, etc [62].

- *Information layer:* In the information layer, the processes associated with data pre-processing, organization and structuring through information models will be defined. The correct definition of these processes will allow the generation of functional services in the subsequent layers. Therefore some basic processes have been included such as: data pre-processing, data normalization, data filtering and cleaning. However, in the context of self-configuration, it is important to define the data models for the elements of the knowledge component defined in the MAPE-K of subsection 4.1.1 These are runtime conditions, KPIs and skills data models from machines; and product recipes data model. Here, it is important to consider some standards that can be used to improve the interoperability among devices. For instance OPC UA provides both communication protocol and information modelling method. The OPC UA CNC Model is a domain-specific model that allows to represent the data from the CNC itself and from peripheral connected devices. The MTConnect Devices Information Model provides a hierarchical representation of equipment metadata that includes the logical and physical

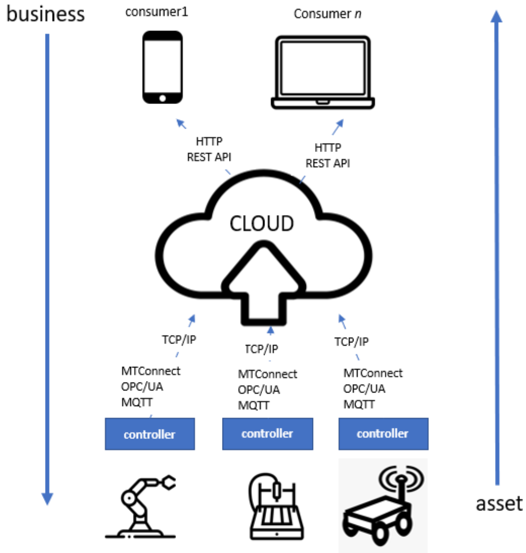

Figure 4.3: Communication Layer perspective

components and sub-components. It also allows to specify the definition of data entities that may be reported by one or more pieces of equipment.

- *Functional layer:* The digitalisation of products allows the creation of smart products with more technological features. Digitalised production machines allow quick configuration and reconfiguration of tools (PnP) enhancing production functionalities and real time monitoring. Both assets (products and production machines) are able to exchange information and data in order to create a more efficient manufacturing ecosystem, supported by functionalities in decision-making. This enables a production shaped around innovative enterprise strategies and services, focusing on viable market demands.

- *Business layer:* Includes meeting market requirements of mass customisation, One-of-a-Kind Production (OKP) or Batch Size of One (BSo1), but with sustainability as a driver (market and sustainability pulls) for a more connected while decentralised production. The embrace of this manufacturing objectives follows new business creation processes (sharing and/or rental Business Models (BM) and the embrace of the PTO paradigm, establishing innovative enterprise values and objectives which, in turn, are generally created at work centre level.

  A summary and representation of the elements of a self-configuring framework considering the RAMI4.0 approach is shown in following Fig. 4.4

4.2.2 Self-diagnosis

This section describes a self-diagnosis system represented in the frame of Rami 4.0 architecture layers. Each layers are described as to enables the development of self-diagnosis concepts in manufacturing so that various components and processes in manufacturing can be interconnected including the physical devices and cyber space tools.

