The Digital Manufacturing and Design Training Network

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

Deliverable 4.2

Development of a Data Model for Proactive Intelligent Products

Lead parties for Deliverable: UNOTT
Deliverable due date: April 2022
Actual submission date: May 2022
Dissemination level: Public

Contents

1 Introduction and Context

Modern manufacturing in western economies such as the European Union are increasingly reliant on the design and manufacturing of high-complexity, high-value products, including electronic and computing devices [1]. Even where a product might not be considered a computing device, an increasing proportion of products have electronics and computers embedded within them [2]. So-called *smart devices* range from consumer goods such as smartphones and electronic toothbrushes, wearable devices such as smart watches and shoes which monitor the wearer's activity, smart home devices such as home assistants and smart thermostats, and larger products such as modern cars and aircraft. Such smart devices are characterised by their ability to (i) sense the environment in which they are operating, (ii) use actuators to influence and change the environment around them, (iii) process information from their sensors to determine how to utilise their actuators, and optionally (iv) interfaces that allow them to communicate with other devices and systems [2].

Although electronics manufacturing in a general sense is reliant on volume production with long pre-production development and low customisation opportunity [3], smart device manufacturing can be more responsive by using the electronics in dynamic and variable ways and by customising the non-electronic aspects of the products. Increasingly, the manufacture of smart products is as reliant on the software and configuration of the devices, as it is on the hardware and physical aspects of the device. For example, an Amazon Echo home assistant that is purchased via Amazon, will arrive pre-configured with the purchaser's Amazon account and Wi-Fi credentials.

In general, the complexity of manufactured products – and therefore the systems required to manufacture them – is increasing [4]. For smart products, as the complexity of such hybrid software/hardware manufacturing processes increases, there is a need to manage this complexity to prevent errors and maintain profitability. Traditional inspection processes such as visual inspection or measurement cannot cope with errors in software installation, and conventional integrated circuit testing methods such as probe testing and functional testing [5] are less able to catch manufacturing customisation errors or complex software issues. There may also be concerns over user data protection and privacy (such as the use of login credentials) and protection of intellectual property (IP) when IP owners vary between the hardware and software components of a product.

Decentralised manufacturing control holds promise for complexity management in modern manufacturing systems. Approaches such as holonic manufacturing control paradigms [6] and intelligent agent control [7] tackle complexity by breaking the problem into smaller localised problems which are managed by individually simple control processes. The holistic control process emerges from the interactions of simple components into a broader coordinated system.

Conventional manufacturing control processes control the manufacturing machinery to produce products - the products are passive materials to which manufacturing occurs. However, the intelligence of smart products presents an opportunity for products to become active participants in the manufacturing process. Holonic manufacturing paradigms such as PROSA [8] include Products and Resources (the P and R in PROSA respectively) as active participants in manufacturing control, and production is achieved through the collaboration of these entities, along with Orders and Staff.

Decentralised approaches such as PROSA separate the concerns of the *requirements* of a product to be manufactured, from the *capabilities* of the manufacturing system that may produce it. In figure 1 these are equivalent to the production knowledge (for the requirements) and the process knowledge (for the capabilities). [1] This allows for each concern to be considered in isolation, where the design of the product and the available resources are considered independently.

This separation of the requirements from the capabilities is a trend contrary to the common practice of *design for manufacturing* - where the design and method of manufacturing a product

---

[1] A discussion of the process execution knowledge will be covered in the following deliverable 4.3

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

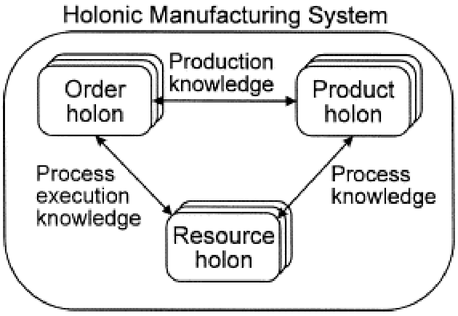


Figure 1: Key elements of the PROSA architecture, and the knowledge exchange between them. Taken from [8]

are interlinked and considered in tandem [9]. This remains an essential practice where a product is designed and manufactured by the same company, or the design and manufacturing companies are distinct entities but have an established relationship and are open with data exchange. This is also a critical approach for large volume manufacturing, where small changes to a product's design could save significant time and money over a long production life-cycle.

However, there is an increasing push for new business models, and new dynamic approaches to manufacturing control for low batch sizes or batch-size-of-one mass customisation. Manufacturing-as-a-service – sometimes called cloud manufacturing [10], is a trend where companies offer their manufacturing capability on-demand to other companies. This enables designers to create products without owning a manufacturing facility themselves, for companies to use third-party capacity for demand balancing, or for a company to temporarily use a production resource without investing in the machine themselves. This is currently most common for additive manufacturing [11] but is expanding to full product life cycles, including design-, manufacturing-, testing-, and management-as-a-service [12].

This new approach offers opportunities for products to take a proactive interest in their own manufacturing. However, a product does not need to be 'smart' to achieve this. Even a product without integrated computational power can have a representation in a distributed manufacturing system. Thus, we distinguish a 'smart product' (i.e. one with computing power integrated) from an 'intelligent product', which features the following properties (taken from [13]):

1. Possesses a unique identification.
2. Is capable of communicating effectively with its environment.
3. Can retain or store data about itself.
4. Deploys a language to display its features, production requirements, etc.
5. Is capable of participating in or making decisions relevant to its own destiny.

Note that it is not required that than an intelligent product and the intelligence that enables properties 2-4 be co-located. Indeed [13] states that provided there is an identifier link (i.e. property 1) that can be read by the production system, then the other properties can be achieved with an agent or holon "either locally or on the network". This means it is not required that an intelligent product be also a smart product.

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

1.1 Objectives of this Deliverable

This deliverable focuses on properties 4 and 5 above – the ability for the product to display its features and production requirements, and for it to be capable of participating in and making decisions relevant to its own destiny i.e. to be 'proactive'. This deliverable describes two data models:

1. The runtime conditions of manufacturing assets being offered as a service – corresponding to the capabilities or process knowledge described earlier, but adding a time-sensitive dimension to it.
2. A process model for manufacturing processes – corresponding to the requirements or production knowledge described earlier, enabling an intelligent product to represent its own production requirements.

Together, these models enable the intelligent product to represent its own production requirements, and gives it the contextual information about a set of manufacturing assets to produce a production plan or bill of process. To achieve this, it will need to match its requirements to the capabilities available. This matching process will be discussed in the following deliverable 4.3, this deliverable only concerns the data models.

Team Involved in Deliverable Writing

ESR1: Fan Mo, The University of Nottingham
ESR2: Agajan Torayev, The University of Nottingham
ESR3: Angela Carrera, Mondragon Unibertsitatea
ESR5: Miriam Ugarte Querejeta, Mondragon Unibertsitatea
ESR7: Jose A. Mulet, STIIMA-CNR
ESR8: Fabio Marco Monetti, KTH Royal Institute of Technology
ESR9: Sylvia Nathaly Rea Minango, KTH Royal Institute of Technology
ESR12: Muhammad Bilal Khan, TECHNALIA
ESR14: Hamood Ur Rehman, The University of Nottingham and TQC Ltd
Supervisor: Dr Jack C Chaplin, The University of Nottingham

2 Runtime Conditions

The objective of this section is the definition of a suitable data model for representing the runtime condition monitoring of manufacturing assets, through the selection and evaluation of the main requirements of such a model followed by the creation of the model itself. For that purpose, this section firstly explains the concept of runtime condition monitoring in the manufacturing context, secondly presents a set of requirements that enables intelligent manufacturing, then reviews existing literature and works that approach the formulation of the most suitable data model. This section closes with a discussion of the main requirements and characteristics that specify the required model.

2.1 Definition of Runtime Condition

The increasing need for rapid and responsive manufacturing across industries such as aerospace and defense, automotive, and healthcare requires improved monitoring systems for manufacturing assets. The rising demand for energy-efficient products will also create growth opportunities. However, the high cost of maintenance, capital investments, and maintenance costs of industrial equipment are restraining the market's growth. Since runtime condition monitoring enables rapid and responsive manufacturing, there is a need for a runtime condition monitoring system of manufacturing equipment. Condition monitoring supports a variety of goals for manufacturers, including:

- Reducing and eliminating unplanned machine downtime
- Optimizing machine health and performance
- Improving quality and reducing scrap parts
- Driving a higher performing maintenance program based on accurate machine data
- Enabling automation based on real-time machine condition data

Hence, this section analyzes the term ”runtime condition monitoring” in manufacturing provided in the literature to define it.

The term “runtime” takes its root from computer science, where it describes the period of time during which a program is running. In the manufacturing domain, it refers to the period of time in which a piece of manufacturing equipment is functional. According to the Overall Equipment Effectiveness (OEE) glossary [14], runtime means the manufacturing process is scheduled for production and is running. The runtime is calculated by subtracting downtime from planned production time (see Fig. 2). The runtime includes when the process could be experiencing small stops, reduced speed, and making reject parts.

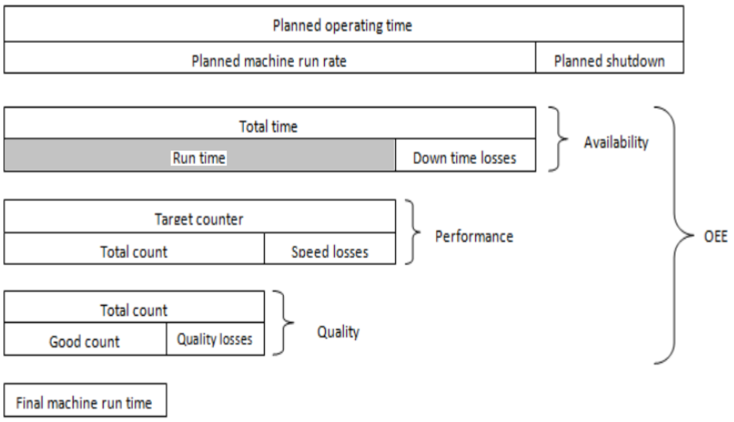


Figure 2: Main components of OEE 1 [15]

There is a variety of reasons for the installation of a monitoring system in a manufacturing process. Modern manufacturing equipment must be flexible, sustainable, and operative with minimum human interface, while machine tools must operate free of errors [16]. Monitoring is becoming a driver for the development and sustainability of manufacturing industries nowadays. Monitoring is the collection of sensor measurements (e.g., force, vision, temperature) to determine the manufacturing asset's state. Condition monitoring is the technology that provides runtime information for optimization [17]. In [17], authors propose the following monitoring of manufacturing assets:

- Energy consumption monitoring
- Tool wear monitoring
- Production state monitoring
- Inventory monitoring

In [18] authors define a cyber-physical system (CPS)'s monitoring as a lightweight verification technique that provides a practical way to monitor and verify CPSs at runtime. In [19], the authors introduced a generic condition monitoring system, which is the enabler for predictive maintenance and online product quality monitoring in this area. In [20], authors consider the use of a multiagent system incorporating hidden Markov models for the condition monitoring of gas turbine (GT) engines. Hidden Markov models utilizing a Gaussian probability distribution are proposed as an anomaly detection tool for GTs components. The use of this technique is shown to allow the modeling of the dynamics of GTs despite a lack of high-frequency data.

In order to monitor the condition of operation and predict a failure in the machine elements, many monitoring techniques have been developed and widely used. In the literature, there are generally two types of condition monitoring techniques:

1. Model-based condition monitoring
2. Data-driven condition monitoring

Model-driven condition monitoring techniques rely on either mathematical models of the manufacturing system or semantic models of the manufacturing systems. For the model-based condition monitoring techniques, ontology-based methods are popular among researchers. Model-based methods have good qualities, such as the explainability and interpretability of models; however, these models require expert knowledge to model the system.

Data-driven approaches rely on manufacturing data and machine learning algorithms. The benefit of employing data-driven condition monitoring methods is the abundance of manufacturing data and the effectiveness of off-the-shelf intelligence algorithms. However, data-driven methods often lack the interpretability required in the manufacturing domain.

Based on the literature, runtime condition monitoring requires the following characteristics:

- **Heterogeneity and mobility**: a model should be able to express different types of information, and provide management of the information depending on its type. For example, a model should be robust to image, text, time-series, and other manufacturing data types.
- **Relationship and dependencies**: various relationships between different types of information have to be captured to ensure the correct behavior of applications. For example, the relationship between image data captured and the time-series data representing the status of equipment should be aligned.
- **Timeliness (context histories)**: the past and future states should be captured by models and managed by management systems. For example, the old OEE and the possible future OEE should be captured.
- **Imperfection**: ability to model information quality (incorrect, incomplete, conflicting context information) to support reasoning about context. The data in manufacturing is very noisy because of unexpected faults or stops. Hence, the model should be robust to these kinds of imperfections.
- **Reasoning**: ability to support consistency verification and reasoning techniques to derive new context facts from existing facts and/or reason about high-level abstractions of real-world situations. Reasoning techniques should be computationally efficient. For example, a model should reflect that another machine uses certain manufacturing equipment by analyzing semantic relationships.
- **Usability of modelling formalisms**: the ease of translating real-world concepts to the modeling constructs, the ease of usage, and context manipulation by context-aware applications.
- **Efficient provisioning**: efficient access to the information in the presence of large models and numerous data objects. The data in manufacturing is generated in huge amounts. A model should be able to distill the information for further decision-making.

Based on the literature review, we define runtime condition monitoring as:

The process of capturing the internal condition or status of any operational assets - such as products being produced and manufacturing equipment -, in order to guard the correct execution of manufacturing processes or improve manufacturing operations by the identification and prediction of behaviour changes, property violations or malfunctions.

2.2 Requirements for runtime condition data model

Smart and autonomous manufacturing systems are described by a set of *self-x* behaviours, namely self-organization, self-configuration, self-learning, self-diagnosis. In order to allow these autonomous behaviours, modularity, interoperability, scalability and robustness are key requirements [21]. Hence, these main features may be extrapolated to characterise the runtime data model needed to implement smart manufacturing operations.

The concept of *modularity* defines singular manufacturing assets, then sets clearly and unambiguously a representation of any asset in a logical structure. High modularity relates to the concept of high granularity, which defines the level of abstraction detail. Smart manufacturing systems are composed of a set of interconnected individual components that perform independently but in collaboration to reach common production goals.

In order to create this network of resources that work in collaboration, the *integration* of multiple components must be guaranteed. Multiple tools, components, and sensors must use similar communication protocols to enable their integration, thus allowing to work together.

A system is *scalable* when allows its extension without restrictions, incorporating new parts and subsystems -that were not initially conceived during the first design of the system- or removing them, without huge engineering or programming effort, thus improving efficiency.

Robustness may be defined as the ability to remain operative on a stable and high performance level despite the given risks. Therefore, robustness provides systems the ability to work within acceptable tolerance values (KPI's) assuring the system's reliable response under instabilities; i.e disturbances, changes or conflicting policies. In addition, authentication, selective disclosure, and revocation of permissions under privacy and cybersecurity policies may be also needed to ensure secure data systems and total privacy of data.

2.3 Existing manufacturing standards

This section reviews related literature on runtime data models. The main contributions have been focused on i) current developments in data models for runtime representation; ii) existing deployments of data models in industrial scenarios using the Asset Administration Shell; and iii) related standards that cover digitalization of assets and selection of performance indicators in manufacturing contexts.

An information model allows the abstraction of the specifications of concern of a physical device. Standards contribute in providing information models for describing several physical objects for manufacturing purposes. Figure 3 presents a timeline based representation of standards currently targeting to describe machine tools conditions in the manufacturing processes [22]. Each of the standards will be briefly discussed below.

