The Digital Manufacturing and Design Training Network

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

Deliverable 3.3

Guide to develop and deploy CPS resources

Lead parties for Deliverable: KTH
Deliverable due date: April 2023
Actual submission date:
Dissemination level: Public

Contents

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 focuses on the development of a guide aimed at the implementation of services that exploit the CPS architecture and enable the analysis of production performance, monitoring and optimization activities. The included services should be self-contained and combinable in order to maximize their reusability and service aggregation. The guide aims at directing companies (especially SMEs) in the implementation of two services, namely self-configuration and self-diagnosis, instrumental for the deployment of a CPS architecture and ultimately to the industrial adoption of CPS, by enabling fast integration, re-configurability and scalability of automatic production resources.

Team involved in deliverable writing

ESR5: Miriam Ugarte Querejeta, Mondragon Unibertsitatea
ESR8: Fabio Marco Monetti, KTH Royal Institute of Technology
ESR9: Sylvia Nathaly Rea Minango, KTH Royal Institute of Technology
ESR10: Luis Alberto Estrada Jimenez, Universidade NOVA de Lisboa
Supervisor: prof. Antonio Maffei, KTH Royal Institute of Technology

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

Foreword

DiManD aims to develop a high-quality multidisciplinary, multi-professional and cross-sectorial research and training framework for Europe. The purpose is to improve Europe's industrial competitiveness by designing and implementing an integrated programme in the area of intelligent informatics driven manufacturing, which will form the benchmark for training future Industry 4.0 practitioners. This will be done in compliance with the industrial requirements such revolutionary production systems will pose, and in specific this deliverable will represent one further step forward, by attempting to crystallize a finite set of system requirements, derived from real-world conditions, to be leveraged for the correct implementation of self-configuration and self-diagnostic services.

1 Introduction

In this deliverable, we will investigate the development of services that leverage the Cyber-Physical Systems (CPS) architecture. The primary focus will be on two key services: self-configuration and self-diagnosis. These services play a crucial role in enabling the analysis of production performance [1], monitoring [2], predicting the outcome [3] and optimizing activities within CPS environments [4], thus enabling the future of intelligent, reconfigurable manufacturing systems [5, 6, 7]. By creating services that can work independently and be easily combined with other services, we can reuse and aggregate their capabilities to create bigger services [8, 9].

This deliverable is part of Work Package 3 (WP3), dedicated to the development and evaluation of a CPS architecture. It builds upon the outcomes of previous tasks, which involved a comprehensive analysis of the state-of-the-art in CPS and its adoption in industry, and the identification of requirements for a CPS architecture bridging the gap between existing industrial standards and ICT infrastructures, enabling fast integration and configuration of CPS resources [10, 11, 12].

This deliverable serves as a guide that will aid industrial practitioners in the development and deployment of CPS resources using the previously developed CPS architecture. This architecture will be extended and further developed in the present document. The guide will provide instructions for the implementation of self-configuration and self-diagnosis capabilities in CPS environments. By following this guide, practitioners will be able to deploy services within a CPS for analyzing production performance, monitoring activities, and optimizing industrial processes.

Cyber-Physical Systems have emerged as a critical technology in modern industrial settings, as these systems integrate physical components with advanced computing, and communication technologies, enabling companies to scale up their levels of automation, efficiency, and flexibility [13, 14, 15]. CPS have huge potential to greatly influence current and future industry practice, however their successful integration in the current manufacturing environment still requires proper guidance [16].

The complexity of CPS architectures and the variety of technologies involved pose significant challenges to industrial practitioners. To address these challenges, there is a need for a comprehensive guide that provides step-by-step instructions and insights into the development and deployment of CPS resources. This deliverable is aimed at facilitating the adoption of CPS by enabling fast integration, re-configurability, and scalability of automatic production resources.

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

Therefore, the main objectives of this deliverable are as follows.

- Develop self-contained and combinable services that work in the CPS architecture described in D3.2.
- Describe the upgrade of the CPS architecture and provide practical guidelines for the development and deployment of CPS resources.
- Present the implementation requirements of self-configuration and self-diagnosis capabilities to allow the analysis of production performance, monitoring, and optimization activities.
- Maximize the reusability and service aggregation of the developed CPS services.

This guide will therefore cover topics such as the selection of appropriate technologies, the design of the CPS architecture to support them, and the development of self-contained and combinable services that enable the analysis of production performance, monitoring and optimization activities. As part of the transformation inspired by the fourth industrial revolution, it will also consider factors such as interoperability, scalability, integration and digitization; since it will also consider the rise of what has been called Industry 5.0, it will explore the role of humans within the described CPS system, emphasizing their significance as central to the activities involved.

As mentioned, this deliverable will focus on two key services within CPS: self-configuration and self-diagnosis. Self-configuration aims to enable automatic rearrangement of hardware and software configurations in response to demanding manufacturing requirements. Self-diagnosis focuses on the automatic detection, understanding of root causes of failures, and resolution of faults. With the increasing importance of CPS in industrial settings, the development of a comprehensive guide for their implementation is essential, and therefore with this document we hope to give practitioners an easy-to-follow, schematized and structured set of instructions and recommendations for properly developing CPS resources.

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

2 Self-X services and reference architectures

The implementation of self-configuration and self-diagnostic services within this WP necessitates a comprehensive description of the underlying architecture, including inputs, outputs, required machinery and equipment, and information flow. This architectural framework can be applied to any other service that provides self-X behavior within a CPS. As previously documented [10], services like the ones featured in this deliverable facilitate enhanced interconnection and interoperability among machines within an autonomous system. Moreover, their functionalities can be easily utilized both within and outside a company, enabling CPS to provide highly scalable web services.

In [11], self-configuration and self-diagnosis were described as two general requirements of autonomous cyber-physical production systems, and were referred to as Autonomy Requirement (AR) number 5 and Autonomy Requirement number 6:

AR5 self-configuration, refers to the capacity of autonomously configuring and adjusting components and systems, including their auto re-adjustment, if necessary. In manufacturing systems, it can apply to modules that can start working without requiring explicit programming;

AR6 self-diagnosis, involves the capacity of a system to understand and detect failures, examine the status of machines, and identify the root cause of the failure.

For a detailed explanation and an extensive overview of the two services within the architectural framework, please refer to the previous deliverable [11].