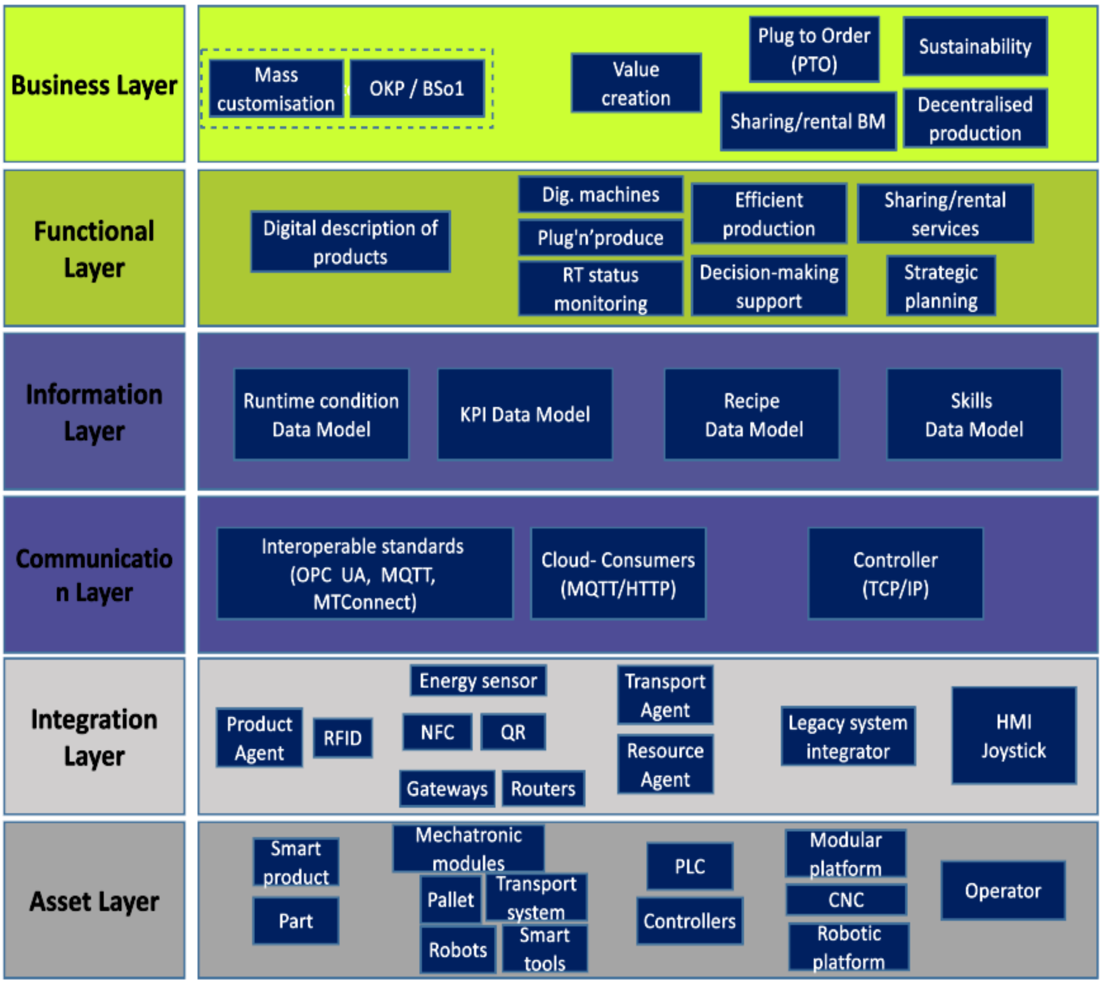

Figure 4.4: Rami 4.0 architecture layers for self-condiguration

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 814078

- *Asset layer:* Asset layer identifies and describes the physical distribution of all participating real assets from shop floor. It includes physical machines, devices, sensors, actuators, robots, conveyors, controllers, stations, documents, and human beings (operators) Fig. 4.5. This layer recognise the products and parts in relation to monitoring requirements for machines [63] and gives the initial idea to establish the integration in between different physical assets for controlling of the faults.

- *Integration layer:* The integration layer is responsible to provide interface in between the physical things and cyber world in order to perform the transition from physical world to the cyber world. This layer consists of tools associated with IT, including human machine interface Human Machine Interface (HMI), identification of materials and product with the help of radio frequency identification Radio Frequency Identification (RFID), near field communication Near Field Communication (NFC), QR, and barcode readers, and manage the faults from different sensors, and actuators [64]. The different types of sensors such as position, torque, gyro, voltage, current, temperature, vibration, charge-coupled device charge-coupled device (CCD), complementary metal oxide semiconductor complementary metal oxide semiconductor (CMOS), and force sensors can be used according to occurance of faults in the manufacturing. The integration layer allows an interoperable information exchange via communication layer.

- *Communication layer:* The communication layer is vital to establish the communication among the operational assets, and the top layer (i.e. functional layer, business layer) functions and services. This includes communication protocols to make the system interoperable in both directions. Profibus [57], Ethercat, Controller Area Network (CAN), FlexRay, Media Oriented Systems Transport (MOST), etc. [36] are some of the transport protocols used in the field level. Additional resources are frequently connected to the Industrial Internet of Things (IIoT) through communication technologies such as Bluetooth, WiFi, Ethernet, 5G, etc. [57].