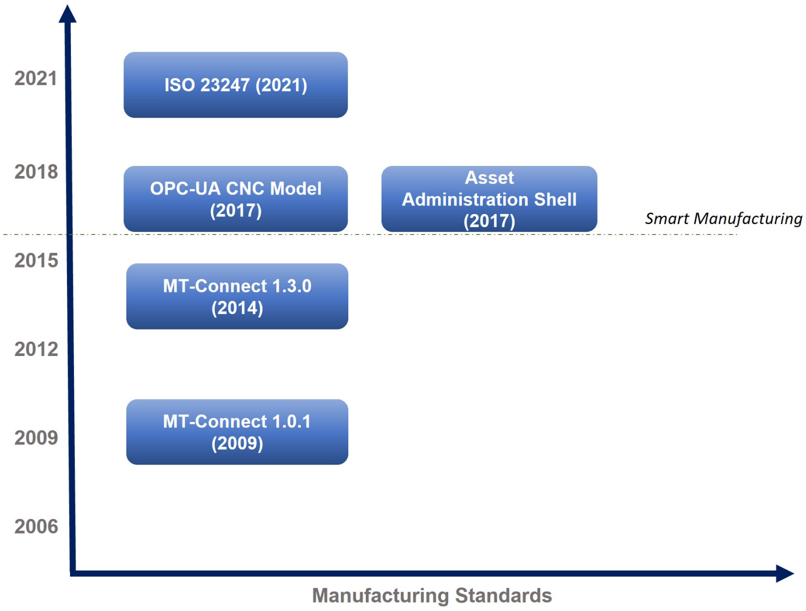


Figure 3: Timeline-based depiction of standards for manufacturing stage. (Based on Lu et al. [22])

2.3.1 MT Connect

MTConnect is an open-source, data and information exchange standard for describing manufacturing operations.The standard was designed to enhance the data-acquisition capabilities of manufacturing equipment, systems, and software applications and enable a plug-and-play environment[23]. MT-Connect has been extensively used to represent static properties as well as to monitor machine tools status [24], especially for CNC machines. Other types of data that can be obtained includes on-machining measuring data (OMM), sensor data (i.e vibration, temperature) etc. [25]

The semantic data models included in MTConnect Standard allows a clear and unambiguous characterization of the information. Furthermore, it also allows to establish the relationship of the data of the manufacturing operations and their sources of origin. Currently, four semantic data models are included in MTConnect: devices, streams, assets, and errors. [26].

- *Devices Information Model*
  Provides a hierarchical representation of equipment metadata that includes the logical and physical 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 [27].

- *Stream Information Model*
  Provides a representation of the values reported by a piece of equipment used for a manufacturing process, or for any other purpose. Each stream can report an EVENT, SAMPLE or CONDITION defined previously as data entities on the Device Information Model[28].

- *Asset Information Model*
  Defines entities that are used in the manufacturing process, which are not considered a device nor a component of a device, but they can be related with different devices during their lifecycle. Two types of assets are addressed, CuttingTool and CuttingToolArchetype [29].

- *Error Information Model*
  Establishes the terminology when errors occurs in requests of information from client software. [27]

However, MTConnect has its limitations and the data models will need to evolve as machinery and equipment present new advances and they are upgraded with new elements. Therefore extensions of MTConnect models are presented in the literature. Lei et al.[25] extended MTConnect data models for touch-trigger probe data used in on-machine measurement tasks and sensor-based intelligent fixturing related information. Edrington et al. [30] presented a monitoring web application for MTConnect compatible machines. The application provides shop-floor personnel with production information to improve overall equipment effectiveness (OEE).

2.3.2 OPC/Unified Architecture - OPC/UA for CNC Model

OPC/UA is an open and extensible set of standards and interoperability framework based on internet technologies like TCP/IP, HTTP, Web Sockets. The main goal is the online data exchange between devices and HMI or SCADA systems, providing both communication protocol and information modelling method. Therefore, it can be easily used to model digital twins of manufacturing facilities [24]. The data is represented in an object-oriented structure and using XML as a serialization language. The multilayered information model is composed by the information model building blocks and the core information model (e.g analog data, alarms, state machine) [31]. This is the base for the development of domain-specific models, that in contrast with MT-Connect is a more generic method that allows to cover a broader range of equipment and systems [24].

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. Objects are used to represent structuring elements of the CNC data interface such as *CNCAxisListType* or *CNCSpindleListType*. In Figure 4 we present an example of the representation of a machine with 2 axis, according to the information model. Each list type define a grouping element of hardware and software components of a CNC system. For instance CNCAxisList is composed by CNCAxisTypeX and CNCAxisTypeY and each of these individual components can have several properties and data associated (i.e. Direct Position, Indirect Position, etc.).

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

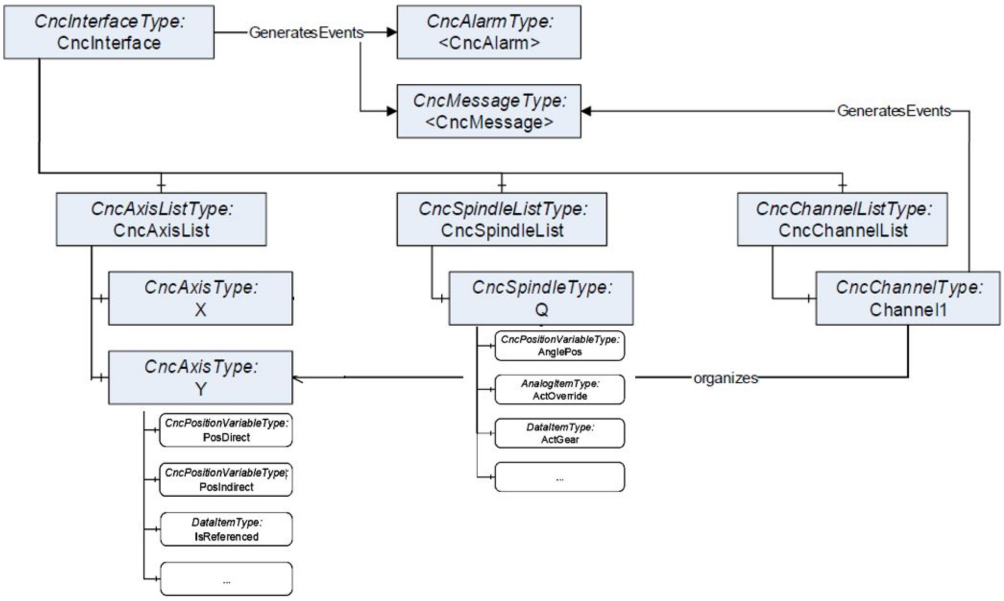
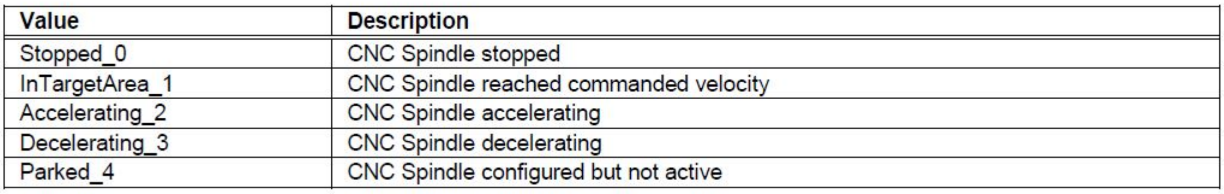


Figure 4: OPC/UA for CNC representation of a Machine with 2 axis (Adapted and according to model [31])

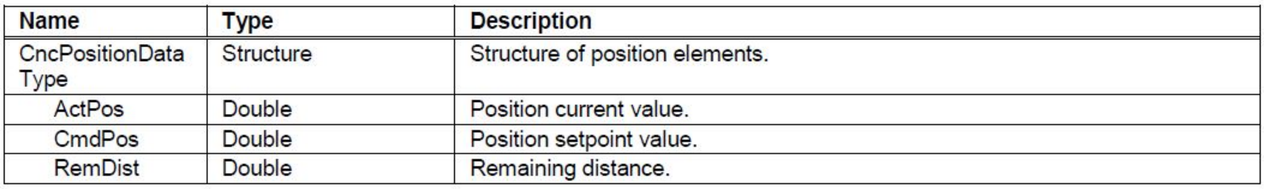
The data can be divided into five categories[31]:
Parameters: configuration data from the CNC system.
State data: current status of the hardware within CNC, that can be defined by the multiple provided enumerations of the information model. Figure 5 presents an example of the CNC Spindle status enumeration.



Value	Description
Stopped_0	CNC Spindle stopped
InTargetArea_1	CNC Spindle reached commanded velocity
Accelerating_2	CNC Spindle accelerating
Decelerating_3	CNC Spindle decelerating
Parked_4	CNC Spindle configured but not active

Figure 5: CncSpindleStatus Enumeration Values (From [31])

Command & Process Data: command data refers to the setpoint values and commands that are communicated from the Machine Interface to the CNC system, process data are the actual values of the hardware devices. Figure 6 presents an example of *CncPositionDataType* used to represent current position and setpoint position within CNC.



Name	Type	Description
CncPositionData Type	Structure	Structure of position elements.
ActPos	Double	Position current value.
CmdPos	Double	Position setpoint value.
RemDist	Double	Remaining distance.

Figure 6: CncPositionDataType Structure(From [31])

Alarms: Events of the CNC System, e.g. by the CNC kernel, the PLC, but as well by the UI, informing about errors or other notifications.

2.3.3 Asset Administration Shell

The concept of the AAS was created and specified by Platform Industrie 4.0 in order to enable cross-company interoperability [32]. Thus, AAS conforms a standardised communication interface that connects an asset into industrie 4.0, enabling interoperability across the network [32, 33]. Figure 7 describes the general structure of the AAS:

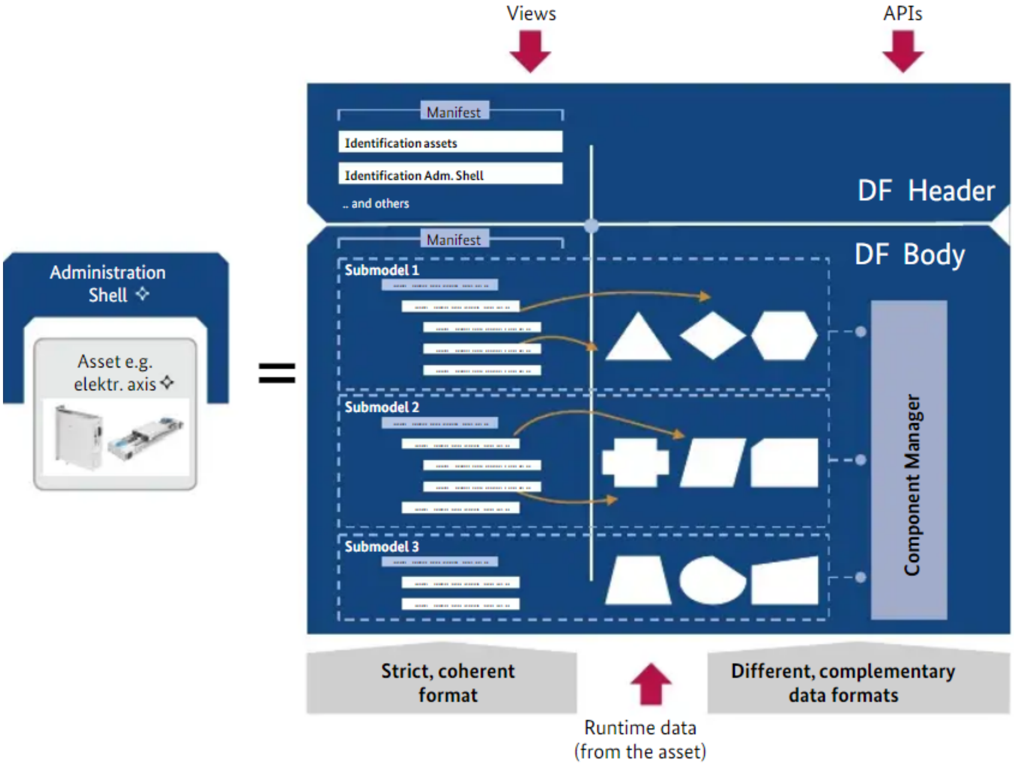


Figure 7: General strucutre of the Asset Administration Shell ([34])

The asset linked to the AAS can be a physical or logical object [33] and it has a unique identifier.

The AAS has also a unique identifier and it describes the properties and capabilities with a semantic description. These properties are stored in submodels, which are intended to contain any data of the life cycle [35], or any function that represent different application aspects of an asset [34].

The overall structure of the AAS is composed by a manifest and a component manager. The manifest contains the meta-information of all data, functionalities and identification of assets and it is externally accessible [34]. It consists of a header and a body. The header contains general information about the identification of an asset, whereas the body contains information about the asset itself, detailing its properties and submodels. On the other hand, the component manager is responsible for maintaining and self-managing resources (i.e. submodels) of an I4.0 component and providing external access to the resources. Component manager services are accessible through an I4.0 compliant service-oriented architecture (SoA) via Application Programming Interfaces (APIs).

There are different types of AAS based on its active/proactive capability. As such, Eclipse BaSyX [36] defines three types of Asset Administration Shells:

- Type 1 AAss are serialized files, e.g. XML, or JSON files. Serialized Asset Administration Shells contain static information and may be distributed as files.
- Type 2 AASs exist as runtime instances. They are hosted on servers, and may contain both static, i.e. less frequently changing information, but may also interact with other components.
- Type 3 AASs extend type 2 Asset Administration Shells. They additionally implement an active behavior, i.e. they can start to communicate and to negotiate on their own.

Table 1 sets out the existing implementations of the Asset Administration Shell and classifies them into different types of AASs according to its capabilities:

Table 1: Existing implementations

Name	DOI	Type	Description	Communication Protocol
NOVAAS [37]	[38]	2	API microservice based architecture. The implementation is based on docker images and it is build up on the framework proposed by the platform for Industrie 4.0. The connection with the physical entity is bidirectional to allow both monitoring and control data flows. It does not support System of Systems AAS.	REST, MQTT
PyI40AAS [39]		1	This project hosts a Python module for manipulating and validating AAS that complies with the meta model and interface specification provided in the ”details of administration shell v2.0.1	XML, JSON
Eclipse BaSyx [36, 40]		1, 2	Open-source middleware (SDK) that encapsulates an interface and communication with end-user APIs and edge devices by JSON serialization, and directory/discovery services. Available languages: Java, C# and C++.	OPC UA, MQTT, HTTP/REST
SAP I4.0 AAS [41]		1	Provides a system based on Docker images implementing the RAMI 4.0 reference architecture (including AAS).	HTTP, REST

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

SAP I4.0 AAS type2 [42]		2	same as SAP I4.0 but includes the implementation of type 2 AAS.	HTTP, REST
openAAS [43]		1, 2	Development Repository for Asset Administration Shell that uses a model based runtime environment ACPLT/RTE as well as OPC UA. It is based on the specifications of Platform Industrie 4.0	OPC UA, JSON
ThingWorx Asset Management [44]		1, 2	It provides the ability to monitor assets and their alerts. The application provides built-in user management capabilities.	
AASX Server [45]		1	AASX Package Explorer is a C# based viewer / editor for the Asset Administration Shell.	REST, MQTT, OPC UA
AAS-based-PnP-Demo [46]	[47]	3	AASs implementention for plug and produce field devices. AutomationML and OPC UA is used which adhere to the I4.0 conventions and comply with the RAMI 4.0 mode	AutomationML, OPC UA, HTTP
CoreAAS [48]	[49], [50], [51]	1, 2	OPC UA implementation of the Asset Administration Shell (AAS). It is an extension of the OPC UA SDK for Node.js. The authors have also developed the AAS implementation for IEC 61131-3 programs.	OPC UA, REST

The communication protocol is set out based on the runtime capabilities, i.e. OPC UA/ MQTT for sharing runtime data, and XML/JSON for sharing static data. As outlined in the table, most of the existing implementations have been implemented for exchanging static (type 1) and/or dynamic (type 2) data among the assets but they lack of an active behaviour. To date, type 3 proactive behaviour implementations in the manufacturing industry are very scarce. For instance, [46] is the only solution who has implemented an AAS for dynamically proactive plug and produce devices.

2.3.4 ISO 23247

The realisation of the production system in the digital domain is carried out by recording static and dynamic information. ISO 23247 is a standard that represents the production system specification. Static part of the data is concerned with those manufacturing properties that do not change during operation. If the properties change then the data is dynamic. The representation of the manufacturing system essentially formulates the digital twin of the system and assists in distributed control.

The standard assists in achieving the purpose of gathering information pertaining to production system. Relate it to categories and link the aspects of categories to drive and influence the performance. The digital twin can be used to simulate scenarios and assist in decision making. Effective classification into categories helps to further classify as per operation specific, application specific, communication specific and resource specific domains. Individually these domains can be controlled to provide a holistic view of the production system and a certain degree of modularity.

The standard makes categories of attributes of production system and assigns properties to these categories. Some of these are mandatory and other optional. At this instance these are; identifier, characteristics, status, location, report and relationship. A more detailed description can be viewed in the following figure (to be attached).

Mostly this standard targets discrete manufacturing domain. However, this standard is made modular where a different standard can be used for modelling as per need for instance it can be combined with ISO 15926 for oil and gas, ISO 16739 for building and construction applications.

2.3.5 ISO 16300

ISO standards from series 16300 addresses the user and supplier requirements for manufacturing applications with the purpose of promoting interoperability in the area of industrial automation. This series of standards integrates components in an automation system by combining capabilities. It provides an interface to integrate capabilities among resource platforms. It also provides an assessment mechanism for validating and verifying capabilities among manufacturing components.