Deliverable D3.1 [10] provided a shortened list of requirements for the self-X services under consideration, based on a combination of existing literature and firsthand experience of the researchers and practitioners involved in the deliverable writing. The following deliverable [12] presented a formalization of the self-X behaviors using the MAPE-K framework and a mapping with RAMI 4.0, outlining the technologies and standards needed to deploy smart manufacturing applications. However, it also highlighted that a comprehensive, generic implementation guideline is lacking. Therefore, the purpose of this guide is to bridge this gap by presenting manufacturers with a set of best practices for achieving improved levels of autonomy. It would also be beneficial to utilize this guide for enhancing the autonomy of a manufacturing system and subsequently evaluating it in combination with a recently developed maturity model for the autonomy of manufacturing systems, which represents an additional outcome of this project [17]. With this guide, we aim to provide specific implementation instructions that will enable interested practitioners to leverage the benefits of CPS implementation.

In this manuscript, we revisit and expand upon the previously identified requirements, thoroughly examining and extending the list to ensure its comprehensiveness. Our goal is to provide companies with a user-friendly and easily understandable compilation of requirements that can serve as a practical guide for implementing self-configuration and self-diagnosis in autonomous systems. This guide has been designed and written to facilitate future effortless extension and inclusion of other self-X and smart services.

The two complete lists of requirements are presented in the following Sections, and extensively explained in the text flow. Additionally, the reader will find them presented in Table 1 and 2. These tables can be read like this: from left to right the reader will find that each row contains a generic high-level *requirements*, which is sub-divided into more specific *sub-requirements*, whose recording is essential to extrapolate the important information needed for defining what *resources* are necessary to the implementation. Each resource is also represented in an *example*, which makes the interpretation of such information accessible and straightforward. Finally, the last four columns specify the *inputs* necessary to any specific resource to properly function, as well as the *input sources*, followed by the outgoing *outputs* and the *output sources* providing them. With such a structure, we believe that the information can be accessed and interpreted easily for SMEs and big companies alike. In a few cases it is not applicable or not possible to to uniquely tell the input and output sources for a specific sub-requirements, therefore some columns may present null values corresponding to such rows. However, before listing all the requirements, it appears necessary to frame the novel architecture that we will be using in the deliverable. Therefore, first the architecture will be described in the following Section 2.1, and then the two main Tables containing the requirements will be presented and described in detail in Sections 3, and 4.

2.1 Proposed novel architecture

Before delving into the description of the specific requirements, examples, inputs and outputs of the two proposed services, we present a high-level architectural representation of how one example-service like this would fare within an autonomous system. It is composed of several layers, including an asset layer consisting of a sensors layer and machine layer that encompass all necessary resources for data collection. Additionally, there is an edge layer located on-site, as well as an external fog and cloud layer.

This architecture draws inspiration on the previous work performed during the DiManD project, through the previously mentioned deliverables, as well as previous research efforts presented in such work as Qi and Tao (2019) [18], Caggiano (2018) [19], and others [20, 21].

Qi and Tao (2019) present a reference architecture for smart manufacturing systems that incorporates edge computing, fog computing, and cloud computing. In our proposal, we adopt their definitions to incorporate this hierarchical computing composition, and include edge, fog, and cloud components. In summary, smart manufacturing systems consists of multiple layers of devices and computational assets. The foundation lies in the smart equipment portfolio, where data are sourced and edge computing takes place. This is followed by a transmission layer for data transfer and where fog computing occurs. The final layer is the cloud, where big data is stored and analyzed. Through the integration of edge computing and fog computing, only essential information is sent to the cloud, reducing the data flow and minimizing service downtime while maintaining system robustness. Edge computing, fog computing, and cloud computing collaborate to meet the requirements of smart manufacturing applications more effectively.

Caggiano (2018) presents how cloud-based manufacturing processes are monitored for smart services like the ones we investigate. Here, the cloud manufacturing architecture is layered in a hierarchy composed of: (i) physical resources; (ii) local servers; and (iii) cloud servers; to allow for a shared computational effort between resources. In this case, the cloud manufacturing server utilizes sensor data collected at the factory level to provide timely online diagnosis of tool conditions. Such diagnosis is achieved through knowledge-based algorithms and other pattern recognition paradigms. Assisted by the cloud-based services, the local server initiates appropriate

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

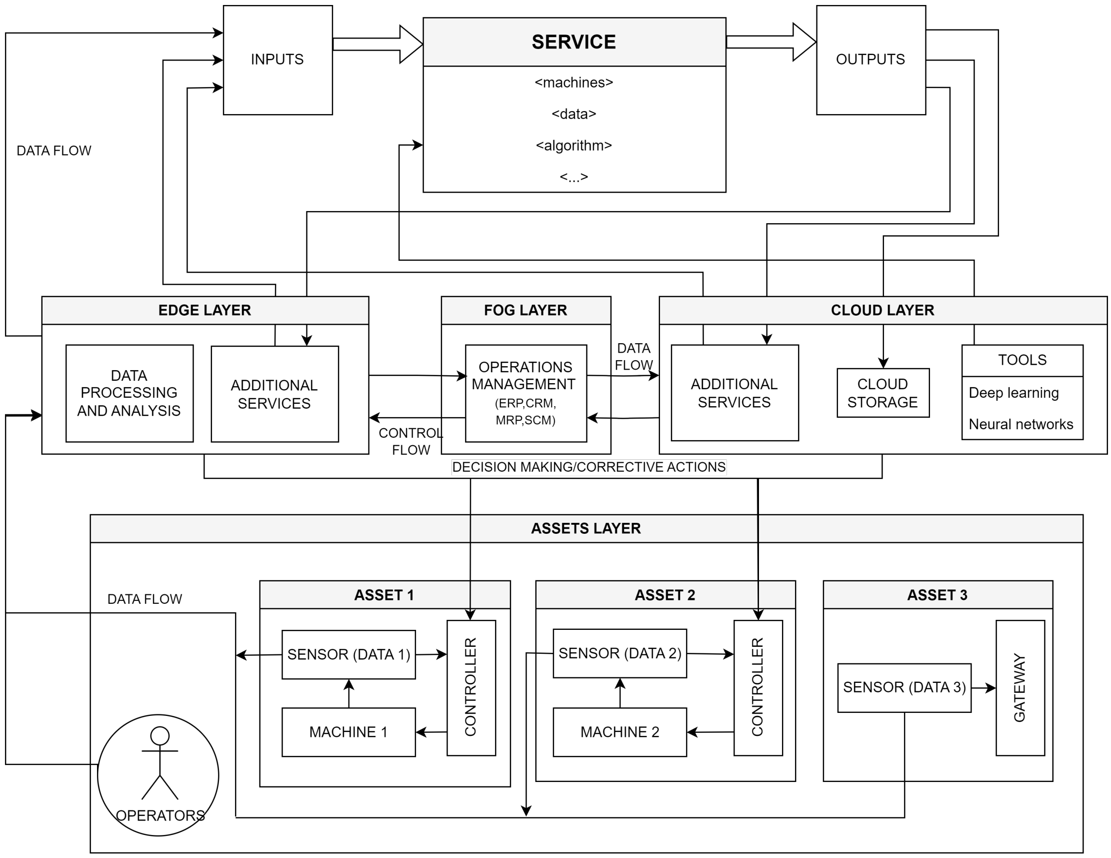

Figure 1: Schematized representation of the novel architecture for self-X services.

restoring actions, such as tool replacement, process interruption, or parameter adjustments. The server then sends the necessary command to the machine tool control for implementation.