  In regard to Internet of things (IoT) bridge, Nguyen et al. [57] developed a self healing framework based on the ClouT platform [65], which connects the Internet of Things with Internet of People via Internet of Service. Besides, OPC UA, Automation Markup Language (AutomationML) and MT Connect are some of the prominent communication standards for data exchange and interoperability across horizontal and vertical layers, as defined in [66].

- *Information layer:* The information layer describes the semantics and standardised data modelling languages for data exchange. Formal modeling languages such as Architecture Analysis and Design Language (AADL) [67, 68, 69], Unified Modeling Language (UML) [70, 71, 72] and Modeling and Analysis of Real-Time and Embedded systems (MARTE) (based on UML) [73] have been widely used for fault diagnosis during the design phases [74]. UML state machines have not solely used during the design phase but also at runtime. M. Illarramendi et al. [75] presented an adaptable UML approach to automatically adapt the behaviour in case of errors or unexpected situations.

- *Functional layer:* In the functional layer, this allows to access the the gathered data in the relational databases stored in the information layer. Subsequently, this functional layer is equipped with the analytical capabilities performing advanced data processing. Specifically, it is in charge of processing the data collected by the machines and/or intelligent agents, sensors and environment and operational changes. Subsequently, the functional layer aggregate the processing result of the data to build knowledge on machine and business processes and also behavioral models for each machine and business process.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 814078

This aggregated knowledge and built behavioral models allow for the detection and monitoring abnormality (e.g., failure detection and modes) and then develop accordingly adjustment policies and symptoms for manufacturing self-diagnosis. This function layer is implemented on the top of Big Data solutions that execute diagnosis and abnormality or failure detection algorithms in a computing cloud. These algorithms are diverse and context dependent, such as: time series analysis, regression analysis.

- *Business layer:* In the business layer, this layer details high-level information offered by the functional layer and differentiate them to involved stakeholders, e.g. operational details of machines to engineers, business supporting services to machine owners to better perform their machines and investigate their daily performance. This business layer is also in charge of visualization and business dashboard. While visualization offers graphical functions for the data collected from the machines to facilitate the detection of abnormalities, the business dashboard provides an interface dedicated to its stakeholders (e.g., engineers who use machines), allowing them to monitor and make decisions on the adjustment policies and symptoms.

4.3 Discussion

After a comprehensive study of the state of the art in autonomous manufacturing automation, specially in what concerns self-configuration and self-diagnosis of manufacturing systems, it was found that a generic framework or implementation guideline is somehow missing. Most use cases are application specific and therefore it is not straightforward to apply them to heterogeneous contexts. Also, emerging requirements such as interoperability, scalability, integration and digitization as part of the fourth industrial revolution need to be considered in current and future industrial applications to take advantages of new technologies like Internet of things, high speed of internet connectivity, cloud computing, etc. This will clearly promote the creation of new business models i.e. plug to order and the promotion of factors such as modularity and sustainability highly relevant considering the sustainable development goals from the united nations.

In this context the creation of a guideline, or standardized framework is imperative, so that companies and industries can easily apply best practices to achieve new levels of automation/autonomy. Sections 4.1 and 4.2 intent to fulfill these objectives by matching main characteristics of autonomous manufacturing automation systems with well known reference frameworks i.e. Rami 4.0, MAPE-K. Those have considered best practices of state of the art works and some of the missing gaps to complement and potentiate their requirements. Although specific implementation details are missing, we believe that interested practitioners can take advantages of such mappings to see various of benefits that the implementation of CPS have.

In this regard the formalization (flow of information) for both the self-configuration and self-diagnosis concepts can be seen in Figs. 4.1 and 4.2. Necessary function blocks are also described. On the other hand technologies, requirements and standards required can be exemplified in Figs. 4.4 and 4.5 as well as overview of descrption of main characteristics to consider.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 814078