The standard series of capabilities can assist in representing a set of capabilities provided by components, along with verifying the specific component capability among the set. It elaborates on the dependencies, access and matching capabilities of the components in the production system.

This series of standards specifies a framework for describing a production system in terms of its capabilities. It details a template for describing capability and mapping rules. These series of standards specify a framework for verifying interoperability in terms of capability of manufacturing components along with a search methodology for determining candidate capability manufacturing components.

In this set of standards (ISO 16300) manufacturing systems are considered to be a set of manufacturing applications indicated by a representative set of manufacturing activities and its associated resources. These manufacturing activities are performed through a set of processes that execute their functionality as per control and interoperability requirements. These executions, that can be considered as functions, follow an established time scheduled usually centralised in nature. The functions are triggered under specific events happening on the shop floor. The manufacturing resources implement these manufacturing functions, which can be configured to meet the interoperability requirements for instance for devices, sensors, actuators, valves etc.

The capability to enforce functionality of the manufacturing production systems is embedded in manufacturing functions of the process, housed in manufacturing applications. These processes in a production system interact with each through their functions under a common set of criteria and conditions. These standards lay the groundwork for capturing these conditions, requirements, and criteria for functionality execution and interoperability. They encapsulate information about the manufacturing application global structure that will govern the activities and process, the manufacturing resource capability, the interdependencies and the information about the configuration and deployment.

In these set of standards, the manufacturing applications rely on a centralised planning and scheduling structure to execute manufacturing functionality. The capabilities and interoperability representation in these standards are made to suit this requirement. There exists a lack of guidance for capturing information for addressing needs of decentralised production systems.

2.3.6 ISO 10303

ISO 10303 provides an approach to capture related information for data interchange. It covers a wide variety of product types like process plants and automation solutions. The standard follows the neutral-file approach, the transfer of information between the systems is a two stage process. Constraints are transmitted as they are captured by this methodology.

It enables effective interchange of information between systems by making a separation between the information model and its physical implementation. The constraint entity can be expressed in schema defined in the standard. They are represented in schemas that can be verified for semantic validity and syntactical correctness.

At the initial stage, the data is translated from native format of the system to neutral format and then into native format of receiving system. The exchanging format that promotes interoperability is ASCII file.

The existing gap in the standard is too much generalisation and lack of tracking and integrating of requirements as they are generated for distributed manufacturing scenarios.

2.4 Standards for capturing Key Performance Indicators (KPI)

Value creation can be improved by utilising Key Performance indicators (KPIs) for the management of manufacturing activities. Capturing performance and relating it to KPIs enables manufacturing industries to quantify the benefits of activities. Some standards have been proposed that capture the performance measures for such objectives of performance improvement. ISO 22400 achieves this by combining measurements from processes into KPIs. The formulated KPIs can be used to measure against set performance objectives, view relative trends and extract useful information. The standard calculates the KPIs through data collected from production control and processing it to provide a general means of decision support for managing production operations.

ISO 22400-1 comprises of an overview and basic concepts of KPIs, proposing a framework for industry. It establishes the terminology for designing KPIs. ISO 22400-2 compiles a list of KPIs that can be implemented and used in manufacturing industry. The standard provides definitions, scopes, formulas and descriptions for capturing KPIs. Overall Equipment Effectiveness (OEE), and Overall Throughput Effectiveness (OTE) provide industrially adopted KPIs for determining the efficiency of machines and systems.

MESA developed the Key Performance Indicators Markup Language (KPIML) that enables capturing and exchanging KPIs in manufacturing organisations. This language can be used to model KPIs as per the standard in form of XML schema. This KPIML implementation basis its realisation on two standards the ISO 22400-1 and ANSI/ISA-95.00.05-2006 Enterprise-Control System Integration Part 5: Business to Manufacturing Transactions.

A relevant gap in this standard is towards generalisation, relationship, conditions and computation. The approach taken by the standard to define the KPI is too general to be defined for all production operations. Certain set of KPIs need to be defined per application. A means of classifying which KPI is relevant must be in place. A means of redefining KPIs for application must be instated. In the standard the boundaries of each KPI are defined, however, the relationship and working conditions for each application is different. There is a need for representing useful elements for defining KPI as per application. Some of the defined KPIs have limitation as they cannot be computed, so a more grounded outlook is needed enforced by mathematical representation.

2.5 Assessment of Data Models

Once presented main existing data models, standards and implementations, this section aims at evaluating data standards against main requirements from section 2.2. This literature represents a base to build the data model that considers the runtime condition of machines that enables a proactive and smart manufacturing system. Table 2 summarises main requirements in existing data models and standards.

Singular manufacturing assets are *modular* if they are defined unambiguously and are represented in a logical structure [26, 52, 53, 54]. The structure facilitates keeping track of the correlation of data and their associated components, including physical or virtual ones. Each singular asset is then coupled to an element of the data model, clearly defined by associated attributes, such as identifiers, definition, functions, and others. However, the asset representation or design in the data model must be isolated from the underlying technology and computing mechanism as well [55].

The *integration* requirement may be assured by the use of profiles or subsets, helping to specify the main data characteristics to which data assets or agents may claim conformance (e.g. using a conformance model [55]). Detailed descriptions of runtime conditions for shop-level assets including personnel [52], or components like PLCs, CNCs, robots, or simpler ones like sensors, may need to be combined and related (i.e., horizontal integration). Similarly, this integration should allow data usage by higher level services or applications as required (i.e., vertical integration) [53]. For that purpose, the data structures must enhance the communication capabilities by ensuring access to *interoperable* mechanisms for data sharing in accordance to a set of rules and mechanisms [26, 54].

The use of dictionaries and semantic data models means that implementers may extend the data model to include information required for a specific implementation in an easy and quick manner allowing effortless deployment and expansion of the initial design, thus improving *scalability* of the data model [26]. A tree-like structure may simplify the model design under a *hierarchical* or logical structure representing relations and abstraction levels with a detailed description of the assets in terms of functions or activities [54].

Due to the aforementioned requirements, systems - and then their data model - may require highly complex representations, not easy to be represented or understood by humans. In this sense, a *graphical* description to capture a simple and understandable structure of the syntax, behaviour, meaning and relationships is highly desirable.

According to the assets' operational context data storage/access mechanisms may be challenged. For example, the volume and the frequency of data collection may be large, and data may be a combination of static and dynamic data types. The data model must be built considering *time restrictions*, facilitating data writing and reading access while assuring *time traceability* for later monitoring and analysis purposes [53, 54].

Lastly, model *robustness* must be assured to support industrial acceptance from its first conception [55]. That will give a strategic ability to remain functional and to keep information always available, assuring the system's reliable response under disturbances and changes - internal or external -, keeping secure communication mechanisms and a system prone to errors during the execution time.

								Assets					
	Modularity	Integration	Scalability	Robustness	Hierarchical	Temporal charact.	Graph represent.	General	PLCs	CNCs	Robots	Personnel	Implementation
MT Connect [26]	✓	✓	✓	✓	✓	✓				✓			[23, 24, 25, 30]
OPC UA [55]	✓	✓		✓				✓	✓	✓[31]	✓ [56]		[36, 40, 43, 45, 46, 48]
ISO 23247 [52]	✓							✓	✓	✓	✓	✓	
ISO 16300		✓		✓				✓					
ISO 22400 [54]		✓			✓	✓	✓	✓					
AAS type 1	✓		✓	(-)	✓		✓	✓					[39, 36, 40, 41, 43, 44, 45, 48]
AAS type 2	✓	(-)	✓	(-)	✓	(-)	✓	✓					[37, 36, 40, 42, 43, 44, 48]
AAS type 3	✓	✓	✓	✓	✓	✓	✓	✓					[46]

Table 2: List of covered requirements in existing data models and standards.

The requirements established in Table 2 can be used to adapt the production system to run-time condition necessities. The production system can be expanded to include aspects as that address the run-time condition requirements. Runtime condition modelling emphasises on the need for modularity, integration, scalability, robustness, hierarchy, temporal characteristics and visual representations. Converging to the level of production systems, these conceptual needs must be grounded to production system concepts.

The ISO 22400 establishes assessment indicators of general assets under a hierarchical structure, considering temporal variations and the transitory nature of parameters. In contrast, ISO 23247 and ISO 16300 set the requirements for digitalising different assets - such as PLCs, CNCs, and robots - considering their integration and robustness.

Almost all the ISO standards and standard data models have been conceived to be modular. In consequence, guidance on how to represent general manufacturing assets can be found, while only few consider specific attributes for more specific assets - i.e., PLCs, robots, personnel or CNCs. The integration of the assets is the main objective of all standards, although it represents a huge challenge in practice, being only enabled by the automatic mechanisms deployed in AAS of type 3 (and in some measure also in type 2). The highly desired requirement of robustness requires security mechanisms such as privacy and cybersecurity policies, encryption, protection and authentication; which may be partially accomplish in AAS of type 1 and 2, but fully completed in type 3.

The models and standards discussed in earlier section address some or individual components of the needs, but not all or in context of runtime-conditions. There exists a need to further expand or adapt these standards and models to run-time condition needs. Therefore, a model must be developed that takes aspects of all these standards and models, moulds them in context of run-time conditions and presents an approach to represent the run-time condition for production systems at an instance.

A production system model that enables the needs for run-time conditions, can be represented by including concepts of requirements, configuration, capability, operations and processes, constraints, key performance indicators (KPIs) and relationships. These concepts are broken down to finer granularity to provide an approach to capture this information for production systems. These granular decompositions of the concepts are their dependencies.

For run-time conditions in production systems, these dependencies capture all the information necessary to represent the internal state of the production system and external state of the production environment. The status of the manufacturing assets are represented, like equipment and products, for monitoring and control in line with production execution. This run-time condition representation is necessary to encapsulate the manufacturing operation. This representation, combined with enabling technologies, ensures that correct manufacturing operation is carried out with proper identification and monitoring along with controlled behavioural changes.

The individual components that fulfil the requirements established in the table2 are shown in figure 8 and detailed as follows;

- **Configuration:** This represents the internal state of the production system, under the assumption that a production system consists of entities that can have multiple arrangements and settings under particular constraints. Therefore, we consider the configuration representation for the production system to have variables that capture the settings of the production system, any relationships that exist between entities of the production system (both cyber and physical), and constraints that affect these relationships and variables.

- **Relationship:** This captures the relations that exist in production system and its relation to the environment. It consists of the concepts of dependence, composition, matching and compatibility. Dependence captures the essence of entity relation in production system and its environment. Composeability breaks down the entity relationship in production system to granular level. The granular level entails a base level assumed, beyond which further breaking down does not exist. Matching is necessary to maintain the integrity of the production system entities among themselves and the environment. A production operation may only be executed if matching exists between entities. A further extension of concept of matching is considered as compatibility. The entities in production system must exist with each other in a state where execution may be possible without issues and problems.

- **Capability:** It is the ability of the production system to perform action. It is further distinguished to cyber and physical capability.
- **Information:** This represents the data that constitutes the runtime-conditions. It has a data type, identification, a type (static or dynamic), a value and a unit of measure. It can also contain information about data aggregation, i.e., if it exists as an aggregated value from multiple sources.
- **Constraints:** This presents the limitation on entities that make up production system. This can be internal or external limitations.
- **KPI:** This captures the goals that govern the production system. It acts on the entity level consisting of attributes. These attributes can be cost, time or quality etc., given by identification, threshold, scope, associated entity, and associated formula.
- **Operations and Processes:** This contains representation for product, process and service. It provides a means of extension to these sub-models.
- **Requirements:** It instates information about necessities of entities like the current status, scheduled program it follows, loaded program, location information, and time specific information.

The information needed to represent run-time condition in the mind-map needs to be captured. Our model provides an approach that captures these listed aspects. A generalised case of this representation is presented to aid future extensions.

2.6 Data Model for Runtime Condition

A generalised representation for runtime-condition is depicted in figure 9. The components are detailed individually and related to each other to establish the collective representation. The detailed data model builds on the requirement generated in previous section 3.5. Each component, fulfilling the requirement, is elaborated, defined and detailed.

The section follows a pictorial representation of the runtime-condition data model. This is followed by elaboration of each component. The components are defined, their relationship to other components established, and the UML modelling for each component detailed.

2.6.1 Asset

The *asset* represents one of the main entities - physical or logical agents - of the manufacturing system, thus it is depicted as a main object of the runtime data model.

The asset object contains informative attributes regarding its own identification, such as *ID*, *name*, *description*, and *timeslot*; its current physical position or *location*; and function or role in the manufacturing system. Since the asset is a representation of an agent in the system that performs