The extended cloud-based smart services architecture proposed by this deliverable is presented in Figure 1. The computing and service assets in the cloud are interconnected with the physical equipment in the asset layer, such as machines and sensors, forming an advanced CPS. The discussed layered structure of the architecture offers shared computational effort between resources, managed remotely: online communication reduces the physical distance between locations and allows to share results and information at the highest possible speed. The physical resources and the local layer are both available at the factory shop floor, while the fog servers and the cloud storage and services are possibly located elsewhere.

The asset layer comprises machines, transportation units, and other equipment, all equipped with sensors to collect relevant process and machine status data. The local server at the factory shop floor performs initial data preprocessing and easy computing tasks. Working in parallel with the fog servers, it filters out unimportant or redundant data and sends the essential data packages to the cloud servers. In some cases, it might also happen to have a database (with a smaller capacity compared to cloud servers) in the edge, for raw-data collection and storage, such that then data preprocessing can be performed within the edge layer only on sets of data taken in defined sampling periods.

In the cloud servers, which possess high computational resources, the service tasks are then performed. The manufacturing information is stored in a cloud database for monitoring and historical analysis, as well as serving as input for other cloud-based services. The critical data is analyzed, patterns are detected, and important outputs, such as fault root cause recognition, are generated.

Once the computing task is completed, the diagnostic output is sent back to the local server, which utilizes the results for decision-making and suggests corrective actions. The operator can visualize the suggested actions on the terminal and receive warnings if human intervention is required. After the intervention, the operator can confirm the status of the maintenance activity and the operative system, effectively closing the loop with the system.

2.2 Integration with other services

The services in our architecture have the ability to integrate with other services to improve their analysis capabilities and provide more extended insights and competences. Taking the self-diagnostic service as an example, its outputs can be published to the cloud layer for storage, analysis, and integration with additional services. These outputs can then be re-utilized as inputs to many downstream applications and services, such as maintenance planning service, or performance analytics service, available to enterprises on the cloud, for a marginal cost. To further specify the benefits, please refer to the following examples, only partial indication of a more extended ecosystem of available services.

1. *Predictive maintenance*: self-diagnostic services can incorporate predictions from a predictive maintenance service, which uses machine learning algorithms to predict when a machine or component is likely to need care. The combined service can therefore provide proactive maintenance recommendations. Moreover, it can generate optimized maintenance schedules and work orders based on the diagnostic results.
2. *Performance analytics*: self-diagnostic services cooperate with a performance analytics service to provide insights on the OEE of machines and to identify trends and monitoring the long-term performance of the system to identify potential process improvements.
3. *Energy management*: self-diagnostic services can integrate with an energy management service to analyze energy consumption patterns of robots and other equipment alike. Then, it can identify excessive energy usage and highlight inefficiencies.
4. *Storage management*: self-diagnostic services can gather data from a storage management service to check the availability of needed spare parts and components. This combination allows to provide recommendations about the availability and replenishment of materials.

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 Self-configuration

3.1 Definition of self-configuration

The self-configuration service is responsible for automatically configuring and adjusting the settings of machines and equipment to optimize their performance and adapt to changing operational conditions. The system is capable of autonomously starting the operating mode [22], acquiring the related configuration parameters, and initializing itself to provide the desired services [23], or dynamically readjust its settings to react to unexpected, changing conditions [24]. In order to do so, it requires as inputs data from a sensor framework, as well as historical configuration records, operational machine information, and contextual information, thus acquiring the capability to determine the optimal configuration settings. Within the service is utilized a series algorithms and rules to analyze the inputs and generate configuration recommendations or automatically apply the new specified settings. Outputs from the self-configuration service are sent to the machines and the equipment in the shop-floor for implementation and can also be shared with other services for further analysis or integration.

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 1: Requirements for self-configuration.

Requirements	Sub-requirement	Resources	Example	Stakeholders	Inputs	Input source	Outputs	Output source
Interoperability	Semantic interoperability	Standard data models, ontologies	Existing data models, existing ontologies	Data engineers, IT Engineers, Software Engineer	Raw data	sensors, resource data	Structured, organized information in standard models	Post-processors, translators, converters
	Syntactic interoperability	Standard data formats	AutomationML, Unified modeling language	Data engineers, IT Engineers, Software Engineer	Raw data in multiple formats	Sensors, systems, resource data	Data in standard format	Post-processors, translators, converters
	Transport interoperability	Standard communication protocols, standard communication interfaces	MQTT, OPC-UA	Equipment manufacturers, IT engineers, software engineers	Data flows, control signals	Sensors, controllers, end-point-equipment	Standardized data packages	Message servers, communication software
	Technical interoperability	Backward compatibility, standard physical interfaces, retrofitting	Plug-and-play devices, USB sensors/USB PLCs, APIs	System integrators, Equipment manufacturers, IT engineers, software engineers	Historical data in legacy formats	Legacy systems	Fully compatible data	Post-processors, translators, converters
Context awareness	Context information	Physical, logical, and/or virtual sensors, expert systems, data models	temperature, force, pressure measurements	IT Engineer, Software Engineer, Manufacturing Engineer	Raw data	Sensors	Context information	Expert systems, AI

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

Knowledge	Product information	Product models, product recipes, product documentation, BOM, raw material information, PMD systems, CAD/CAM/CAE software	CAD models, engineering drawings, material specifications, CAD software (e.g., CATIA, NX, Solidworks)	Manufacturing engineer, software engineer, designer, supplier, purchase	CAD models, drawings, BOM	CAD/CAE systems	Product information	PDM systems
	Process information	Process plan, process sheets, MBOM, CAD/CAM/CAE/CAPP software	NC data, process simulations, CAM software (e.g., MasteCAM, NX), robot programming software	Process planner, production personnel, manufacturing engineer	Requirements, product information	PDM systems, CAD/CAE systems	Process plan, NC programs	CAM/CAPP systems, MRP systems
	Resource information	Resources' skills/capabilities, production plan, expert knowledge, facility information, MRP systems	Machines' specifications, controllers specifications, plant layout	Equipment manufacturers, manufacturing engineer, system integrator	Process information, resources' technical specifications	CAD/CAM/CAPP systems	Routing plans, layout configurations	ERP systems, MRP systems
	Manufacturing information	Runtime conditions, KPIs, business information	Netsuite	Management, production planner	Production data	ERP systems, MRP systems	Financial reports, operation reports, business projections	Management software