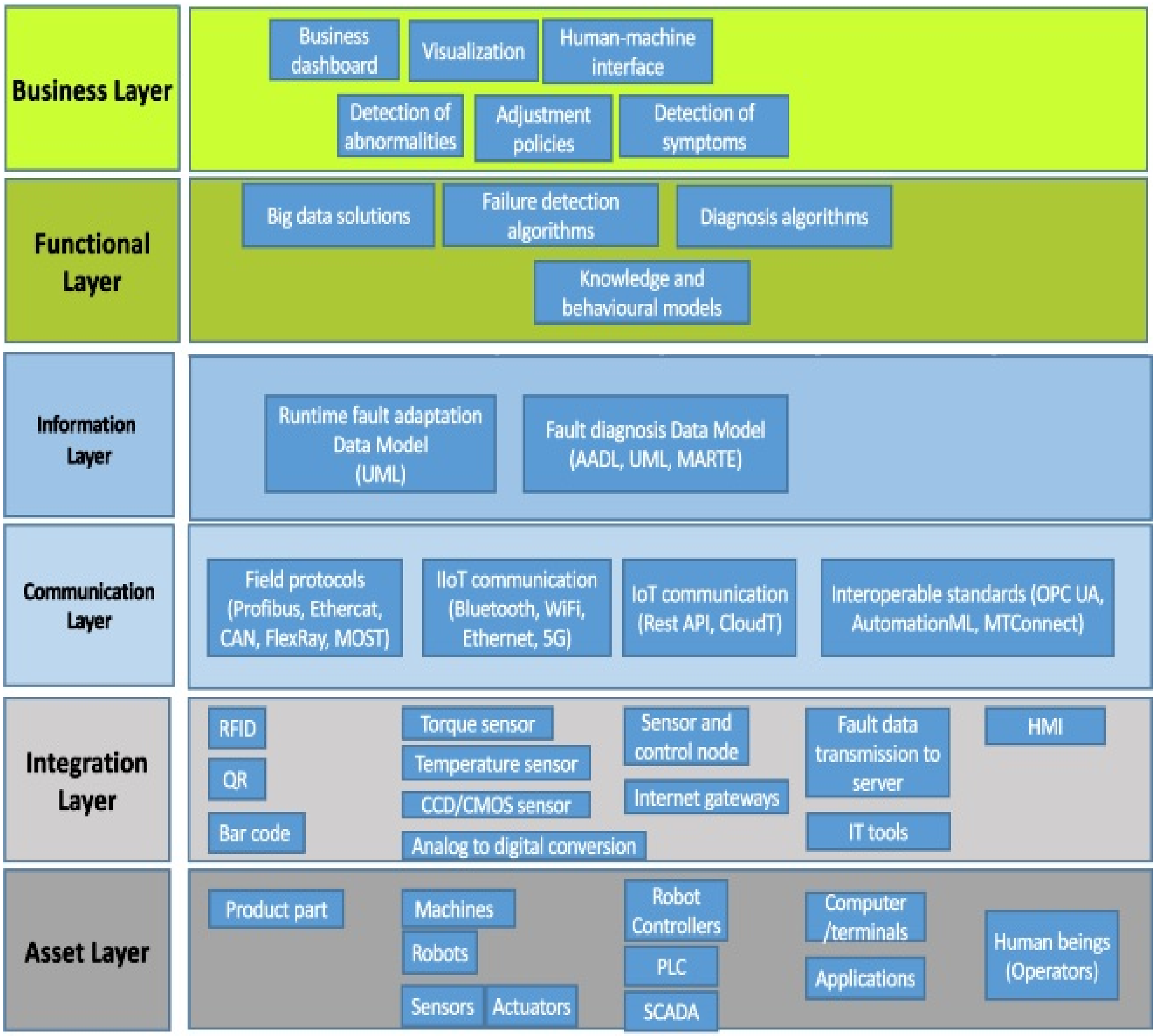

Figure 4.5: Rami 4.0 architecture layers for self-diagnosis

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 814078

Chapter 5

Conclusions

In this research, a basis for the development of a generic and homogeneous approach for self-x behaviour in smart manufacturing has been presented. It is based on a comprehensive literature review on smart manufacturing applications, specifically in the context of self-configuration and self-diagnosis. Those are used as the theoretical foundation to build the basic knowledge to understand the requirements and emerging technologies. As a result, key self-x requirements have been mapped into the well established MAPE-K and Rami 4.0 industrial compliant frameworks. We believe that the presented approach will likely guide industrial practitioners in the development and deployment of CPS towards truly autonomous manufacturing systems.

As future lines, we aim at developing self-contained CPS services for its adoption in industry. We will therefore exploit the presented guidelines on an industrial demonstrator as a proof of concept, being self-x the main enabler of autonomous manufacturing systems.

Bibliography

[1] T. Usländer and U. Epple, "Reference model of industrie 4.0 service architectures: Basic concepts and approach," *At-Automatisierungstechnik*, vol. 63, pp. 858–866, 10 2015.