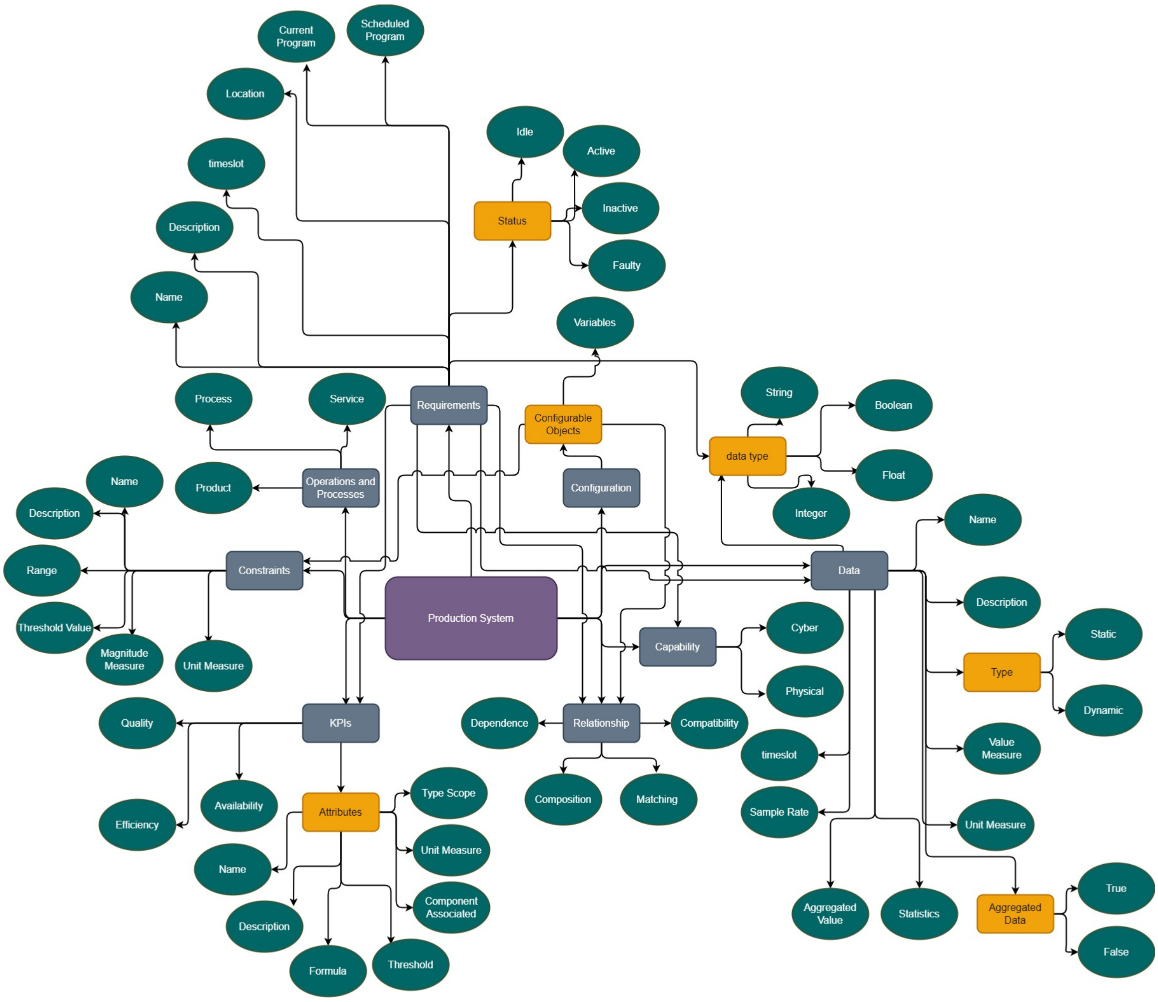


Figure 8: Mind mapping the runtime conditions in production systems

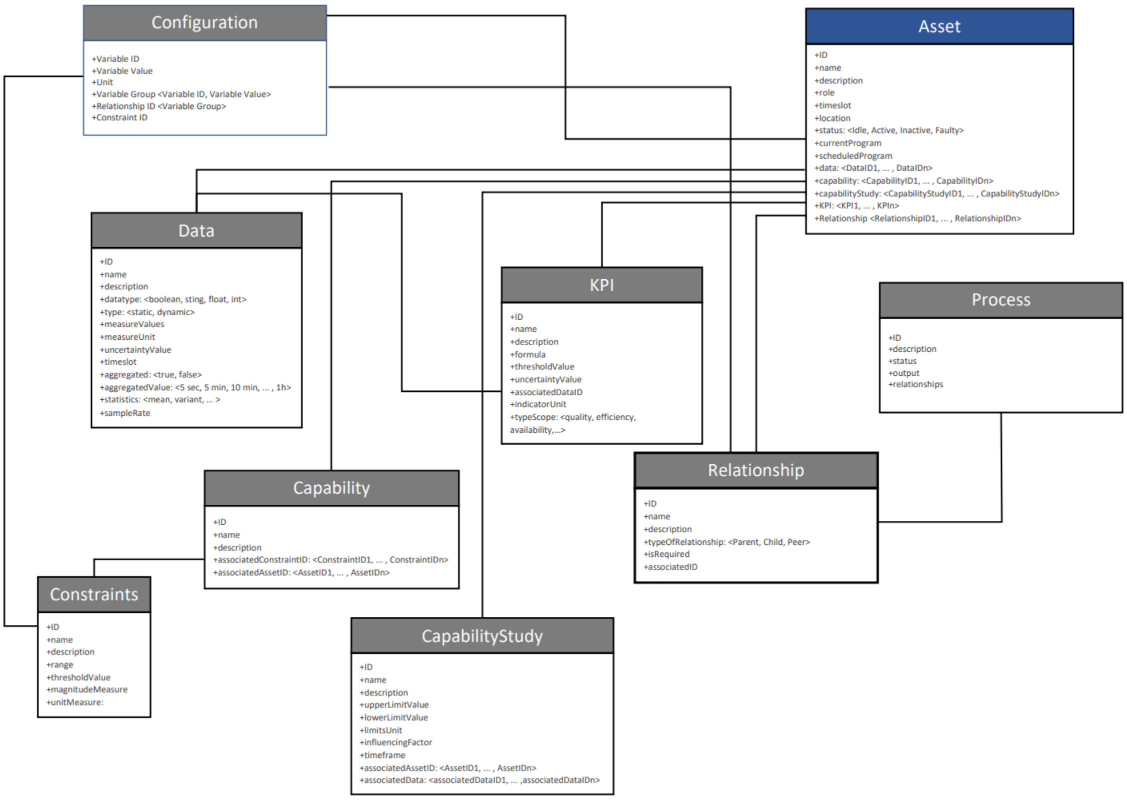


Figure 9: A generalised representation of runtime conditions for a production system. This representation combines all components for runtime condition discussed in previous section.

different tasks, fields that define the current working plan and conditions must be included, that is *current program*, *scheduled program*, and *status*. In addition, further couplings with extended objects are required to complete the definition according to *data* objects, *capabilities*, *performance indicators* and *relationship* (see section 2.6.6, section 2.6.4, section 2.6.3 and section 2.6.2, respectively). The UML syntax for the asset modelling is shown in figure 10.

2.6.2 Relationship

The relationship modelling represents the association information between two or more *Assets* that might function independently but together perform a manufacturing process or operation. Figure 11 shows a UML representation, *Relationships* are identified by ”RelationshipId” and contain descriptive attributes such as ”name” and ”description”. The attribute ”type” allows us to describe the hierarchy of the relationship with the related asset, identified by the attribute ”associatedID”. The relationship can be critically required in a process or operation that is defined in the field ”isRequired”.

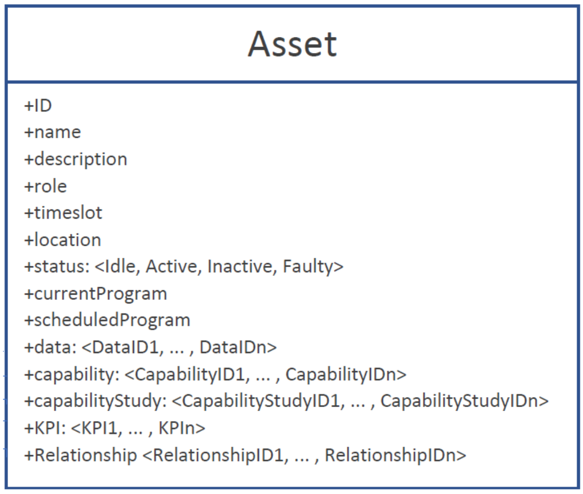


Figure 10: UML representation for Asset modelling

Figure 11: UML representation for Relationship modelling

2.6.3 KPIs

Key Performance Indicators (KPI) are metrics that are used to quantify the success of a manufacturing process step or action. It is a quantifiable measurement of the improvement or deterioration in the performance of an activity critical to the success of a process. A KPI can have a target, a set of ranges, or both, which measure how well a business is achieving its objectives. According to [57] KPIs have the following requirements for a valid KPI engineering:

- KPIs should be in a quantifiable form. i.e., it should be measurable quantitatively.
- A KPI needs to be sensitive to change: the change in a manufacturing process should be reflected in KPI.
- A KPI should be reliable: KPI calculation should be free from semantic errors.
- A KPI should be efficient: it should be cost-effective to calculate. KPI calculation should relative faster than a manufacturing process.

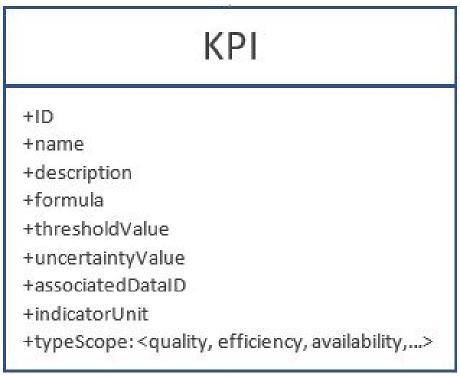


Figure 12: Process modelling for Runtime condition

2.6.4 Capability

An asset's capabilities define a feature or a specific skill of an asset. Each capability has a unique identifier, a name and a description to define a particular feature of an asset. A feature might be limited by a set of Constraints (see section 2.6.7), hence the data model should contain the respective constraintIDs. In addition, *machine* and *process* capabilities might be assessed with a capability study index (further details in section 2.6.5).

Figure 13: UML representation for Capability modelling

2.6.5 Capability Study

A capability study is the measure of the actual quality of a machine or process with respect to its specifications. Machine capability is measured by *Cm* and *Cmk*, whereas process capability is measured by *Cp* and *Cpk*.

- *Cm*: quantitative measure that indicates the spread of the machine within the specified tolerance limits.
- *Cmk*: position of the machine's capability in relation to the specified tolerance limits
- *Cp*: quantitative measure that indicates the spread of the process within the specified tolerance limits.
- *Cpk*: position of the process's capability in relation to the specified tolerance limits.

A capability study has a unique identifier, a name and a description describing the respective quality measure of the asset (i.e. machine, process), and the tolerance limits (upper limit and lower limit) of the index under study. Measurements might differ due to internal or external factors such as the quality of the measuring instrument, wear degradation, environmental conditions, among others. These factors should be therefore included within the capability study UML model.

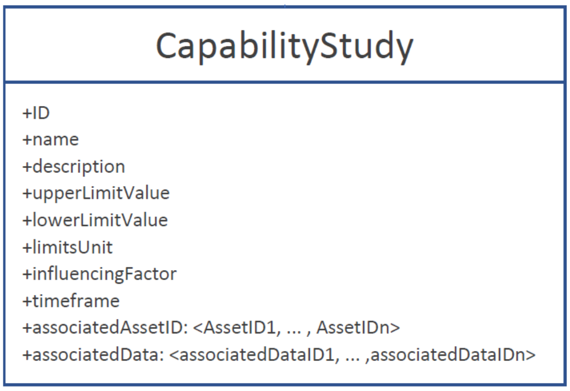


Figure 14: UML representation for Capability Study modelling

2.6.6 Data

The *data* object defines data directly collected from different levels of the manufacturing floor using sensors in a comprehensive and unambiguous manner.

Due to its varied nature it requires not only identification attributes, such as *ID*, *name* and *description*, but the magnitude identification, i.e. *value* and *unit* measure. Attributes that identify the *uncertainty* of the measurement are required as well for later identification of possible source of errors or data sources comparison. Post processing of data into information extraction may be required, so that attributes as *aggregated* attributes and *sample rate* attributes are also relevant. Figure 15 shows the UML representation of the data object.

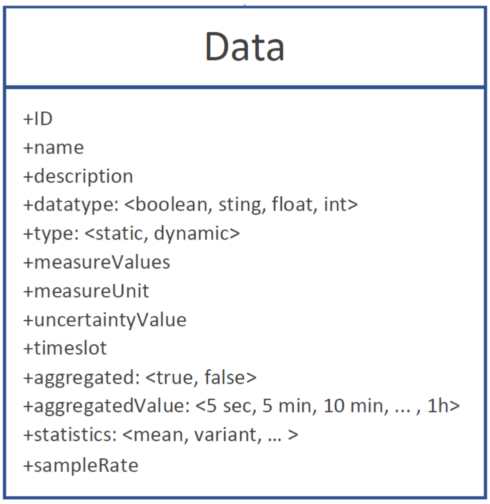
2.6.7 Constraints

The constraints modelling for runtime condition expands the “Constraints ID” listed in capability model. The capability aspect in a production system can be realised under consideration of the constraints it is subjected to. The constraints represent limitations in the capabilities of the production system.

The constraints are identified by a “Constraints ID”, same as that given in capability representation. A “Name” is used to describe the constraint along with a “Description”. The value for the constraint is given by “Range” and its accompanying “Unit”. The “Threshold Value” gives the optimal value for the constraint and variation from it results in a control mechanism trying to maintain the capability as per optimal. The general syntax for constraint is given in figure 16.

2.6.8 Configuration

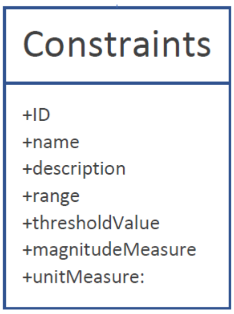
Configuration modelling consists of “Variables ”, their accompanying “Relationship” and the “Constraints” that impact these “Variables”.



Data

+ID
+name
+description
+datatype: <boolean, sting, float, int>
+type: <static, dynamic>
+measureValues
+measureUnit
+uncertaintyValue
+timeslot
+aggregated: <true, false>
+aggregatedValue: <5 sec, 5 min, 10 min, … , 1h>
+statistics: <mean, variant, … >
+sampleRate

Figure 15: UML representation for data modelling

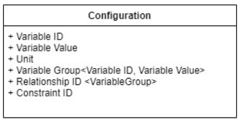


Constraints

+ID
+name
+description
+range
+thresholdValue
+magnitudeMeasure
+unitMeasure:

Figure 16: Constraint modelling for Runtime Condition

The configuration is given by “Variable ID” and its “Variable Value”. Accounting for variables that constitute the production system “Variable Group” links ID to value. “Relationship ID” expands the “Variable Group” as per relationship representation discussed and “Constraint ID” provides link to constraint modelling.The general syntax for constraint is given in figure 17.



Configuration

+ Variable ID
+ Variable Value
+ Unit
+ Variable Group<Variable ID, Variable Value>
+ Relationship ID <VariableGroup>
+ Constraint ID

Figure 17: Configuration modelling for Runtime COndition

2.6.9 Processes and Operations

A process represents a set of operations necessary to complete a manufacturing task. This entity will allow integration with specific *process modelling* data models. As seen in Figure 18, each operation has a unique identifier, a description, the current status and output of the process. The attribute ”relationships” establish the necessary associations of assets to fulfill the process.

Figure 18: Process modelling for Runtime condition

3 Process Model

In the context of Industry 4.0, products become information carriers which can proactively control their creation by communicating their manufacturing requirements to process planning systems. To support this capability, an information structure is needed: a product data model [58]. At the same time, to be able to understand the product requirements and allocate the appropriate resources for their production, the manufacturing processes need to be modelled.

Modelling can be seen as the translation of concepts, relationships, constraints and characteristics into a semantic representation for a specific domain, that is stable and organized [59]. For a manufacturing process, a modelling approach can help in defining:

- what steps does a product require to be manufactured (i.e., select the suitable manufacturing technology);
- what attributes of each process step must be accounted for (e.g., parameters, tools, materials);
- in what order do the steps need to occur in (i.e., work plan).

In this section, we examine the existing endeavours in representing manufacturing processes, as well as the definition of processes, aiming to provide our own definition around which a generic model for manufacturing processes will be proposed. With this aim, first, relevant manufacturing standards are presented to understand what each of them perceive as process and the information requirements they have established for it; secondly, some modeling approaches are analyzed to understand their advantages and shortcomings in representing processes; thirdly, the requirements of all these sources related to both, information requirements and characteristics of the resulting model, are used to asses the aforementioned approaches; finally, a generic model for manufacturing process is proposed based on the collected requirements.

3.1 Existing modeling approaches for manufacturing processes

To better describe the modeling approaches for the manufacturing process, some current existing standards and models of the manufacturing process are researched. Firstly, the current relevant standards for the manufacturing process are discussed, i.e., the specifications that standardization organizations at national or international level have designated for product and manufacturing information representation, in this case specifically for processes. Then, a sample of the models proposed by the manufacturing community, based on those standards or other approaches, for connecting product data to the involved processes and resources is presented.

3.1.1 Relevant standards

The manufacturing standard for the process is the minimum level of quality required to produce a product that meets the customer's requirements. This can include things such as quality control procedures and methods used to ensure that products meet specifications, product design and development; and processes used in production (such as assembly lines).

IS0 10303

The suite of standards ISO 10303, commonly referred to as STEP (Standard for the Exchange of Product Model Data), resulted from one of the largest standardization efforts taken by ISO, which started in 1984 [60] and was closely related to the Product Data Exchange Specification (PDES) proposed by NIST [61]. STEP's objective is ”to provide a neutral mechanism capable of describing

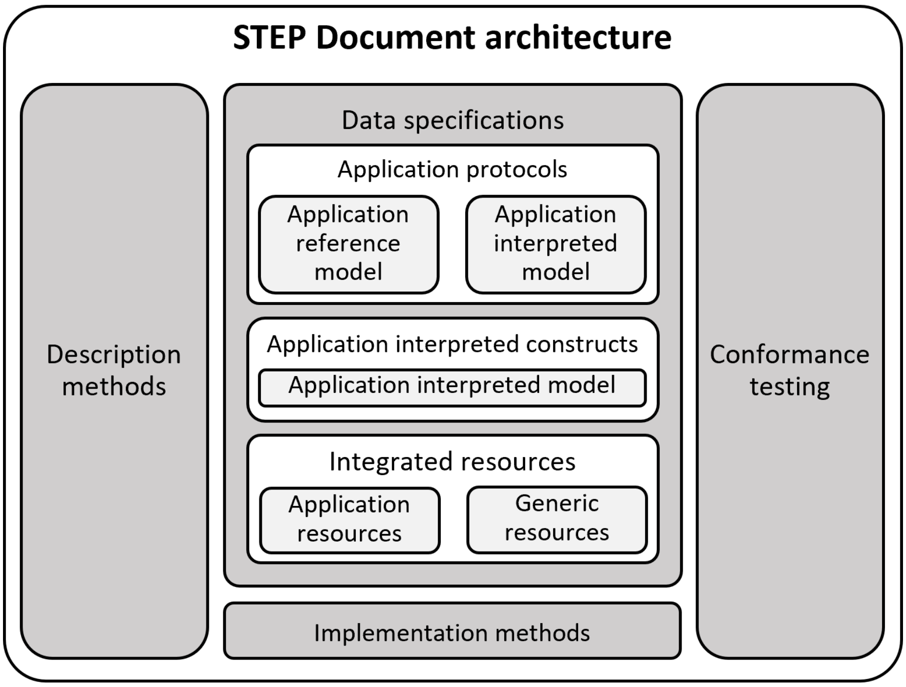


Figure 19: STEP document architecture. Based on [63]

products throughout their life cycle" [62], offering a computer-interpretable representation of product information and a mechanism for the exchange of product data.

STEP provides the characteristics that describe a product [63] in an information model, separated from the conceptional modeling and the exchange format [64]. Its information model is defined in EXPRESS, which is human- and machine-readable and captures the entities and their relationships in schemas [60]. These schemas also contain the rules for entity definition, which allows the verification of its syntactic and semantic correctness, and their appropriate linkage to other schemas. The exchanged product-specific data are instances of the entities defined in those schemas [65].