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

Security	Infrastructure Security	Access control (authentication, authorization, auditing), data encryption, digital signatures, event logs, intrusion detection system, host firewalls, end-point security systems	Authentication protocols, encryption algorithms, log monitoring systems, antivirus, intrusion prevention systems	IT security team, system administrators, infrastructure manager	Access requests, data requests	Internal/external users, suppliers	Only authorized access granted	Security systems
	Identification and access management	Access policies, identity management system, authentication systems, privacy-protection mechanisms	Intrusion detection mechanisms, regular security checks	IT security team, system administrators	Access requests, data requests	Internal/ external users, suppliers	Only authorized access granted	Security systems
	Data protection and security	General Data Protection Regulation (GDPR)-compliant practices	Internal policies, good practices	IT security team, system administrators, network administrator	Business data	Internal/ external users, suppliers	Sensitive data protected	Security policies
	Cloud security as a service	Web security service, storage security, identification and access management service	Cryptographic algorithms	IT security team, system administrators, network administrator, infrastructure manager	Production data	Internal/ external users, suppliers	Only authorized access granted	Cloud-based security systems

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

Connectivity	Reliable IT infrastructure	Networking devices, software for network connectivity, endpoint equipment, reliable communication network (wired/wireless), consistent access to data (internet/LAN)	Network sniffers, antivirus	Network administrators, System integrators, IT personnel	Data flows, control signals	End-point devices, communication devices, software, security systems, data-generator devices	Reliable communication flow	IT infrastructure
	Secure and scalable WAN/LAN	Network monitoring, routing and access policies	Network monitoring software	Network administrators, System integrators, IT personnel	Data flows, control signals	Own IT infrastructure/ external services	Encrypted and secure information flow	IT infrastructure, ISP, external infrastructure providers
Data integration and management	Updated operative information	Product data management (PDM), manufacturing execution systems (MES), customer relationship management (CRM), supply chain management (SCM), enterprise resource planning (ERP) systems	Automation software	Network administrators, System integrators, IT personnel	production data	CAD/CAM/ CAE/CAPP systems,resource-generated data	production information	PDM, ERP, MES, CRM, SCM systems
	Storage	Storage (cloud/local), database structure	Google cloud, AWS, Azure	Network administrators, System integrators, IT personnel	Production data and information	Internal servers	Data backups	Cloud services, local servers

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

Robustness	Fault-tolerance	Infrastructure redundancy, flexible layout, responsive production planning	Error recovery, dynamic scheduling	Automation Engineer	Production orders	Internal/ external clients	Fault-free processes	All systems
Monitoring	Runtime condition monitoring, location/destination monitoring	IoT communication (real time), Positioning devices	Temp, vibration sensors, GPS	IT Engineer, Control engineer, Mechanical engineer	Sensors: vibration, temperature, speed, productivity.	IoT Monitoring services	Monitoring information	Tool wear, energy consumption, health machine status
Real time capability	Low latency, fault tolerance, Scalability, Security	Firewall, real time processing frameworks	Apache kafka, Apache flink, google cloud data flow, microsoft azure	IT Engineer, Software Engineer	Latency, security threats	Latency Services, Security threat services	Monitoring, actuation signals	Sensors information input, PLC outputs
Analysis	Adaptation identification, Performance evaluation, Feasibility analaysis	IoT platforms, Manufacturing execution system, Digital twin technologies	Cumulocity IoT Platform, Microsoft Azure IoT Suite, Google Cloud IoT Platform, Plex Smart Manufacturing Platform, Siemens Opcenter Execution, Proficy Smart Factory MES, aPriori Digital Manufacturing Simulation Software, Autodesk Digital Twin	IT Engineer, ML Engineer, Data analytics Engineer	Num and type of machines, Num and type of products, Productivity, Health Status of machines, Energy consumption of machines., Layout, Recipe/Set of tasks of the product., Product geometry, Products due date	Machine status service, Energy consumption service, Health status service., Task allocation service, Parameter configuration service, Optimal machine configuration service	Configuration information	machine status, energy, health, task allocation parameter configuration, optimal machine configuration

Planning	Component localization, Components grouping, Task prioritization, Balance load distribution, Workflow of operations planning, Generation of skills	Advance analytics and ML, Supply chain management system	Microsoft Azure Machine Learning Studio, RapidMiner	ML Engineer, Data analytics, production engineer	Components position, products due date, machine capacity, machine positioning, available skills	Production engineer	ERP	Scheduling, load balancing
Scalability	Cloud-based, Service-based	AWS, Azure, Google cloud, SAP Digital Manufacturing Cloud, IEC 61499, ROS	AWS, Azure, Google cloud, SAP Digital Manufacturing Cloud, IEC 61499, ROS	IT Engineer, Software Engineer, Control Engineer	—	—	—	—
Plug and produce	Intelligent infrastructure	Modular hardware, modular software	Multi-agent based, service-based technology	Automation engineer	Infrastructure	Available modules and skills	Intelligent infrastructure	Layout reconfiguration
Modularity	Agent-based elements, Service oriented, Fine granularity	MAS frameworks, Industrial and robotics platforms	Jade framework, IEC 61499, ROS	IT Engineer, Software Engineer, Automation Engineer	intelligent orchestrator	—	—	—
Self-learning	Continous data storage	Servers, cloud storage	Storage servers	Software engineer, automation engineer	Previous configuration stored	Production engineer	Configuration parameters of machines	Manufacturing modules, machines
Decentralization	Decentralized computation	Edge computing, fog computing, cloud computing	Google cloud, edge devices, Rasp pi, Arduino, smart controllers	IT engineer, software engineer	—	—	—	—

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

3.2 Requirements

R1. Interoperability: refers to the "ability of two or more systems or applications to exchange information and to mutually use the information that has been exchanged", as provided by the ISO 22123-1:2023 standard [25]. In this sense, information at all levels must be checked for consistency throughout the system. Consequently, the implementation of global standards is recommended to expedite information exchange and the potential scalability of the system through the integration of new components. Data flows in all forms need to comply with this requirement for which the infrastructure, software, and IT engineers will be responsible for defining the protocols, data models, file formats, and software packages reigning all the production system.