[2] Plattform Industrie 4.0, "DIN SPEC 91345:2016-04: Reference Architecture Model Industrie 4.0 (RAMI4.0)," 2016. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/din-spec-rami40.html

[3] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, and K. Ueda, "Cyber-physical systems in manufacturing," *CIRP Annals*, vol. 65, pp. 621–641, 2016.

[4] Laboratory for Machine Tools and Production Engineering RWTH Aachen University. [Online]. Available: https://www.wzl.rwth-aachen.de/cms/~sijq/wzl/lidx/1/

[5] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 3, pp. 18–23, 2015.

[6] G. Platform Industrie4.0 Karsten Schweichhart, "Reference architectural model industrie 4.0 (rami 4.0)," 2016.

[7] A. Christoph Binder Salzburg University of Applied Sciences, "Introduction to the rami 4.0 toolbox," 2020.

[8] D. G. D. L. Iglesia and D. Weyns, "MAPE-K Formal Templates to Rigorously Design Behaviors for Self-Adaptive Systems," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 10, no. 3, pp. 1–31, oct 2015. [Online]. Available: https://dl.acm.org/doi/10.1145/2724719

[9] P. Arcaini, E. Riccobene, and P. Scandurra, "Modeling and Analyzing MAPE-K Feedback Loops for Self-Adaptation," in *2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. IEEE, may 2015, pp. 13–23. [Online]. Available: http://ieeexplore.ieee.org/document/7194653/

[10] S. Jahan, I. Riley, C. Walter, R. F. Gamble, M. Pasco, P. K. McKinley, and B. H. Cheng, "MAPE-K/MAPE-SAC: An interaction framework for adaptive systems with security assurance cases," *Future Generation Computer Systems*, vol. 109, pp. 197–209, aug 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167739X19320527

[11] J. Kephart and D. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, jan 2003. [Online]. Available: http://ieeexplore.ieee.org/document/1160055/

[12] "Self-adaptation for cyber-physical systems: A systematic literature review," *Proceedings - 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2016*, no. May, pp. 75–81, 2016.

[13] L. A. Estrada-Jimenez, T. Pulikottil, N. N. Hien, A. Torayev, H. U. Rehman, F. Mo, S. N. Hojjati, and J. Barata, "Integration of cutting-edge interoperability approaches in cyber-physical production systems and industry 4.0," in *Design, Applications, and Maintenance of Cyber-Physical Systems*. IGI Global, 2021, pp. 144–172.

[14] A. Maffei, "Characterisation of the business models for innovative, non-mature production automation technology," Ph.D. dissertation, KTH Royal Institute of Technology, 2012.

[15] H. U. Rehman, J. C. Chaplin, L. Zarzycki, and S. Ratchev, "A framework for self-configuration in manufacturing production systems," in *Doctoral Conference on Computing, Electrical and Industrial Systems*. Springer, 2021, pp. 71–79.

[16] C. on Networked Systems of Embedded Computers, N. R. Council, C. Science, T. Board, D. on Engineering, and P. Sciences, *Embedded, everywhere: A research agenda for networked systems of embedded computers*. National Academy Press, 2001.

[17] Z. Qin and Y. Lu, "Self-organizing manufacturing network: A paradigm towards smart manufacturing in mass personalization," *Journal of Manufacturing Systems*, vol. 60, pp. 35–47, 2021.

[18] K. T. Park, S. H. Lee, and S. D. Noh, "Information fusion and systematic logic library-generation methods for self-configuration of autonomous digital twin," *Journal of Intelligent Manufacturing*, pp. 1–31, 2021.

[19] M. Onori, N. Lohse, J. Barata, and C. Hanisch, "The ideas project: Plug & produce at shop-floor level," *Assembly Automation*, vol. 32, pp. 124–134, 2012.

[20] C. Schou and O. Madsen, "A plug and produce framework for industrial collaborative robots," *International Journal of Advanced Robotic Systems*, vol. 14, no. 4, p. 1729881417717472, 2017. [Online]. Available: https://doi.org/10.1177/1729881417717472

[21] D. Sanderson, J. C. Chaplin, and S. Ratchev, "A function-behaviour-structure design methodology for adaptive production systems," *The International Journal of Advanced Manufacturing Technology*, vol. 105, no. 9, pp. 3731–3742, 2019.