Because of the broad scope of the STEP standard, its documentation is divided in several parts, as summarized in Figure 19. The description methods include the specifications of EXPRESS and EXPRESS-G, and are defined in the 10303-1x parts. The implementation methods enclose the structure of the physical file, the standard data access interface (SDAI), and comprised the 10303-2x parts. The conformance testing contains a conformance testing methodology and framework, and abstract test suites. Finally, the data specifications include primarily the integrated resources, application protocols, and application interpreted constructs, of which the applications protocols can be considered individual standards in themselves [63].

The STEP application protocols (AP) – the only parts of the standard intended to be implemented – define the models to one or several product life cycle stages of a product class. They are built by a collection of application modules (AM), which are themselves composed of a set of fundamental constructs (integrated resources) applied to a specific domain [65]. The role of the AP is twofold; first, to understand and capture the information requirements within a domain, which is represented in an application reference model (ARM); and second, to specify the structure of data for exchange by mapping those requirements into the application integrated resources (AIM), to guarantee compatibility to other STEP parts [66]. For the purpose of modeling processes, the following parts of the STEP standard are relevant:

- Part 49. Integrated generic resources: Process structure and properties. This part of STEP defines a process as "a particular procedure for doing something involving one or more steps or operations" [67]. It also provides the basic constructs to specify the actions to execute a process, such as [67]:
  - representation of a process and resource
  - properties of a process and a resource
  - relationships between processes, and of the process to the product
  - effectivity of a process
  - resources required for the process

  These constructs can be used in the context of an application to represent a process plan data model. [68]
- Part 224. Application protocol: Mechanical product definition for process planning using machining features. This application protocol defines the information requirements to produce a process plan of a mechanical part, including "part identification, tracking, shape, representation of the shape and material data necessary for the definition of a part for process planning" [69]. The goal is to use the information about the shape, enclosed in the manufacturing features, to support automated process planning [68, 69].
- AP238 Application protocol: Model based integrated manufacturing. ISO 14649-Data model for computerized numerical controllers, commonly referred to as STEP-NC, defines the ARM to specify a feature-based machining process to computerized numerical control (CNC) machines; it represents the machining process with respect to the part, as well as their sequence, parameters and required tools [70]. AP238 translates the STEP-NC data model into the full integrated resources of STEP [71]. This AP includes the context, scope, information requirements for numerical controllers and associated processes, augmented with product geometry, geometric dimension and tolerance, and product data management information [70].
- AP242 Application protocol: Managed model-based 3D engineering. This application protocol specifies the representation of engineering data and product data based on a three dimensional model [72], converging the STEP AP203 for design geometry, and AP214 for automotive design industry. This AP has two editions. The main purpose of edition 1 was to merge AP 203 and AP214 with minor extensions, while edition extended its capabilities to the electrical design domain, product data management, 3D geometry with product manufacturing information, composite and mechanical design [73]. The result of STEP AP242 ed2 is depicted in Figure 20.

The STEP Standard and its application protocols are considered enablers for the manufacturing digital thread, especially in the design and manufacturing stages. However, manual assistance is still required to convert data in AP242 to AP238, which hinders the realization of a true manufacturing thread [22].

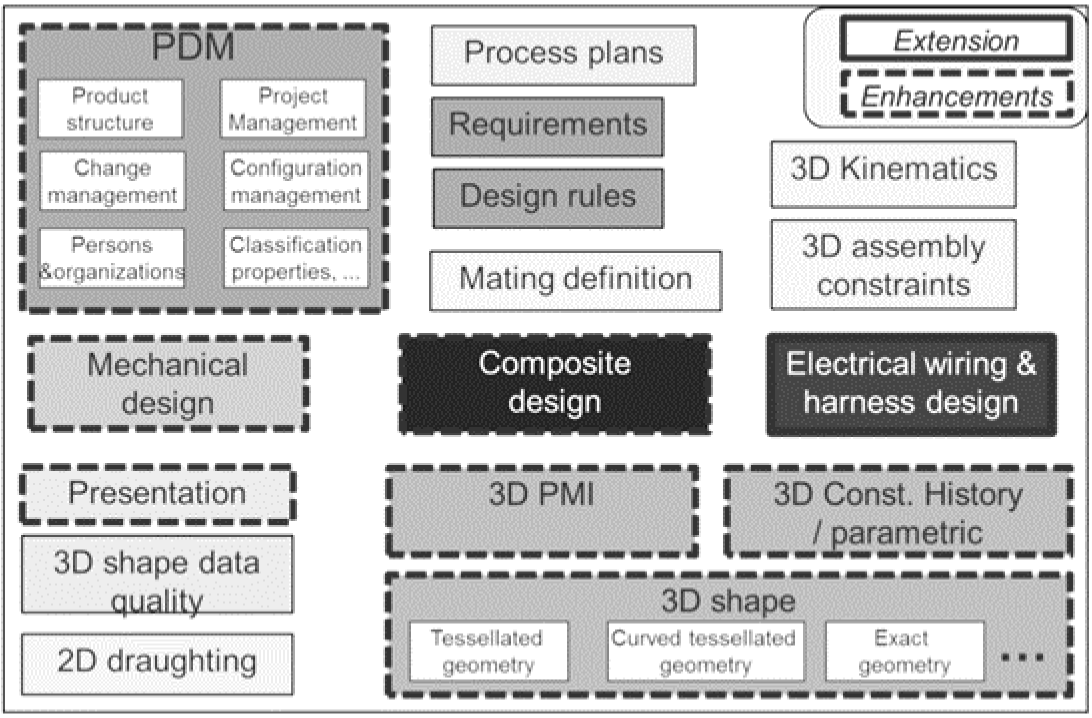


Figure 20: STEP AP242 ed2 structure [73]

ISO 23247

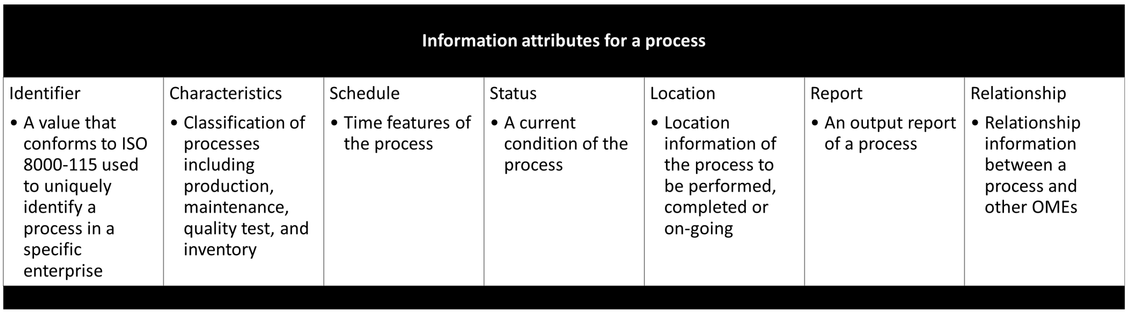
This standard provides a framework for digital twins in the context of manufacturing, including personnel, equipment, materials, manufacturing processes, facilities, environment, products, and supporting documents. The premise is having a digital twin representation that can be shared across the whole product life-cycle. Therefore, the standard defines the general principles for developing digital twins in manufacturing, a reference architecture, basic information attributes, and technical requirements for information exchange between entities within the reference architecture [74].

Regarding processes, ISO 23247 states that "a process consists of the operations necessary to complete a manufacturing task" [52]. It also offers a list of process information attributes, which can be of static or dynamic nature (see Figure 21) [52].

CMSD

The Core Manufacturing Simulation Data (CMSD) was defined by NIST and later standardized by the Simulation Interoperability Standards Organization (SISO) in SISO-STD-008-2010 and SISO-STD-008-01-2012. This standard aids the exchange of manufacturing information between simulation and other applications. It specifies the required entities to built manufacturing-oriented simulations [75].

One of the principal entities proposed by CMSD is the process plan that specifies the activities needed to transform materials into final products. The process plan class is related to a process class, which defines a process as "a manufacturing activity or group or manufacturing activities that are

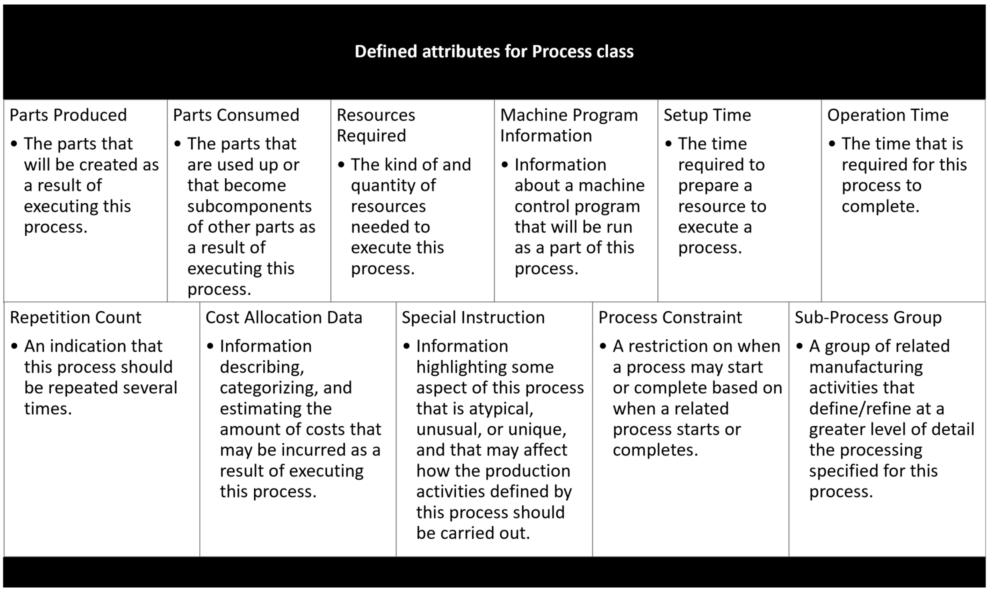


Information attributes for a process

Identifier	Characteristics	Schedule	Status	Location	Report	Relationship
• A value that conforms to ISO 8000-115 used to uniquely identify a process in a specific enterprise	• Classification of processes including production, maintenance, quality test, and inventory	• Time features of the process	• A current condition of the process	• Location information of the process to be performed, completed or on-going	• An output report of a process	• Relationship information between a process and other OMEs

Figure 21: ISO 23247 attributes for a process. Based on [52]

a part of a detailed strategy for creating a part" [76], and also specifies the information attributes required to represent a process (see Figure 22).



Defined attributes for Process class

Parts Produced	Parts Consumed	Resources Required	Machine Program Information	Setup Time	Operation Time
• The parts that will be created as a result of executing this process.	• The parts that are used up or that become subcomponents of other parts as a result of executing this process.	• The kind of and quantity of resources needed to execute this process.	• Information about a machine control program that will be run as a part of this process.	• The time required to prepare a resource to execute a process.	• The time that is required for this process to complete.

Repetition Count	Cost Allocation Data	Special Instruction	Process Constraint	Sub-Process Group
• An indication that this process should be repeated several times.	• Information describing, categorizing, and estimating the amount of costs that may be incurred as a result of executing this process.	• Information highlighting some aspect of this process that is atypical, unusual, or unique, and that may affect how the production activities defined by this process should be carried out.	• A restriction on when a process may start or complete based on when a related process starts or completes.	• A group of related manufacturing activities that define/refine at a greater level of detail the processing specified for this process.

Figure 22: CMSD process class attributes. Based on [76]

ISO 18629 (PSL)

The Process Specification Language aims at facilitating interoperability for industrial data integration. It is written in Knowledge Interchange Format (KIF), is based on set theory and first order logic, and offers a set of definitions, relations and axioms for representing processes in several applications, especially discrete manufacturing [77].

This standard defines a process as "structured set of activities involving various enterprise entities, that is designed and organised for a given purpose" [78]. It provides the semantic to describe a process,through a lexicon, an ontology, and a grammar, independent of its representation model along the product life-cycle.

3.1.2 Relevant models

A manufacturing process model is a way of describing the manufacturing process. It is used to describe the manufacturing processes in order to make it easy to understand, analyze and control. The manufacturing process model also helps to identify problems during current production processes.

Knowledge Based Generation of Petri Net Representation of Manufacturing Process Model Entities

Researchers have been investigating the industrial production process and the relation between design and manufacturing for several years. Originally, organizational and data issues were tackled separately as sub-problems, either concerning the geometry domain or the process domain [79].

Although still written on paper for distribution through the manufacturing chain, process plans were already being defined and comprised of:

- input material;
- process activity sequences;
- machine tools and fixtures;
- estimated machining time.

However, the translation of such plans into NC-program (Numerical Control) was still done manually, until the spread of CAD systems and APT-language (Automatically Programmed Tools), technologies that allowed for the use of CAD-models as input data for NC machines, integration of tools and fixtures databases, and extension to process planning functions [80].

New process modeling concepts developed for the integration of CAD model on the manufacturing floor were already using standards such as STEP to avoid hardware dependencies and differences in data structures.

A structured analysis of process planning functions became the base for modelling integrated process planning: a number of features started being defined, such as basic components for integrated parts and process modelling. Above the so-called features stood instances of manufacturing process feature classes, a higher level definition of what was needed in a model: such instances are still known as model entities [81]. Such features and entities are the foundation of any manufacturing process; a complete model should take into account every possible feature, for every possible process that could be used in any plant. Such model does not exists and it is also not computationally viable.

When methodologies for creating such models started being used, manufacturers needed something that would – in the first place – be applicable for a given task, and then extendable to the general process, and could be a standard for future modifications [82]. This approach is still viable nowadays, and is a focus point of new data models.

Therefore a structured four-level approach to manufacturing process modeling was proposed, where process features were placed into this structure (see also Figure 23):

- process (Ll);
- setup (L2);
- operation (L3);

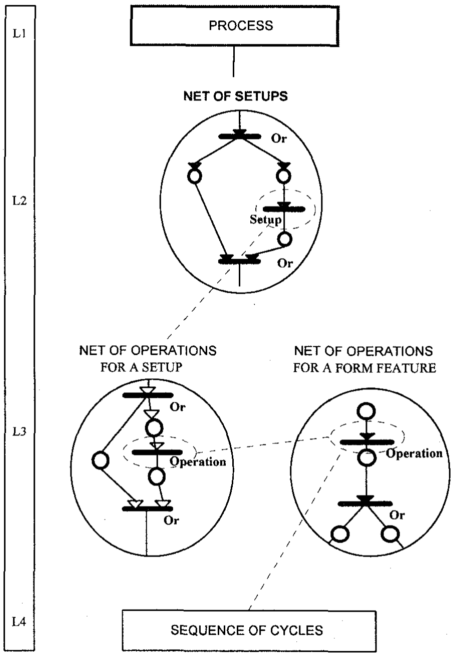


Figure 23: Levels and features in the process model [81]

- numerical control (NC) cycle (L4).

This was a method for creating manufacturing process model founded on Petri Net (PN) model representations, which are useful to describe generic model entities. The generation of process model entities used classification trees, structural description and unstructured knowledge [81].

Since knowledge technology was applied when creating the model, and the process model entities contained knowledge for evaluation, the process model included data exchanged between different modeling systems. The modeling method could be applied for developing plans oriented to the inclusion of open CAD/CAM systems in the manufacturing environment [82].

PNs have since been used to model a wide range of manufacturing systems and system elements; they are based on four primitive elements (tokens, places, transitions, and arcs) and the rules to follow the operation flow. PNs provide a number of advantages for modeling manufacturing systems, they can easily be extended or synthesized and many software provide compatibility with this data modelling approach [83].

Description of manufacturing processes using AutomationML

The engineering of manufacturing systems has changed within the last years: intelligent tools and control devices have now different and higher level of abstraction than before, and can be programmed, configured and customized differently.

These tools are heterogeneous, for they are internally developed and provided by separate vendors. Standardization can help developing and applying a unified and élite group of tools; one of the main barriers for standardizing the creation of tools is the format of data that is exchanged within tools and machines.