R2. Context awareness: to adopt a new configuration, the system must be aware of its environment and use this information to provide the best solution according to current circumstances [26]. A wide variety of sensors are available in the market to collect the required information, but it will be the organization, the one deciding which information is important and how it will be extracted from the raw data. Measured variables will depend upon the system's tasks and goals.

R3. Knowledge: the amount of data and information generated by the whole system must be processed, classified, and stored adequately, considering the different levels of the organization to which they are relevant. It is essential to identify which data is used to generate the desired inputs, the frequency in which it is collected, how and where it will be stored, and the guidelines for accessing these data. The role of the software platforms is critical for this requirement since they need to be fully compatible and generate a seamless integration of systems.

R4. Security: since the system relies on data circulating throughout the infrastructure, a well-defined security strategy is vital to protect the system and its users. Security must be assessed frequently, considering control access, authentication, data encryption, and GDPR policies. This assessment should be done at the internal organization level and at its cloud service providers [27]. Furthermore, security should not depend only on the IT department, but it needs to be part of the organizational culture making everybody aware of the existing risk and good practices to follow. Further guidelines can be found at ISO/IEC 29180:2012 standard, Information technology — Telecommunications and information exchange between systems — Security framework for ubiquitous sensor networks; ISO/IEC 27033:2015, Information technology — Security techniques — Network security; and ISO/IEC 27000 standard family, Information security management.

R5. Connectivity: access to data is crucial for a self-configuring system to respond according to the circumstances. For this, a reliable network is necessary, tolerant to faults, with the right cost-benefit in terms of the required infrastructure and system requirements.

R6. Data integration and management: all data generated in and by the system needs to be available for the relevant stakeholders at the different levels of the organization. Then, relevant and updated information needs to be spread through the data management platforms

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

at all levels, which is highly related to the interoperability and connectivity requirements mentioned previously. The frequency in which these systems should be updated needs to be assessed internally, considering the amount of data collected and the infrastructure capabilities. Additionally, how much of this information is stored for the long term, with which frequency and the backup method need to be considered to optimize costs without jeopardizing the system in case of faults.

R7. **Robustness**: the production system and its internal subsystems need to be able to handle disturbances and be fault-tolerant to maintain the productivity level at the desired range [28]. Several strategies can be considered, including dynamic scheduling and self-healing methods, most of which will require a flexible and redundant infrastructure at all levels.

R8. **Monitoring**: the elements of a system or the system itself are able to keep track of its own performance, logging process data to further analyze it [23]. Data capture devices like sensors are necessary to keep track of this information.

R9. **Real time capability**: the elements of a system can respond fast enough to an event without having a noticeable delay that compromises the normal functioning of the operations.

R10. **Analysis**: the term is taken from the context of autonomous computing. In manufacturing automation, it refers to the capacity of analyzing and interpreting manufacturing data from various sources to gain insights, identify trends, and malfunctions, and with that consider proper actions.

R11. **Planning**: unlike analysis, planning refers to the capacity of the manufacturing system to generate a sequence of actions that lead to a specific goal, e.g. reconfiguration strategies: change of position of stations, change of machine parameters, etc.

R12. **Scalability**: refers to the capacity of a system to adapt its infrastructure when adding or removing elements e.g., stations, modules, etc. efficiently enough to accommodate available resources and optimize its usability.

R13. **Plug and produce elements**: elements that can work once they are physically plugged. An intelligent infrastructure is able to recognize and orchestrate specific functionalities to the various manufacturing tasks.

R14. **Modularity**: self-contained manufacturing modules include necessary hardware and software to work stand-alone or to be integrated into an intelligent infrastructure.

R15. **Self-learning**: refers to the capacity of the system to extract knowledge e.g., from the operator decision-making and reuse it in an equal/similar context. This learning activity can increase process adaptability.

R16. **Decentralization**: decentralization in manufacturing is accomplished when individual elements within the system, such as workstations, machine tools, AGVs, and products, have the capability to make independent decisions in real-time, all while working towards

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

a shared organizational goal. In this setup, there is no central control unit governing these elements. The systems operate autonomously, even in the face of external disruptions, specific exceptions, or conflicting objectives, and are specifically designed to achieve overall objectives by utilizing localized operational information.

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 Self-diagnosis

4.1 Definition of self-diagnosis

The self-diagnostic service is designed to monitor and analyze the operational state of machines in real-time, enabling proactive maintenance and minimizing downtime. It utilizes advanced algorithms and machine learning techniques to detect anomalies, predict faults, find the root-cause of failures, and provide recommendations for maintenance actions. Self-diagnostic is responsible for analyzing the data collected through sensors from the machines and equipment to determine their health and performance. The service will take inputs from sensors built for monitoring variables as temperature, pressure, vibration, and any other relevant feature. The system then analyzes this data to find and predict any anomalies or issues and send an output to specify the root cause of the problem and possible solutions.

*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 2: Requirements for self-diagnosis.

Requirements	Sub-requirement	Resources	Example	Stakeholders	Inputs	Input source	Outputs	Output source
Connectivity	Sensor selection and placement	Sensors and machines	Sensors: position, torque, vibration. . .	Manufacturing engineers, Equipment technicians	Variables of interest (for faults)	Expert knowledge	Raw data	Sensors
	Sensor data and historical data	Edge/cloud storage	Cloud-based storage: Amazon S3, Google Cloud Storage, Microsoft Azure	Data analysts, Data scientists	Production data	Production	Raw and historical data	Sensors, Database
	Communication protocols	Standards and protocols	Profibus, Ethercat, CAN. . .	Automation engineers, Network administrators	Std and protocols	Standard refs.	Information	Sensors
Data managing	Data collection and storage	Database, storage capacity, edge/cloud storage	Ethernet, Wi-Fi, 5G; OPC UA, AutomationML, MT connect	IT infrastructure team, Database administrators	Cloud data, Production data	Cloud, Production	Database	Edge/cloud
	Security and anonymity of data	Security protocol	Access levels, shareability, anonymity, firewall	Information security officers, Privacy officers	Security protocols	Security manager	Secure/safe data	Database
Monitoring and analysis	Data pre-processing	Data filtering, normalization	Data filtering: Moving average filter, outlier removal; Normalization: Min-Max scaling, Z-score normalization	Data engineers, Data analysts	Raw data	Monitoring service	Pre-processed data	Analysis service