[22] P. Ferreira, N. Lohse, M. Razgon, P. Larizza, and G. Triggiani, "Skill based configuration methodology for evolvable mechatronic systems," in *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*. IEEE, 2012, pp. 4366–4371.

[23] H. Tang, D. Li, S. Wang, and Z. Dong, "Casoa: an architecture for agent-based manufacturing system in the context of industry 4.0," *IEEE Access*, vol. 6, pp. 12 746–12 754, 2017.

[24] S. Scheifele, J. Friedrich, A. Lechler, and A. Verl, "Flexible, self-configuring control system for a modular production system," *Procedia Technology*, vol. 15, pp. 398–405, 2014.

[25] T. Müller, N. Jazdi, J.-P. Schmidt, and M. Weyrich, "Cyber-physical production systems: enhancement with a self-organized reconfiguration management," *Procedia CIRP*, vol. 99, pp. 549–554, 2021.

[26] Y. Zhang, Z. Guo, J. Lv, and Y. Liu, "A framework for smart production-logistics systems based on cps and industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4019–4032, 2018.

[27] N. Antzoulatos, E. Castro, D. Scrimieri, and S. Ratchev, "A multi-agent architecture for plug and produce on an industrial assembly platform," *Production Engineering*, vol. 8, no. 6, pp. 773–781, 2014.

[28] M. Wojtynek, J. J. Steil, and S. Wrede, "Plug, plan and produce as enabler for easy workcell setup and collaborative robot programming in smart factories," *KI - Kunstliche Intelligenz*, vol. 33, pp. 151–161, 6 2019.

[29] J. Pfrommer, D. Stogl, K. Aleksandrov, S. E. Navarro, B. Hein, and J. Beyerer, "Plug & produce by modelling skills and service-oriented orchestration of reconfigurable manufacturing systems," *At-Automatisierungstechnik*, vol. 63, pp. 790–800, 10 2015.

[30] D. Sanderson, J. C. Chaplin, L. D. Silva, P. Holmes, and S. Ratchev, "Smart manufacturing and reconfigurable technologies: Towards an integrated environment for evolvable assembly systems." Institute of Electrical and Electronics Engineers Inc., 12 2016, pp. 263–264.

[31] H. Kaindl, M. Vallée, and E. Arnautovic, "Self-representation for self-configuration and monitoring in agent-based flexible automation systems," *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, vol. 43, pp. 164–175, 2013.

[32] "Self-diagnosis," 2020. [Online]. Available: https://en.wikipedia.org/wiki/Self-diagnosis.

[33] L. Christiansen, A. Fay, B. Opgenoorth, and J. Neidig, "Improved diagnosis by combining structural and process knowledge," in *ETFA2011*, 2011, pp. 1–8.

[34] P. W. Kalgren, C. S. Byington, M. J. Roemer, and M. J. Watson, "Defining PHM, a lexical evolution of maintenance and logistics," in *AUTOTESTCON (Proceedings)*, 2006, pp. 353–358.

[35] T. P. Fries, "Multi-agent fault diagnosis in manufacturing systems using soft computing," in *2007 international conference on integration of knowledge intensive multi-agent systems*. IEEE, 2007, pp. 168–173.

[36] Y. Jeong, S. Son, E. Jeong, and B. Lee, "An integrated self-diagnosis system for an autonomous vehicle based on an iot gateway and deep learning," *Applied Sciences*, vol. 8, no. 7, p. 1164, 2018.

[37] C. P. Gatica and A. Boschmann, "Enabling self-diagnosis of automation devices through industrial analytics," in *Machine Learning for Cyber Physical Systems*. Springer, 2019, pp. 107–115.

[38] J. Long, Y. Chen, Z. Yang, Y. Huang, and C. Li, "A novel self-training semi-supervised deep learning approach for machinery fault diagnosis," *International Journal of Production Research*, pp. 1–14, 2022.