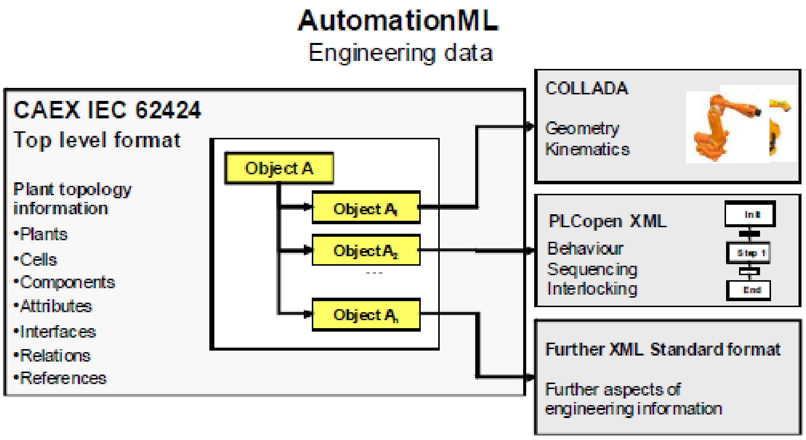


Figure 24: Basic structure of AutomationML [86]

Process descriptions and process modeling have resulted in several different exchange formats, since they have been considered separately within different areas of manufacturing. Three of the most known and used are STEP, ISA 95, and PSL, though the existence of many other worthy formats should be highlighted.

As already mentioned, STEP is the acronym for Standard for the Exchange of Product model data and is specified within ISO 10303 [84]. STEP provides means for the expression and exchange of product information along the complete product life cycle. As production process information are an integral part of process descriptions, the STEP standard also addresses process description. Nevertheless, these descriptions cover only product related process information. Manufacturing resource related information are not included. In addition, only some parts of STEP are based on XML representations. For the other two aforementioned formats, literature is available and recommended [85, 77].

Since information exchange among tools has to be lossless, fault free, and with the same semantics of the information, and no consistent process modelling methodology and no data exchange format based on it are available (at least the ones usable within a wide range of manufacturing applications). A report from 2010 described such a data exchange format to solve this issue, aiming at examining problems and creating solutions through the AutomationML consortium, based on an initiative of KUKA-Roboter GmbH [86]. The Automation Markup Language (AutomationML) was developed by the AutomationML initiative from 2006 [87]. The reader can find a complete description of the parts of AutomationML in the literature [86], and also see a schematic structure in Figure 24.

Process description in AutomationML: to achieve a simple but thorough description of any process within the AutomationML environment, created products and available resources are needed, because they are the foundation of the process itself. The following are the relevant information for the data model:

- sequence of activities;
- process activities in relation to the product properties;
- necessary resources and their parameters.

Special roles for processes, products and resources are defined within the AutomationML, enabling an evident separation of the three data sets and a clear interconnection. The representation of the process as a whole comes from linking resources and products, using the sequence of process actions [86].

The power of AutomationML is standardization: this format for describing processes takes into account process actions and parameters and relates them to products and available manufacturing resources. It can be used in all kinds of manufacturing installments, it is not only limited to a small range of applications. Since the AutomationML data format covers all relevant information required to model processes within all the engineering phases of the product, the description of the process can be used within different engineering tools along the engineering process, with no changes, with no limitations, with no stoppage and re-calibration [86].

CPM 2: A revised core product model for representing design information

The Core Product Model (CPM) was introduced by the National Institute of Standards and Technology (NIST), with the aim of providing a product representation comprising engineering concepts beyond geometry, and mapping relationships between form and function [88]. A reviewed version of this model, the CPM 2 [89], was presented as a progress report to document the changes made in CPM based on experiences working with it in other projects, as well as to present its implementations in JAVA and XML.

Because the primary motivation of the CPM model was to support product information representation throughout the whole product development process, some of its entities provide information that can be useful for product realization. CPM key entity is the Artifact. An Artifact is associated by its attributes to the Specification and is an aggregation of Form, Function and Behaviour. Another CPM core entity is the Feature, which is defined as a portion of the Artifact and is an aggregation of Form and Function. Additionally, Form is an aggregation of Geometry and Material, and Specification is an aggregation of Requirement. CPM also contains Relationship entities and Utility classes.

As can be seen, CPM's focus is the information produced during design stages and will need support from other representations to communicate with subsequent manufacturing activities. Still, some of the product's attributes could implicitly constrain the manufacturing process to be used for its manufacturing depending on the level of detail used to represent certain entities, e.g., requirement and specification. However, implicit information representation is not a suitable approach for production stages, which requires very specific and fully detailed information. An information model specifically defined for representing processes that deliver explicit and complete information for product manufacturing would be a more appropriate approach.

A manufacturing process information model for design and process planning integration

Another effort towards interoperability proposed by NIST was an object-oriented manufacturing process information model capable of representing manufacturing processes and the concurrent, parallel or alternative activities involved, as well as manufacturing equipment, estimated costs and time [90]. This model was later used as a base for the ISO 16100, which provides "standard interface specifications that allow information exchange among software units in industrial automation systems developed by different vendors" [91].

The model, represented in the Unified Modeling Language (UML), specifies the manufacturing processes of an 'Artifact to be made' using the Manufacturing Activity entity as its core. This activity is described by the Manufacturing Parameters, Manufacturing Resource and Cost & Time classes, and has Setup, Handling, Processing, Load/Unload and Idling as sub-classes (see Figure 25). Each of these sub-classes has several aggregations, e.g., the Processing class has Inspection, Part

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

Making and Assembly as subsequent classes, and those classes contain other aggregated classes until an activity is fully described and the Manufacturing parameters are complete. In the case of Cost & Time, an estimation is calculated based on the resources used and an overhead cost, according to several equations proposed by the authors. For the Manufacturing resource, two main types are represented, Labour skill and Manufacturing equipment; the last one requires several aggregations according to the type of equipment and tool required for a specific process.

The proposed model was tested and implemented in a process planning prototype system, being able to capture all the information related to the process plan in a relational database. Other inherent advantages of the proposed model were its flexibility to be extended and reused, its open and neutral nature, and its capability to feed back information related to costs and time to the designers from early stages of the product development. However, NIST's individual information models were abandoned later to concentrate its efforts in supporting development of the STEP application protocols [63].

A PDES/STEP-based information model for computer-aided process planning

In the area of computer-aided process planing (CAPP), Ming, Mak & Yan [92] proposed an approach that not only aimed to represent all the information needed by CAPP systems, but also to integrate it with other manufacturing systems. The authors established that three models would be needed for this objective:

- a part information model,
- a process plan information model, and
- a production resource information model.

The proposed information models were represented in an object-oriented structure and complied with the early PDES/STEP standard (ISO 10303). The part information model describes the nominal shape (including general information), features (form, functional and manufacturing features), dimensions and tolerances, material, and surfaces. In the case of the process plan information model, it was divided into four hierarchical levels: generic plan, referring to the metal removing processes and general manufacturing information; macro plan, containing the sequence of processes in the process routes; detailed plan, specifying the setup methods, machine datums, and fixtures; and the micro plan, determining the process parameters, tools, costs and time. As for the production resource information model, it considered the representation of manufacturing resources such as machine tools, cutting tools, gauge tools, fixtures, and external process information. The three models were related to one another as shown in Figure 26.

The model was represented in the EXPRESS language and complies with the PDES/STEP standard, which guarantees semantic correctness. The authors claim that the model includes almost all information required by CAPP systems. However, an implementation of the model in a complex CAPP system was not realized due to the lack of a tool to translate EXPRESS into another programming language at the time and other processing limitations; therefore, its capabilities of integration with other manufacturing systems were not assessed.

GO0DMAN Data Model – Interoperability in Multistage Zero Defect Manufacturing

This study proposes a common data model that takes into account not only the production capabilities of the various components, but also the description of all the events, variables, and resources that can point to quality issues.

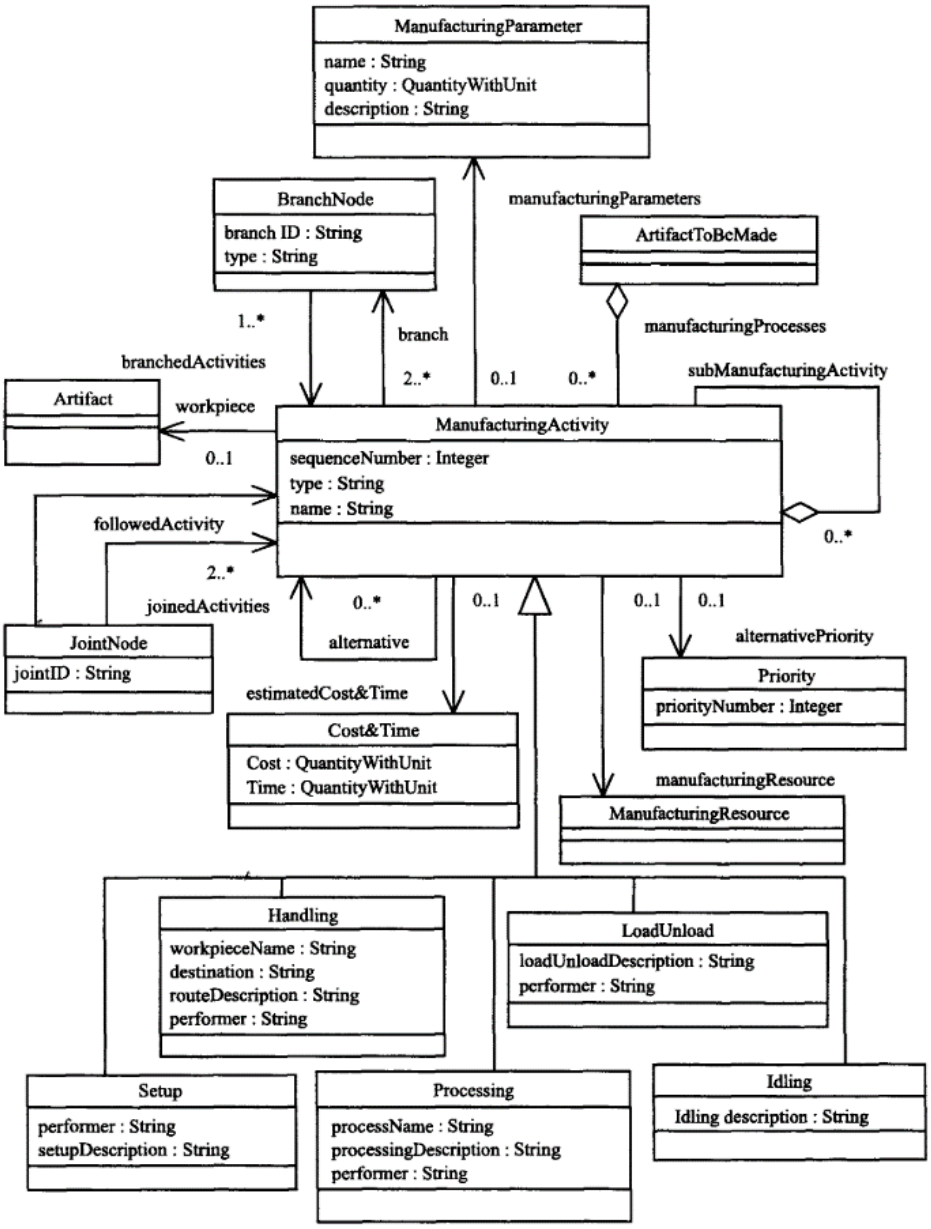


Figure 25: Manufacturing process object model main entities [90]

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

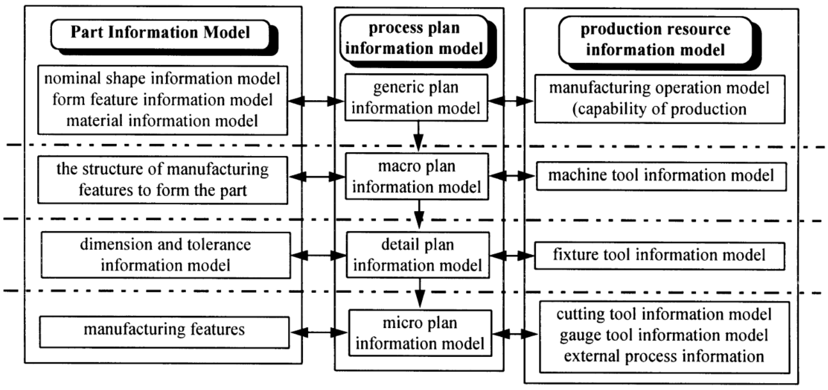


Figure 26: Relationship between part, process plan, and production resource information model [92]

While the CPS's multi-agent component must deal with its own data exchange interactions with the various agents that make up the system, it also must integrate data from other engineering domains, including in this case data from the Data & Knowledge Management layer and quality control data from the multi-stage production system and its corresponding SIT for quality control.

To address this, GO0DMAN employs a shared data representation not only to facilitate communication between agents within the MAS, but also to enable both generic and application/domain-specific semantic and syntactic descriptions, allowing for the specification of use cases regarding topology and functionality as blueprints to aid in the deployment, management, and pluggability of the MAS-CPS (Figure 27).

A detailed description of the GO0DMAN data model's design is provided, including the GO0DMan Topology (GMResource, GMSubsystem, GMsystem, GMConnector, GMValue), GO0DMan Product, GO0DMan Skills, GO0DMAN Management Rules, GO0DMAN Events, GO0DMAN Quality Control. This paper provides a detailed overview of both the design and implementation stages of the GO0DMAN Data Model, with the goal of assisting in overcoming the interoperability difficulty that is commonly encountered in these sorts of I4.0 heterogeneous systems.

Interoperability of manufacturing applications using the Core Manufacturing Simulation Data (CMSD) standard information model

Using the Core Manufacturing Simulation Data Information Model (CMSDIM), this study proposes a method for improving interoperability across production applications in order to expedite design and manufacturing processes throughout the product life cycle. To do this, a system foundation is initially developed to allow interoperability.

Figure 28 provides an overview of the scope of the work.

The framework can increase interoperability across manufacturing applications throughout numerous phases of the product life cycle, including the design for manufacturing process and production planning and control.

Figure 29 shows the benefits of using a standard neutral information model such as CMSD as a neutral information format for translation among various manufacturing applications.

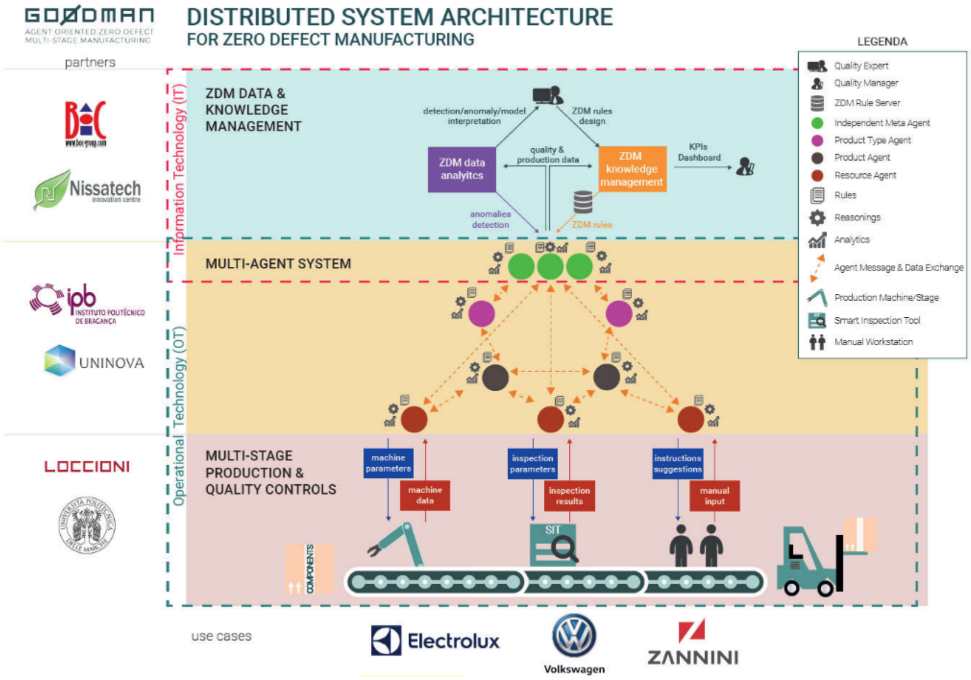


Figure 27: The GO0DMAN Distributed System Architecture

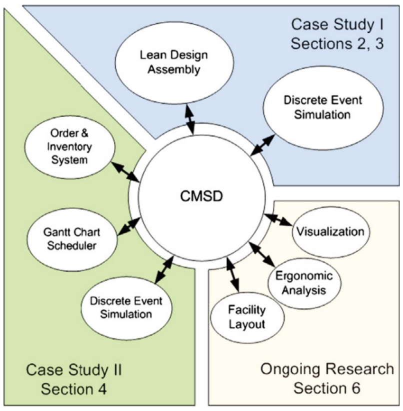


Figure 28: Scope of proposed work for applications using CMSD

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

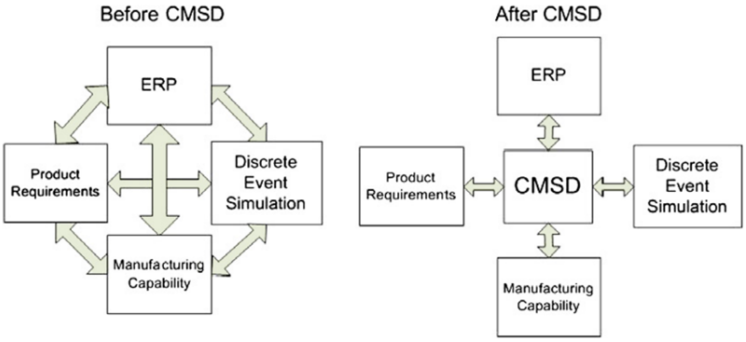


Figure 29: Simplicity of implementing CMSD approach

Process-oriented Information Model (PIM) for sustainable manufacturing

A process-oriented information model (PIM), focused on integration of sustainable manufacturing with the product design information, is proposed in [93]. It is designed to assist the product designers in making sustainability related choices at the stage of product design. The model has two components. The first component is a generic process-oriented information model which contains high-level information. It defines the broader concepts related to the manufacturing processes, products, materials, and sustainability. The second component has expanded models that are based on the generic process information model (GPIM). It includes low level information regarding the manufacturing processes. The first one works as the information core but does not work without the expanded models. Thus, the expanded models must be first defined. The paper focused on GPIM by proposing a sample scenario of expanded model related to injection molding process. Unified modeling language or UM was used to create a class diagram of entities and relationships of the PIM as the modeling method. The interface-workflow of the model, to get the energy consumption, contained several steps. It starts with login to the database where a manufacturing process can be selected for evaluation. Now a material must be chosen for the part with input of appropriate process parameters. By running the calculation, the user-interface will show the results. For an application such as choosing a small gear part out of the two candidate designs, based on less energy consumption to manufacture, two manufacturing processes were compared. These were injection molding and powder metallurgy. Based on the results, it was shown that the propose model can be implemented in a software to gather suitability related information, which will help product designers to reduce costs and wastes in product design and manufacturing.