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

	Data analysis	Fault identification, classification algorithms	Machine learning algorithms: Decision trees, Support Vector Machines (SVM)	—	Pre-processed data	Analysis service	Failure data (analysis results)	Data analysis team
	Fault detection	Algorithm to locate root cause of failure	Fault tree analysis, Root Cause Analysis (RCA)	Maintenance engineers, Reliability engineers	Failure data, KPIs for optimization	Analysis service	Failure report and fault root cause	Analysis service
	Performance analysis	Forecasting algorithms	Performance monitoring software: Grafana, Prometheus	Operations managers, data analysts	Failure data	Planning service	Prediction of failure/predictive maintenance	Planning service
Planning	Alerting and accessibility	Safety signals, secure access for operators	Failure visualization, alarm signals; Two-level authentication, security	Operations supervisors, Maintenance technicians	Failure	Analysis service	Alarms and notifications	Safety system
	Fault assessment	Manuals and maintenance knowledge	Locating fault through RCA report, analysis of state of system	Maintenance team	Failure reports	Analysis service	Fault assessment report	Maintenance team
	Policies adjustments	Knowledge, user feedback, fault analysis results	Adjusting policies based on performance metrics, user feedback, and industry standards	Policy managers, Process team	Feedback and analysis results	Analysis service, maintenance, operators	Policy adjustment proposal	Planning service
	Adaptation plan/Recovery plan	Contingency plan based on diagnostic results, policies, previous knowledge	Target software, actuators, operator intervention	Maintenance team	Diagnostic results	Analysis service	Contingency plan	Planning service

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	Plan execution	Standards, policies and safety compliant action plan	Safety measures needed to isolate root cause and actuate fault correction	Maintenance team	Recovery plan	Planning service	Fault correction possible	Maintenance team
	Fault correction	Spare parts, manuals, work permits, software patches, actuators	Replacing faulty component, restart machine, check safety, updates on software, actions on actuators	Maintenance team	Contingency plan results, asset information	Analysis service, maintenance	Executed action on asset/software/actuator, maintenance report	Maintenance team
	Reporting	Edge/cloud storage, operators	Business dashboards and HMI for failure visualization: Tableau, Power BI	Operations supervisors, data analysis team	Failure data	Analysis service	Failure report	Operators
Knowledge	Historical data	Edge/cloud storage	Historical data saved and retrieved for knowledge acquisition and updating: Database records, cloud storage	Database administrators	Production data	Production	Report	Data scientists
	Policies	Knowledge management system	Rules saved, evaluated, and updated based on events and conditions: Rule-based engine, knowledge base	Compliance officers, Policy administrators	Rules and conditions for failures and faults	Expert knowledge	Policies	Policy administrators

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

	Symptoms and conditions	Edge/cloud storage	Events and conditions bringing faults are saved: Log files, cloud storage	Operations technicians, Maintenance operators	Failure reports	Operators	List of failure causes	Operators
	Information	Runtime conditions, KPIs, data models	Runtime conditions: operating parameters; KPIs: OEE; data models: product recipe specifications	Operations supervisors, Data analysts	Info on production	Operators	Knowledge	Knowledge management system

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 Requirements

R1. Connectivity: enabling connectivity among various equipment is essential. Therefore, all manufacturing resources should be accessible and capable of communicating with other resources in the network. This entails establishing connections between data source elements (e.g., sensors), manufacturing assets (e.g., machines), services (e.g., self-X), and all network layers including edge, fog, and cloud. To achieve seamless interoperability and facilitate data exchange among these resources, it is crucial to employ open standards and protocols for connectivity in IIoT and IoT networks. At the field level, commonly utilized transport protocols include Profibus, Ethercat, and CAN. In addition, open standards such as OPC UA, AutomationML, and MT connect play a pivotal role in enabling data exchange and fostering interoperability within the IIoT and IoT. IoT applications are time-sensitive and require streaming instead of batch-processing in real-time [29], which rely heavily on wireless networking, such as WiFi, and 5G.

R2. Data managing: the manufacturing infrastructure should have the capability to collect data from the edge devices, transfer data between different layers, and store it in the storage systems. Each layer is in charge of a specific data management task:

1. The edge layer should be responsible for data acquisition from the edge devices, performing data pre-processing, and transferring the pre-processed data to the fog layer.
2. The fog layer serves as the bridge between the edge and the cloud layers and is responsible for data transfer.
3. The cloud layer should have data storage capacity and computing power.

Therefore, it is necessary to establish the required IT infrastructure. This includes defining the data acquisition frequency (i.e., sampling period), determining the required data storage capacity, defining information structure and data models (e.g., AADL, UML, MARTE), and implementing the necessary IT resources for data transfer (e.g., routers, switches, servers, gateways). Additionally, data management should give particular attention to data security and privacy by establishing the required data security protocols, data protection tools, and policies. It is important to note that while the cloud layer offers significant computational power and storage capacity, it introduces some latency to the data. Therefore, if real-time data analysis is required, the edge layer should also have some additional data storage and computing capability.

R3. Monitoring and analysis: continuous monitoring and analysis of data is essential to identify faults, breakdowns, malfunctions, and anomalies, as well as diagnose the underlying causes of failures. This encompasses a range of events, from simple changes in resource states to more complex events like value fluctuations or other complex patterns [30]. Thereby, observed data must be continuously processed in the edge layer. To begin with, raw data should be pre-processed by applying data filtering, cleaning by eliminating outliers with density clustering, normalization and scaling, and feature extraction techniques. Machine learning techniques like decision trees can be employed for feature extraction, although deep learning methods have gained attention due to their ability to automatically extract features from raw data and accurately establish nonlinear mappings of different

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

health conditions [31]. The extracted feature data should be sent to the cloud to store it as historical data to further train machine learning models, i.e. artificial neural networks, support vector machines, and random forests. Fault diagnosis is usually a classification problem, thus, pre-processed data can then be used to detect failures using fault identification or classification algorithms, such as machine learning or deep learning data-driven approaches. Fault diagnoses can also be used to determine the root cause of failure, or predict the Remaining Useful Life (RUL) of the asset with forecasting algorithms. Establishing the required KPIs is crucial in this context. The outcome of this phase should be a change request to further explore in R4 for contingency plans.

R4. Planning: self-diagnosis requires a mechanism for making reports and contingency plans. Fault assessment reports should be derived from expert knowledge and root cause analysis reports retrieved from R3. In the event of failures, contingency plans should be designed to create actions with the goal of addressing the issues, making recovery plans or providing necessary actions to the operators to mitigate the problem. The developed plan can range from a basic action like shutting down the system to more intricate tasks like altering the structure or the process model. Lastly, new knowledge obtained from the failure reports should be used to provide feedback to the internal knowledge storage systems (i.e., R6), and make any necessary adjustments to KPIs and policies.