[39] J. Lee, J. Ni, D. Djurdjanovic, H. Qiu, and H. Liao, "Intelligent prognostics tools and e-maintenance," *Computers in Industry*, vol. 57, no. 6, pp. 476–489, 2006, e-maintenance Special Issue. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361506000522

[40] R. Yu, B. Iung, and H. Panetto, "A multi-agents based e-maintenance system with case-based reasoning decision support," *Engineering Applications of Artificial Intelligence*, vol. 16, no. 4, pp. 321–333, 2003, intelligent Manufacturing. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197603000794

[41] K. Feldmann and J. Göhringer, "Internet based diagnosis of assembly systems," *CIRP Annals*, vol. 50, no. 1, pp. 5–8, 2001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0007850607620587

[42] S. Zhou, J. Zhang, and S. Wang, "Fault diagnosis in industrial processes using principal component analysis and hidden markov model," in *Proceedings of the 2004 American Control Conference*, vol. 6, 2004, pp. 5680–5685 vol.6.

[43] T. P. Fries, "Multi-agent fault diagnosis in manufacturing systems using soft computing," in *2007 International Conference on Integration of Knowledge Intensive Multi-Agent Systems*.

[44] S. Ploix, S. Gentil, and S. Lesecq, "Isolation decision for a multi-agent-based diagnostic system," *IFAC Proceedings Volumes*, vol. 36, no. 5, pp. 465–470, 2003, 5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes 2003, Washington DC, 9-11 June 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667017365357

[45] S. McArthur and E. Davidson, "Multi-agent systems for diagnostic and condition monitoring applications," in *IEEE Power Engineering Society General Meeting, 2004.*, 2004, pp. 50–54 Vol.1.

[46] R. Pinto, J. P. Reis, V. Sousa, R. Pinto da Silva, and G. Gonçalves, "Self-diagnosis and automatic configuration of smart components in advanced manufacturing systems," 10 2015.

[47] J. Barata, L. Ribeiro, and M. Onori, "Diagnosis on evolvable production systems," in *2007 IEEE International Symposium on Industrial Electronics*, 2007, pp. 3221–3226.

[48] D. Kaufmann, I. Nica, and F. Wotawa, "Intelligent Agents Diagnostics—Enhancing Cyber-Physical Systems with Self-Diagnostic Capabilities," *Advanced Intelligent Systems*, vol. 3, no. 5, p. 2000218, may 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/aisy.202000218

[49] E. A. Hinojosa-Palafox, O. M. Rodríguez-Elías, J. A. Hoyo-Montaño, J. H. Pacheco-Ramírez, and J. M. Nieto-Jalil, "An Analytics Environment Architecture for Industrial Cyber-Physical Systems Big Data Solutions," *Sensors*, vol. 21, no. 13, p. 4282, jun 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/13/4282

[50] B. Dowdeswell, R. Sinha, and S. G. MacDonell, "Finding faults: A scoping study of fault diagnostics for Industrial Cyber–Physical Systems," *Journal of Systems and Software*, vol. 168, p. 110638, oct 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0164121220301114

[51] Q. Lin, Y. Zhang, S. Yang, S. Ma, T. Zhang, and Q. Xiao, "A self-learning and self-optimizing framework for the fault diagnosis knowledge base in a workshop," *Robotics and Computer-Integrated Manufacturing*, vol. 65, p. 101975, oct 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0736584519305666

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 814078

[52] I. Djelloul, Z. Sari, and K. Latreche, "Uncertain fault diagnosis problem using neuro-fuzzy approach and probabilistic model for manufacturing systems," *Applied Intelligence*, vol. 48, no. 9, pp. 3143–3160, sep 2018. [Online]. Available: http://link.springer.com/10.1007/s10489-017-1132-8

[53] R. Pinto, J. Reis, V. Sousa, R. Silva, and G. M. Gonçalves, "Self-diagnosis and automatic configuration of smart components in advanced manufacturing systems," in *INTELLI 2015, The Fourth International Conference on Intelligent Systems and Applications*, 2015.

[54] D. McFarlane, S. Sarma, J. L. Chirn, C. Wong, and K. Ashton, "The intelligent product in manufacturing control and management," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 49–54, 2002.

[55] J. Leng, P. Jiang, C. Liu, and C. Wang, "Contextual self-organizing of manufacturing process for mass individualization: a cyber-physical-social system approach," *Enterprise Information Systems*, vol. 14, pp. 1124–1149, 9 2020.

[56] M. Doran, R. Sterritt, and G. Wilkie, "Autonomic architecture for fault handling in mobile robots," *Innovations in Systems and Software Engineering*, vol. 16, no. 3, pp. 263–288, 2020.

[57] T. A. Nguyen, M. Aiello, T. Yonezawa, and K. Tei, "A self-healing framework for online sensor data," in *2015 IEEE International Conference on Autonomic Computing*. IEEE, 2015, pp. 295–300.