A Reconfigurable Method for Intelligent Manufacturing Based on Industrial Cloud and Edge Intelligence

An intelligent manufacturing model, which is also cloud-assisted and self-organized, based on intelligent edge devices in the production system, was presented in [94]. The intelligent production edge (IPE) devices are designed to provide the traditional devices the abilities of data access and self-decision making. The model was proposed to realize a flexible and scalable manufacturing system. It is composed of many field devices, an IPE and the cloud. The IPE provides the interoperability between the cloud and field devices. The task of the cloud was defined such that it transforms a specific process into a task and forward it to the field devices through an established negotiation process. The industrial internet of things (IIoT) elements of the architecture were modeled as agents, making a multiagent system. The dynamic scheduling of the production logic from cloud to the field devices was shown by the results of this agent-based mechanism. It was found that the proposal model is more efficient and robust than traditional scheduling. More importantly, it was found to be very suitable for handling mixed flow production tasks that contain random orders. A limitation of the paper is that it does not take into account the formal modeling of production lines in depth.

PPR information model

This research focused on identification of information requirements for process planning in a concurrent engineering environment [95]. The information requirements were added in the information model using the EXPRESS language. The model identifies (functional, input and output) connections between the product, process, and manufacturing resources. It also describes different elements of the model e.g. separation of generic and specific information concepts that affect information model, and separation of the mating and joint concepts etc. Different cases studies were explored involving management of weld spot and location system information, as well as tender preparation information. The result of the research was an information model, the product process resource (PPR) information model. The PPR model defines the information requirements with consideration of different elements that can be used in information modeling e.g., modularization. A core outcome of the model is that it enables traceability of the evolution of the product, process, manufacturing resource designs, and their interrelations.

3.2 Requirements for process modeling

In this Section the requirements of the presented sources related to both information requirements and characteristics of the resulting model are assessed against the modeling approaches; the aim of the assessment is to collect all the needed requirements to then propose a generic model for manufacturing process, based on the STEP standard.

Such a model needs both requirements from the information perspective – i.e., specification of how data and information going into the model should behave and what characteristics they should have – and requirements from the modeling perspective – i.e., a list of product and process data, attributes and properties that the model should include.

The following Tables 3, 4 list the aforementioned requirements for the model, split in two distinct groups as described above.

Requirements from the information perspective

A model is needed for capturing and transferring information about process steps, to enable autonomy in a product-driven production. The information content of the model requires conditions that need to be met in order to have a precise, clear, thorough and constant stream of data between all parties involved in the sharing.

The proposed model encompasses the requirements from the process point of view; it has to contain information about how to achieve the manufacturing steps to realize the actual product (such as the used material, the needed tools and setups, and so on). However, the first step defines the characteristics of every piece of data that takes part in the model, in order to achieve consistency.

Such characteristics are briefly described in Table 3.

Table 3: Requirements and characteristics of the information content in the model

Requirements	Description	References
Consistency	When expressing complex info in process planning	[96]
Accuracy	In conveying actual manufacturing process info	[59, 96, 93, 97]
Completeness	In communicating data for description of process and products	[96, 98, 59, 97]
Generality	While supporting the expression of variety of processes	[96, 99]
Shareability	Throughout the factory and with external stakeholders	[59]
Integration (1)	Of protocols and formats between companies and industrial standards	[100, 101]
Integration (2)	Horizontal through value chain and vertical in-house from sensors to enterprise	[102, 101]
Semantics	Unified description of information and meanings in production	[102, 97]
Standardization	Across domains and companies	[100, 99, 103]
Safety	Malfunctions cause accidents and stop production – see standards S84, IEC 61508, and IEC 61511 to minimize the dangers of errors	[101, 103]
Extensibility	For the integration of new features	[59, 101, 93, 94, 97]
Availability	Large volume of data is not always possible in manufacturing systems	[103]

Requirements from the modeling perspective

Only after establishing the characteristics of the information that the model needs are we able to list the actual content of the model.

This content is shown in Table 4, with a brief description of what is needed in our model to make a step forward towards achieving autonomy in a product-driven environment.

Table 4: Content requirements for the model, i.e., what the model should include

Requirements	Description	References
Product data*	Include product data and description	[102, 97]
Technological information	Include production processes and description	[102, 97]
Planning	Demands of production planning and scheduling based on orders	[81, 100]
Process	List of operations, set of alternatives, parallel/linear processes	[82, 102, 97, 94]
Setups	Tool selection, clamping solutions and other grippers	[81]
Positioning	Geometrical and positioning information about specific features	[81, 82]
CNC cycles	Relevant information to build CNC cycle	[81, 82]

* Information will be collected from a different Product model

3.3 Data model for process

In Section 3.2 we listed the requirements of the model both by an information attributes and from a modeling content point of view. In this Section we present a description of the data model that will capture and share the information about the process steps needed to manufacture a product between parties in a company (at all levels, from the shop floor to the managerial level). This is achieved in a product-driven environment, where intelligent and proactive products drive the production choices and enable autonomy in manufacturing.

The requirements that the model needs to address are broadly identified by the following points.

- What steps does a product require to be created (e.g., a Bill of Process)?
- What are the attributes of the process step that must be accounted for (e.g., the parameters, tools, materials required)?
- What order can steps occur in (e.g., Computer-Aided Process Planning)?
- How could external systems (e.g., agents) find this information?

The process model here proposed is based on the STEP standard (ISO 10303). This model follows the product-process-resource (PPR) approach explained by [97] and describes the process in a generic way, according to the first level of the process planning proposed by [92]. Its aim is to describe the tasks required for the realization of a specific product without defining a specific manufacturing process, repeating the information already available on the product model, or targeting a specific manufacturing resource. For this, the proposed structure is shown in Fig. 30, where the Product and Resource can be completely described by their respective models, and only the relationships between them are established for describing the process and its tasks.

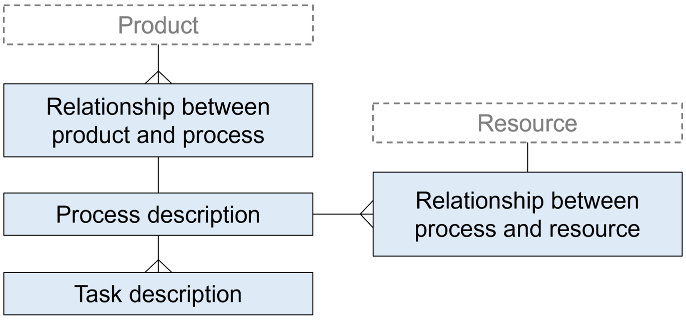


Figure 30: General structure of the proposed process model

The proposed model can be described as follows:

3.3.1 Product

In the context of the STEP, a product can be defined as “a thing or substance or information produced by a process” [62]. Then, using the entities in ISO 10303-41 [104], the physical product is represented by a *product* entity, which can have multiple definitions collected by a *product_definition_formation*. One *product_definition* can be used to define the product from a specific perspective, in this case, manufacturing. It is important to highlight that in this portion of the model, the physical product is completely described by its own model, only a product definition is added for representing the manufacturing perspective or manufacturing project, which can be considered as a product itself.

3.3.2 Relationship between product and process

Once the product is defined, it needs to be associated with a process for its realization. This relationship can be established by using the *product_definition_process*, for specifying the process, and the *process_product_association*, for creating the relationship between them; both entities are part of the ISO 10303-49 [67].

Since the *product_definition_process* in STEP is a type of action, it describes an activity or the result of an activity, e.g., the product manufacture, the assembly, or the product machining.

3.3.3 Process description

An action requires *action_method* entities for its definition, which are described in the ISO 10303-41 [104]. In this case, the *action_method* characterizing the *product_definition_process* will represent the process plan and be used as a base for the several tasks that the process plan could demand.

To specify each task composing the process plan, another *action_method* entity will be used. The relationship between the task and the process plan can be given by its position in the sequence described by the plan. Then, this relationship can be modeled by using an *action_method_relationship* entity, in which the task will depend on the process plan, i.e., the *action_method* representing the task will be the related method, while the relating method will be the one representing the process plan. Another *action_method*, accompanied by another *action_method_relationship* entity, can be used for a more specific definition of the task.

This part of the standard also defines sub-types of *action_method_relationship* that allow the representation of serial (*serial_action_method*), sequential (*sequential_method*) and parallel (*concurrent_action_method*) tasks in a process. In the case that alternative tasks need to be described in the process plan, an intermediary *action_method* entity can be used for relating each of the alternative tasks to an specific position in the process sequence.

Additionally, since a process plan can be represented by an *action_method*, several process plans could be nested into a main one through the *action_method_relationship* entity.

3.3.4 Task description

In a process plan each task is described by its own parameters which will vary depending on its nature. Then, a generic way of modeling these parameters could be by using the process properties described in the ISO 10303-49 [67]. Each *action_method* (task) can be characterized by one or several *action_property* entities (parameters), that can aid in defining its behavior, required capabilities, or performance. To represent a parameter, an *action_property_representation* can be used to link it to its correspondent representation (parameter value).

To model the parameter values, the representation structures defined by the ISO 10303-43 [105] and other Integrated Resources can be employed. A representation entity will link a numerical value, e.g., a *value_representation_item* or a *value_range* with its appropriate context (e.g., *global_unit_assigned_context*) and, subsequently, its unit (e.g., *si_unit*).

3.3.5 Relationship between process and resource

Finally, each *action_method* entity can be used for establishing an *action_resource_requirement*, i.e., a characteristic that a resource must posses to perform the *action_method*, which is defined in the ISO 10303-49 [67]. This characteristic can be linked to one or more processes and be related to a generic class or group by using the *resource_requirement_type*. In this way. The process will define the characteristics of the required resource without limiting the choice to a specific resource.

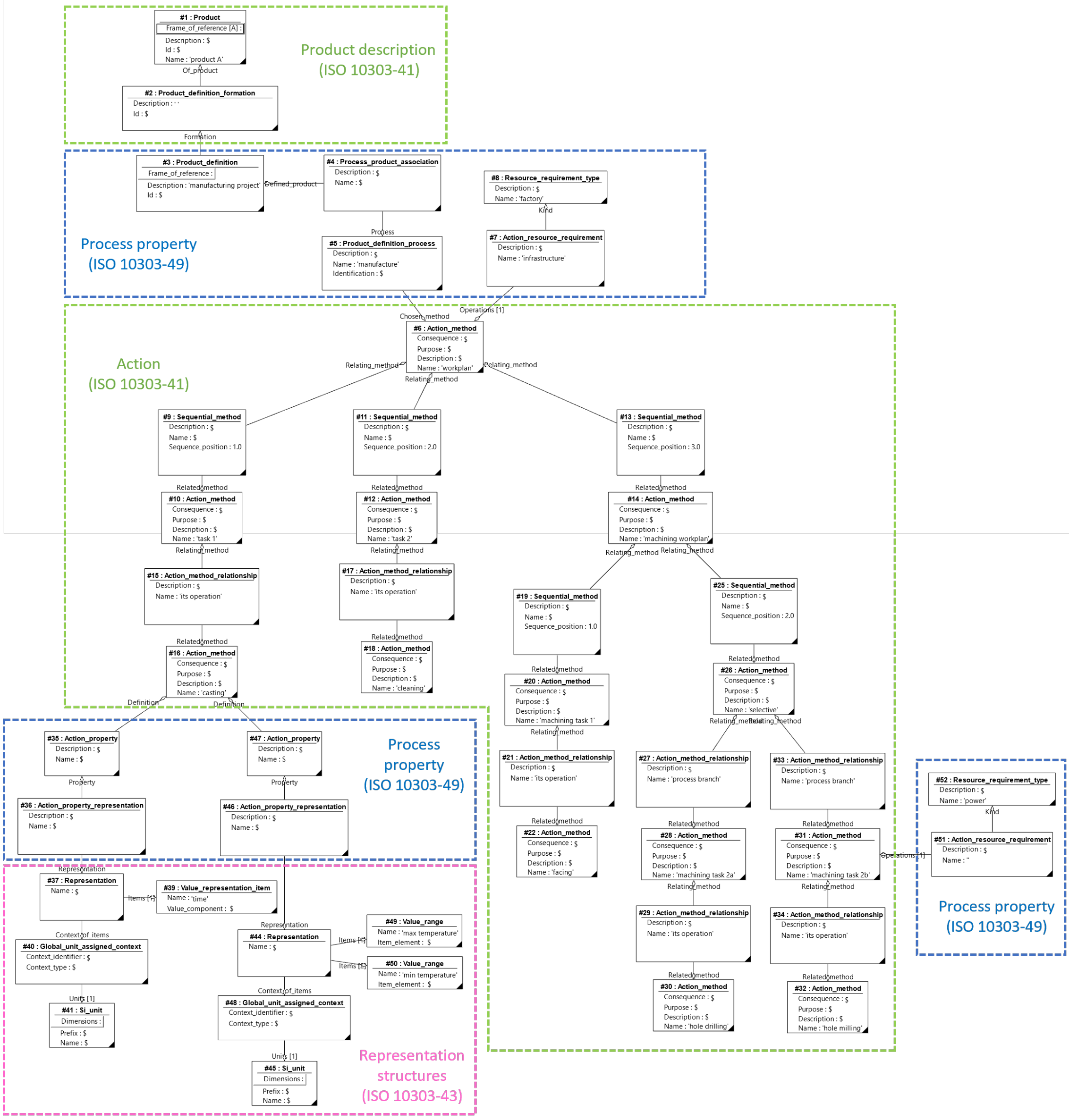


Figure 31: Example model structure

This proposed model tries to keep a generic representation of the process such that it can be adapted to a variety of manufacturing processes, as well as process plans with different level of complexity. An example of this structure is shown in Fig. 31.

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 Model assessment

This section presents an evaluation of the model presented in Section 3.3, to check if it fulfills the established requirements (see Section 3.2). We evaluated the model against the requirements of Table 3, and report the results of the evaluation in Table 5. Many of the listed requirements are intrinsically fulfilled by the employment of the STEP standard to build the model, other requirements need more testing in future applications of the model, and finally some requirements need to be evaluated again after the development of the next Work Package in this Project. The only requirement that has not yet been taken into account is the Safety requirement.

Table 5: Assessment of the requirements of the information content in the model

Requirement	Evaluation
Consistency	Yes, needs testing
Accuracy	Yes, needs testing
Completeness	Yes, transferred data is complete for process
Generality	Yes, the model is general enough to comprehend various processes
Shareability	Yes, the model uses STEP standard, which is shareable
Integration (1)	Yes, refer to STEP standard
Integration (2)	Is fulfilled if model is shared through value chain and in-house
Semantics	Will depend on application in following deliverable
Standardization	Yes, with STEP standard
Safety	Has not yet been taken into account in this model
Extensibility	Further study for extensibility is required
Availability	Will depend on the user

Then, in Table 6 the requirements from the modelling perspective are listed along with the entities that are used in the model to specify them. Figure 31 can help understanding and navigate the model. The requirements of Table 3 are fulfilled by the model entities explained in Section 3.3.