R5. Executing: contingency plans developed in R4 should be implemented either in the target software, which might imply there termination of a task, require some correction in the actuators, or provide a clear instruction plan to the operators for execution. In addition, all failure reports need to be promptly alerted and visualised through business dashboards or HMIs to the relevant stakeholders i.e., operation supervisors and maintenance technicians. It is important to note that this information may contain confidential data, hence it is crucial to ensure secure access to the information by establishing proper security measures, such as two-level authentication.

R6. Knowledge: all data and acquired knowledge should be stored in edge/cloud storage databases, as they form the core body of the self-diagnosis service. This encompasses a wide range of information, including historical data, policies and rules, fault symptoms and conditions, runtime conditions and KPIs, among others. These databases are continuously updated through the feedback obtained from the self-diagnosis agent.

5 Conclusion

In this deliverable, we presented a guide for the implementation of self-configuration and self-diagnosis services in a cloud-based system environment. Such guide aims to be as complete and comprehensive as possible. The study and collection of requirements was conducted with such a thorough and methodical approach, that we believe to have formulated the most extended and esplicative list of requirements to date. The main goal of this work was to provide practitioners with an implementation guide to enable them to design, prepare, and implement these services in real-world applications. With an in-depth evaluation of previous research, experiential first-hand knowledge, and practical implementation, we have identified the key requirements, analyzed existing architectures, and proposed a novel one that addresses the barriers and problems associated with the deployment of self-configuration and self-diagnosis services.

The main portion of this work focused on the extraction and extrapolation of requirements from the initial list that was suggested in previous deliverables [12]. We carefully reviewed the proposed requirements, extended and focused the list to contain the essential components necessary to fulfill them. We created a framework and a guideline for practitioners to follow by mapping these requirements to the corresponding functionalities of self-configuration and self-diagnosis, as well as identifying common input-outputs relationships between the parts, and highlighting the main stakeholders responsible for each resource involved. This process ensured that the guide captures the core aspects for successful implementation, thus enabling interested companies to create the services they need.

In order to correctly deploy the services and utilize the identified requirements, we conducted an in-depth analysis of existing architectures and frameworks related to self-configuration and self-diagnosis, to identify common patterns and approaches that are employed for similar works in the field, and consequently combined such insights to propose a novel architecture that addresses the specific challenges of self-configuration and self-diagnosis services in a cloud-based system environment.

The proposed architecture takes into account the needs of the system for autonomous decision-making capabilities, emphasizing its scalability, and the adaptability to varying system conditions. The architecture consists of several key components, including an asset layer, an edge and a fog layer functioning as intermediates before the high-level cloud layer, which is the main party involved in the services' activities. All the components are linked to enable self-configuration and self-diagnosis cohesively, ensuring efficient system operation and reducing manual intervention.

Furthermore, the guide not only provides a comprehensive overview of the proposed architecture but also offers guidelines and highlights for the implementation process, by suggesting the main resources to be introduced. From the requirements to system design, to deployment and maintenance, practitioners are able to leverage the guide to analyze and reduce the difficulty involved in realizing self-configuration and self-diagnosis services. By following the recommended architecture and considering the provided insights, practitioners can improve the efficiency and robustness of their systems.

It is worth noting that the guide can be extended to encompass other services that have been presented in previous deliverables, such as self-learning and self-organization. Thanks to future work focusing on the integration of additional services, the guide will offer a comprehensive resource for implementing end-to-end autonomous functionalities in distributed systems.

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

6 Next steps and final remarks

The guide presented in this deliverable is an important milestone towards achieving autonomous and adaptive cloud-based systems. It lays the foundation for practitioners to implement self-configuration and self-diagnosis services, and serves as a road map to address critical needs for system autonomy, offering a list of common requirements and needed resources. By incorporating such insights, practitioners can improve the situation related to heavy effort required for system configuration and diagnosis, increasing system availability and improving the overall system performance.

One of the key advantages of the proposed guide is its ability to extrapolate suggestions for implementation from the list of specifications, which is as essential and complete as possible, and allows practitioners to have a clear understanding of the underlying necessities of an autonomous system, while at the same time allowing them to adapt the guide to their specific system requirements.

The next steps include providing a practical case-study implementation of the services described in this deliverable, by applying the proposed architecture and guidelines to a real-world scenario. This will serve as an evaluation of the effectiveness and practicality of the suggested approach and of as many of the list's requirements as possible. This will also help showing the benefits of self-configuration and self-diagnosis services in improving system performance and adaptability. Future works also include extending the guide to encompass other autonomous services, such as self-learning and self-organization, and integrating them into a comprehensive resource for implementing end-to-end autonomous functionalities in cloud-based systems.

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

References

[1] Zhengyi Song and Young Moon. “Performance analysis of CyberManufacturing Systems”. In: *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 233.5 (Apr. 2019), pp. 1362–1376. DOI: 10.1177/0954405417706996.

[2] Oliver Niggemann et al. “Data-Driven Monitoring of Cyber-Physical Systems Leveraging on Big Data and the Internet-of-Things for Diagnosis and Control”. In: *Proceedings of the 26th International Workshop on Principles of Diagnosis*. 26th International Workshop on Principles of Diagnosis. DX. Paris, FR, 2015.

[3] Jay Lee, Chao Jin, and Behrad Bagheri. “Cyber physical systems for predictive production systems”. In: *Production Engineering* 11.2 (Apr. 2017), pp. 155–165. DOI: 10.1007/s11740-017-0729-4.

[4] Jay Lee, Behrad Bagheri, and Hung-An Kao. “A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems”. In: *Manufacturing Letters* 3 (Jan. 2015), pp. 18–23. DOI: 10.1016/j.mfglet.2014.12.001.

[5] M. G. Mehrabi, A. G. Ulsoy, and Y. Koren. “Reconfigurable manufacturing systems: Key to future manufacturing”. In: *Journal of Intelligent Manufacturing* 11.4 (2000), pp. 403–419. DOI: 10.1023/A:1008930403506.

[6] Hung-An Kao et al. “A Cyber Physical Interface for Automation Systems—Methodology and Examples”. In: *Machines* 3.2 (May 14, 2015), pp. 93–106. DOI: 10.3390/machines3020093.

[7] Daqiang Guo et al. “A roadmap for Assembly 4.0: self-configuration of fixed-position assembly islands under Graduation Intelligent Manufacturing System”. In: *International Journal of Production Research* 58.15 (Aug. 2, 2020), pp. 4631–4646. DOI: 10.1080/00207543.2020.1762944.

[8] A.W. Colombo et al. “Service-oriented architectures for collaborative automation”. In: *31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005.* 31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005. Raleigh, NC, USA: IEEE, 2005, 6 pp. DOI: 10.1109/IECON.2005.1569325.