[58] L. Castaneda and G. Tamura, "A reference architecture for component-based self-adaptive software systems," *Master's Thesis, Universidad Icesi, Colombia*, 2012.

[59] T. A. Nguyen, M. Aiello, T. Yonezawa, and K. Tei, "A self-healing framework for online sensor data," in *2015 IEEE International Conference on Autonomic Computing*, 2015, pp. 295–300.

[60] T. A. Nguyen, D. Bucur, M. Aiello, and K. Tei, "Applying time series analysis and neighbourhood voting in a decentralised approach for fault detection and classification in wsns," in *Proceedings of the Fourth Symposium on Information and Communication Technology*, ser. SoICT '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 234–241. [Online]. Available: https://doi.org/10.1145/2542050.2542080

[61] D. Kaufmann, I. Nica, and F. Wotawa, "Intelligent agents diagnostics—enhancing cyber-physical systems with self-diagnostic capabilities," *Advanced Intelligent Systems*, vol. 3, no. 5, p. 2000218, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/aisy.202000218

[62] X. Ye and S. H. Hong, "An automationml/opc ua-based industry 4.0 solution for a manufacturing system," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1. IEEE, 2018, pp. 543–550.

[63] Y. Wang, T. Towara, and R. Anderl, "Topological approach for mapping technologies in reference architectural model industrie 4.0 (rami 4.0)," 2017.

[64] F. Zezulka, P. Marcon, I. Vesely, and O. Sajdl, "Industry 4.0 – an introduction in the phenomenon," *IFAC-PapersOnLine*, vol. 49, no. 25, pp. 8–12, 2016, 14th IFAC Conference on Programmable Devices and Embedded Systems PDES 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896316326386

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 814078

[65] K. Tei and L. Gürgen, "Clout: Cloud of things for empowering the citizen clout in smart cities," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*. IEEE, 2014, pp. 369–370.

[66] M. Moghaddam, M. N. Cadavid, C. R. Kenley, and A. V. Deshmukh, "Reference architectures for smart manufacturing: A critical review," *Journal of manufacturing systems*, vol. 49, pp. 215–225, 2018.

[67] P. H. Feiler, B. A. Lewis, and S. Vestal, "The sae architecture analysis & design language (aadl) a standard for engineering performance critical systems," in *2006 ieee conference on computer aided control system design, 2006 ieee international conference on control applications, 2006 ieee international symposium on intelligent control*. IEEE, 2006, pp. 1206–1211.

[68] J. Delange and P. Feiler, "Architecture fault modeling with the aadl error-model annex," in *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*. IEEE, 2014, pp. 361–368.

[69] A. Joshi, S. Vestal, and P. Binns, "Automatic generation of static fault trees from aadl models," 2007.

[70] X. Wu, J. Chen, R. Li, W. Sun, G. Zhang, and F. Li, "Modeling a web-based remote monitoring and fault diagnosis system with uml and component technology," *Journal of Intelligent Information Systems*, vol. 27, no. 1, pp. 5–19, 2006.

[71] M. Milovančević, J. S. Marinović, J. Nikolić, A. Kitić, M. Shariati, N. T. Trung, K. Wakil, and M. Khorami, "Uml diagrams for dynamical monitoring of rail vehicles," *Physica A: Statistical Mechanics and its Applications*, vol. 531, p. 121169, 2019.

[72] T. Ma, S. Ali, and T. Yue, "Modeling foundations for executable model-based testing of self-healing cyber-physical systems," *Software & Systems Modeling*, vol. 18, no. 5, pp. 2843–2873, 2019.

[73] M. Z. Iqbal, S. Ali, T. Yue, and L. Briand, "Experiences of applying uml/marte on three industrial projects," in *International Conference on Model Driven Engineering Languages and Systems*. Springer, 2012, pp. 642–658.

[74] B. Dowdeswell, R. Sinha, and S. G. MacDonell, "Finding faults: A scoping study of fault diagnostics for industrial cyber–physical systems," *Journal of systems and software*, vol. 168, p. 110638, 2020.

[75] M. Illarramendi, L. Etxeberria, X. Elkorobarrutia, and G. Sagardui, "Runtime observable and adaptable uml state machines: models@ run. time approach," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 2019, pp. 1818–1827.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 814078