[9] Sergii Iarovyi et al. “From artificial cognitive systems and open architectures to cognitive manufacturing systems”. In: *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*. 2015 IEEE 13th International Conference on Industrial Informatics (INDIN). Cambridge, United Kingdom: IEEE, July 2015, pp. 1225–1232. DOI: 10.1109/INDIN.2015.7281910.

[10] José Antonio Mulet Alberola et al. *DiManD Work Package 3.1a – Overall requirements and preliminary selection*. European Project Deliverable 3.1a. Stockholm, Sweden: KTH Royal Institute of Technology, Mar. 2021, p. 27.

[11] José Antonio Mulet Alberola et al. *DiManD Work Package 3.1b – Full analysis and final set of requirements*. European Project Deliverable 3.1b. Stockholm, Sweden: KTH Royal Institute of Technology, Sept. 2021, p. 51.

[12] Luis Alberto Estrada-Jimenez et al. *DiManD Work Package 3.2 – Cyber Physical Systems Architecture*. European Project Deliverable 3.2. Stockholm, Sweden: KTH Royal Institute of Technology, June 2022, p. 36.

[13] Armando W. Colombo et al. "Industrial Cyberphysical Systems: A Backbone of the Fourth Industrial Revolution". In: *IEEE Industrial Electronics Magazine* 11.1 (Mar. 2017), pp. 6–16. DOI: 10.1109/MIE.2017.2648857.

[14] Borja Ramis Ferrer et al. "Towards the Adoption of Cyber-Physical Systems of Systems Paradigm in Smart Manufacturing Environments". In: *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*. 2018 IEEE 16th International Conference on Industrial Informatics (INDIN). Porto: IEEE, July 2018, pp. 792–799. DOI: 10.1109/INDIN.2018.8472061.

[15] Carolina Villarreal Lozano and Kavin Kathiresh Vijayan. "Literature review on Cyber Physical Systems Design". In: *Procedia Manufacturing*. 10th Conference on Learning Factories. Vol. 45. CLF. Graz, AT: Elsevier, 2020, pp. 295–300. DOI: 10.1016/j.promfg.2020.04.020.

[16] Rima Al-Ali et al. "A guide to design uncertainty-aware self-adaptive components in Cyber–Physical Systems". In: *Future Generation Computer Systems* 128 (Mar. 2022), pp. 466–489. DOI: 10.1016/j.future.2021.10.027.

[17] Fan Mo et al. "A maturity model for the autonomy of manufacturing systems". In: *The International Journal of Advanced Manufacturing Technology* (Feb. 27, 2023). DOI: 10.1007/s00170-023-10910-7.

[18] Qinglin Qi and Fei Tao. "A Smart Manufacturing Service System Based on Edge Computing, Fog Computing, and Cloud Computing". In: *IEEE Access* 7 (2019), pp. 86769–86777. DOI: 10.1109/ACCESS.2019.2923610.

[19] Alessandra Caggiano. "Cloud-based manufacturing process monitoring for smart diagnosis services". In: *International Journal of Computer Integrated Manufacturing* 31.7 (July 3, 2018), pp. 612–623. DOI: 10.1080/0951192X.2018.1425552.

[20] Dazhong Wu et al. "Fog-Enabled Architecture for Data-Driven Cyber-Manufacturing Systems". In: *Volume 2: Materials; Biomanufacturing; Properties, Applications and Systems; Sustainable Manufacturing*. ASME 2016 11th International Manufacturing Science and Engineering Conference. Blacksburg, Virginia, USA: American Society of Mechanical Engineers, June 27, 2016, V002T04A032. DOI: 10.1115/MSEC2016-8559.

[21] Karolj Skala et al. "Scalable Distributed Computing Hierarchy: Cloud, Fog and Dew Computing". In: *Open Journal of Cloud Computing (OJCC)* 2 (2015), pp. 16–24. DOI: 10.19210/1002.2.1.16.

[22] R. Frei and Giovanna Di Marzo Serugendo. "Concepts in complexity engineering". In: *International Journal of Bio-Inspired Computation* 3.2 (2011), p. 123. DOI: 10.1504/IJBIC.2011.039911.

[23] Regina Frei et al. "Self-healing and self-repairing technologies". In: *The International Journal of Advanced Manufacturing Technology* 69.5 (Nov. 2013), pp. 1033–1061. DOI: 10.1007/s00170-013-5070-2.

[24] Harald Psaier and Schahram Dustdar. "A survey on self-healing systems: approaches and systems". In: *Computing* 91.1 (Jan. 2011), pp. 43–73. DOI: 10.1007/s00607-010-0107-y.

[25] *Information technology — Cloud computing — Part 1: Vocabulary*. Technical ISO/IEC 22123-1:2023. Geneva, CH: International Organization for Standardization, Feb. 2023, p. 18.

[26] Charith Perera et al. "Context Aware Computing for The Internet of Things: A Survey". In: *IEEE Communications Surveys & Tutorials* 16.1 (2014). Conference Name: IEEE Communications Surveys & Tutorials, pp. 414–454. DOI: 10.1109/SURV.2013.042313.00197.

[27] Yazhe Wang, Shunan Ma, and Lei Ren. "A Security Framework for Cloud Manufacturing". In: ASME 2014 International Manufacturing Science and Engineering Conference collocated with the JSME 2014 International Conference on Materials and Processing and the 42nd North American Manufacturing Research Conference. American Society of Mechanical Engineers Digital Collection, Oct. 3, 2014. DOI: 10.1115/MSEC2014-4082.

[28] N. Stricker and G. Lanza. "The Concept of Robustness in Production Systems and its Correlation to Disturbances". In: *Procedia CIRP* 19 (2014), pp. 87–92. DOI: 10.1016/j.procir.2014.04.078.

[29] Montdher Alabadi, Adib Habbal, and Xian Wei. "Industrial Internet of Things: Requirements, Architecture, Challenges, and Future Research Directions". In: *IEEE Access* 10 (2022), pp. 66374–66400. DOI: 10.1109/ACCESS.2022.3185049.

[30] Lukas Malburg, Maximilian Hoffmann, and Ralph Bergmann. "Applying MAPE-K control loops for adaptive workflow management in smart factories". In: *Journal of Intelligent Information Systems* (Jan. 25, 2023). DOI: 10.1007/s10844-022-00766-w.

[31] Min Xia et al. "Intelligent fault diagnosis of machinery using digital twin-assisted deep transfer learning". In: *Reliability Engineering & System Safety* 215 (Nov. 2021), p. 107938. DOI: 10.1016/j.ress.2021.107938.

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