As shown, the model not only fulfills the required information, it also can be used to specify further information. Moreover, it is also extensible to as many processes and tasks as needed. Task entities can be defined sequentially, concurrently or in series, and process entities can be nested and paired.

This model only needs inputs from a product definition model, to get information about the product description, and from the resource model, to connect the processes to the respective resources. Therefore, Product and Resource should be defined in the respective models, only the relationships between them and the presented process model are needed and establish the full PPR information flow.

Table 6: Assessment of the requirements from Section 3.2 with the model of Section 3.3

Requirement	Entity	Notes
Product data	*product*, *product_definition_formation*	The product is described by its own model
Technological information	*action_method*, *action_method_relationship*	Actions and tasks are defined
Planning	*action_method*, *product_definition_process*, *action_method_relationship*	Represent one or several nested process plans
Process	*action_method_relationship*, *serial_action_method*, *sequential_method*, *concurrent_action_method*	Sequence of tasks and alternatives from position in the plan
Setups	*action_property*, *action_resource_requirement*	Resource characteristic to perform task
Positioning	*action_property*, *action_property_representation*, *value_representation_item*, *si_unit*	Define task behavior, capabilities, performance and representation with parameter values
CNC cycles*	*action_resource_requirement*, *resource_requirement_type*	Define resource info without limiting to specifics

[*] Not limited to CNC cycles

[**] Resources: ISO 10303-41 [104], ISO 10303-43 [105], ISO 10303-49 [67]

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 Acknowledgments and References

The authors would like to thank Dr. Mikael Hedlind, whose invaluable knowledge and collaboration essentially contributed to the realization of the production process model and its relevant entities and features.

References

[1] DECISION Etudes Conseil and IPC. *Digital Directions, Greener Connections: An Industrial Policy Report on European Electronics Manufacturing.* Tech. rep. Apr. 2021.

[2] Herbert Dawid et al. “Management science in the era of smart consumer products: challenges and research perspectives”. In: *Central European Journal of Operations Research* 25.1 (2017), pp. 203–230.

[3] Thomas Wasley et al. “Enabling rapid production and mass customisation of electronics using digitally driven hybrid additive manufacturing techniques”. In: *2016 IEEE 66th Electronic Components and Technology Conference (ECTC).* IEEE. 2016, pp. 849–856.

[4] Konstantinos Efthymiou et al. “Manufacturing systems complexity analysis methods review”. In: *International Journal of Computer Integrated Manufacturing* 29.9 (2016), pp. 1025–1044.

[5] A. Grochowski et al. “Integrated circuit testing for quality assurance in manufacturing: history, current status, and future trends”. In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 44.8 (1997), pp. 610–633. DOI: `10.1109/82.618036`.

[6] Paulo Leitão and Francisco Restivo. “ADACOR: A holonic architecture for agile and adaptive manufacturing control”. In: *Computers in industry* 57.2 (2006), pp. 121–130.

[7] Paulo Leitão. “Agent-based distributed manufacturing control: A state-of-the-art survey”. In: *Engineering applications of artificial intelligence* 22.7 (2009), pp. 979–991.

[8] Hendrik Van Brussel et al. “Reference architecture for holonic manufacturing systems: PROSA”. In: *Computers in Industry* 37.3 (1998), pp. 255–274. ISSN: 0166-3615. DOI: `https://doi.org/10.1016/S0166-3615(98)00102-X`. URL: `https://www.sciencedirect.com/science/article/pii/S016636159800102X`.

[9] Corrado Poli. *Design for manufacturing: a structured approach.* Butterworth-Heinemann, 2001.

[10] Oliver Fisher et al. “Cloud manufacturing as a sustainable process manufacturing route”. In: *Journal of manufacturing systems* 47 (2018), pp. 53–68.

[11] Thierry Rayna, Ludmila Striukova, and John Darlington. “Co-creation and user innovation: The role of online 3D printing platforms”. In: *Journal of Engineering and Technology Management* 37 (2015), pp. 90–102.

[12] Julia Siderska and Khair S Jadaan. “Cloud manufacturing: a service-oriented manufacturing paradigm. A review paper”. In: *Engineering Management in Production and Services* 10.1 (2018).

[13] Duncan McFarlane et al. “Auto ID systems and intelligent manufacturing control”. In: *Engineering Applications of Artificial Intelligence* 16.4 (2003), pp. 365–376.

[14] AS More, SK Ugale, and DB Unawane. “Overall Equipment Effectiveness”. In: *International Journal of Advance Research in Science and Engineering* ().

[15] EA Nayak et al. "Evaluation of OEE in a continuous process industry on an insulation line in a cable manufacturing unit". In: *International Journal of Innovative Research in Science, Engineering and Technology* 2.5 (2013).

[16] P Stavropoulos et al. "Monitoring and control of manufacturing processes: A review". In: *Procedia CIRP* 8 (2013), pp. 421–425.

[17] Zude Zhou et al. "Condition monitoring towards energy-efficient manufacturing: a review". In: *The International Journal of Advanced Manufacturing Technology* 91.9 (2017), pp. 3395–3415.

[18] Jianhui Mao and Liqian Chen. "Runtime monitoring for cyber-physical systems: A case study of cooperative adaptive cruise control". In: *2012 Second International Conference on Intelligent System Design and Engineering Application*. IEEE. 2012, pp. 509–515.

[19] Hans Fleischmann et al. "Distributed condition monitoring systems in electric drives manufacturing". In: *2016 6th International Electric Drives Production Conference (EDPC)*. IEEE. 2016, pp. 52–57.

[20] Andrew D Kenyon et al. "An agent-based implementation of hidden Markov models for gas turbine condition monitoring". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44.2 (2013), pp. 186–195.

[21] Luis Jimenez et al. *D3.1: CPS State-of-the-art analysis and requirements specification*. The Digital Manufacturing and Design (DiManD) project, 2021.

[22] Yuqian Lu, Xun Xu, and Lihui Wang. "Smart manufacturing process and system automation – A critical review of the standards and envisioned scenarios". In: *Journal of Manufacturing Systems* 56 (2020), pp. 312–325. ISSN: 0278-6125. DOI: `https://doi.org/10.1016/j.jmsy.2020.06.010`. URL: `https://www.sciencedirect.com/science/article/pii/S027861252030100X`.

[23] William Z Bernstein et al. "Contextualising manufacturing data for lifecycle decision-making". In: *International journal of product lifecycle management* 10.4 (2017), pp. 326–347.

[24] Chao Liu et al. "A cyber-physical machine tools platform using OPC UA and MTConnect". In: *Journal of Manufacturing Systems* 51 (2019), pp. 61–74.

[25] Pei Lei et al. "MTConnect compliant monitoring for finishing assembly interfaces of large-scale components: A vertical tail section application". In: *Journal of Manufacturing Systems* 45 (2017), pp. 121–134.

[26] MTConnect Institute. *MTConnect Standard Part 1.0 – Overview and Fundamentals Version 1.8.0*. Tech. rep. MTConnect Institute, Sept. 2021. URL: `https://docs.mtconnect.org/MTConnect_Part_1-0-Overview_and_Fundamentals_1-8-0.pdf` (visited on 12/13/2021).

[27] MTConnect Institute. *MTConnect Standard Part 2.0 – Devices Information Model Version 1.8.0*. Tech. rep. MTConnect Institute, Sept. 2021. URL: `https://docs.mtconnect.org/MTConnect_Part_2-0_Devices_Information_Model_1-8-0.pdf` (visited on 12/13/2021).

[28] MTConnect Institute. *MTConnect Standard Part 3.0 – Streams Information Model Version 1.8.0*. Tech. rep. MTConnect Institute, Sept. 2021. URL: `https://docs.mtconnect.org/MTConnect_Part_3-0_Streams_Information_Model_1-8-0.pdf` (visited on 12/13/2021).

[29] Florian Pauker, Thomas Trautner, and Burkhard Kittl. "Advanced MTConnect Asset Management (AMAM)". In: Aug. 2016.

[30] Ben Edrington et al. "Machine monitoring system based on MTConnect technology". In: *Procedia Cirp* 22 (2014), pp. 92–97.

[31] OPC Foundation. *OPC UA for Computerized Numerical Control (CNC) Systems.* Tech. rep. OPC Foundation, July 2017. URL: https://opcfoundation.org/developer-tools/specifications-opc-ua-information-models/opc-unified-architecture-for-cnc-systems.

[32] B Boss et al. "Digital twin and asset administration shell concepts and application in the industrial internet and industrie 4.0". In: *Plattform Industrie* 4 (2020).

[33] *Details of the Administration Shell. Part 1: The exchange of information between partners in the value chain of Industrie 4.0, version 1.0.* Plattform Industrie 4.0. Nov. 2018.

[34] ZVEI Die Elektroindustrie Platform Industrie 4.0. *Structure of the Administration Shell, Continuation of the Development of the Reference Model for the Industrie 4.0 Component.* online flyer. Apr. 2016.

[35] E Marseu, D Kolberg, and S Weyer. "Exemplary transfer of the RAMI 4.0 administration shell to the smartfactory KL system architecture for industrie 4.0 production systems". In: *White paper SF-2.1* 4 (2017).

[36] *BaSyx-eclipse.* URL: https://git.eclipse.org/r/plugins/gitiles/basyx/basyx/.

[37] *NOVAAS.* URL: https://gitlab.com/novaas/catalog/nova-school-of-science-and-technology/novaas.

[38] Giovanni Di Orio, Pedro Maló, and José Barata. "NOVAAS: A Reference Implementation of Industrie4. 0 Asset Administration Shell with best-of-breed practices from IT engineering". In: *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society.* Vol. 1. IEEE. 2019, pp. 5505–5512.

[39] *PyI40AAS.* URL: https://git.rwth-aachen.de/acplt/pyi40aas/-/tree/master.

[40] *BaSyx.* URL: https://app.swaggerhub.com/apis/BaSyx/basyx%5C_asset%5C_administration%5C_shell%5C_http%5C_rest%5C_api/v1.

[41] *SAP AAS.* URL: https://github.com/SAP/i40-aas.

[42] *SAP AAS type 2.* URL: https://github.com/SAP/i40-aas-type2.

[43] *openAAS.* URL: https://github.com/acplt/openAAS.

[44] *ThingWorx Asset Management.* URL: https://support.ptc.com/help/thingworx_hc/thingworx_utilities_7_hc/en/index.html#page/ThingWorx_Utilities/Converge_AssetMonitor.html.

[45] *AASX.* URL: https://github.com/admin-shell-io/aasx-package-explorer.git.

[46] *AAS-based-PnP-Demo.* URL: https://github.com/hanyangyexun/AAS-based-PnP-Demo.

[47] Xun Ye et al. "Toward the Plug-and-Produce Capability for Industry 4.0: An Asset Administration Shell Approach". In: *IEEE Industrial Electronics Magazine* 14.4 (2020), pp. 146–157.

[48] *coreAAS.* URL: https://github.com/OPCUAUniCT/coreAAS.

[49] Salvatore Cavalieri and Marco Giuseppe Salafia. "Asset Administration Shell for PLC Representation Based on IEC 61131–3". In: *IEEE Access* 8 (2020), pp. 142606–142621.

[50] Salvatore Cavalieri and Marco Giuseppe Salafia. "Insights into mapping solutions based on opc ua information model applied to the industry 4.0 asset administration shell". In: *Computers* 9.2 (2020), p. 28.

[51] Salvatore Cavalieri, Salvatore Mulé, and Marco Giuseppe Salafia. "OPC UA-based Asset Administration Shell". In: *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*. Vol. 1. IEEE. 2019, pp. 2982–2989.

[52] *ISO 23247-3:2021 Automation systems and integration-Digital twin framework for manufacturing-Part 3: Digital representation of manufacturing elements*. 2021. URL: `www.iso.org`.

[53] Chao Liu et al. "MTConnect-based cyber-physical machine tool: a case study". In: *Procedia Cirp* 72 (2018), pp. 492–497.

[54] *ISO 22400-1:2014 Automation systems and integration-Key performance indicators (KPIs) for manufacturing operations management: Overview, concepts and terminology*. 2014. URL: `www.iso.org`.

[55] OPC Fundation. *OPC UA 10000-1: Overview and Concepts*. 2017. URL: `https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-1-overview-and-concepts/`.

[56] OPC Foundation. *OPC 40010-1 - Robotics Part 1: Vertical Integration*. Tech. rep. OPC Foundation, June 2019. URL: `https://opcfoundation.org/developer-tools/specifications-opc-ua-information-models/opc-unified-architecture-for-robotics/`.

[57] Ella Roubtsova and Vaughan Michell. "Modelling and validation of KPIs". In: *Proceedings of the BMSD* (2013).

[58] Reiner Anderl and Ing Reiner Anderl. *Industrie 4.0-Advanced Engineering of Smart Products and Smart Production Industrie 4.0-Holistic Approach for IT-security, safety, privacy and knowledge protection View project Enhancing CAD and PDM Education at TU Darmstadt View project Industrie 4.0-Adv*. Tech. rep. 2014. URL: `https://www.researchgate.net/publication/270390939`.

[59] Y Tina Lee et al. "Information modeling: From design to implementation". In: *Proceedings of the second world manufacturing congress*. Citeseer. 1999, pp. 315–321.

[60] Michael J Pratt. "Introduction to ISO 10303-the STEP Standard for Product Data Exchange". In: *Journal of Computing and Information Science in Engineering,* 1.1 (2001), pp. 102–103.

[61] Bradford M Smith. *Product Data Exchange: The PDES Project - Status and Objectives*. Tech. rep. NIST, 1991. URL: `http://www.dtic.mil/docs/citations/ADA261567`.

[62] ISO. *ISO10303-1:2021 Industrial automation systems and integration – Product data representation and exchange – Part 1: Overview and fundamental principles*. 2021. URL: `https://www.iso.org/obp/ui/#iso:std:iso:10303:-1:ed-2:v1:en`.

[63] Sharon J Kemmerer, ed. *STEP: The Grand Experience*. NIST, 1999.

[64] M. S. Bloor. "STEP STandard for the Exchange of Product Model Data". In: *IEE Colloquium on Standards and Practices in Electronic Data Interchange*. IET, 1991.

[65] "ISO 10303, the STEP standard for product data exchange, and its PLM capabilities". In: *Int. J. Product Lifecycle Management* 1.1 (2005), pp. 86–94.

[66] M. Hardwick and D. Loffredo. "Lessons learned implementing STEP-NC AP-238". In: *International Journal of Computer Integrated Manufacturing* 19.6 (Sept. 2006), pp. 523–532. ISSN: 0951192X. DOI: `10.1080/09511920600627170`. URL: `https://www.tandfonline.com/doi/abs/10.1080/09511920600627170%20http://www.tandfonline.com/doi/abs/10.1080/09511920600627170`.

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

[67] ISO. *ISO10303-49 Industrial automation systems and integration - Product data representation and exchange - Part 49: Integrated generic resources: Process structure and properties.* 1998. URL: `https://www.iso.org/standard/25101.html`.

[68] Mary Elizabeth A Algeo, Shaw C Feng, and Steven R Ray. *A State-of-the-Art Survey on Product Design and Process Planning Integration Mechanisms.* Tech. rep. NIST, 1994, p. 27.

[69] ISO. *Industrial Automation Systems and Integration-Product Data Representation and Exchange-Part 224: Application Protocol: Mechanical Product Definition for Process Planning Using Machining Features.* 2000. URL: `https://www.iso.org/standard/36000.html`.

[70] ISO. *ISO 10303-238 Industrial automation systems and integration — Product data representation and exchange — Part 238: Application protocol: Model based integrated manufacturing.* 2020. URL: `https://www.iso.org/standard/72030.html` (visited on 01/30/2022).

[71] X. Xu et al. "STEP-compliant process planning and manufacturing". In: *International Journal of Computer Integrated Manufacturing* 19.6 (Sept. 2006), pp. 491–494. ISSN: 0951-192X. DOI: `10.1080/09511920600669776`. URL: `http://www.tandfonline.com/doi/abs/10.1080/09511920600669776`.

[72] ISO. *ISO 10303-242 Industrial automation systems and integration — Product data representation and exchange — Part 242: Application protocol: Managed model-based 3D engineering.* 2020. URL: `https://www.iso.org/standard/66654.html` (visited on 01/30/2022).

[73] STEP AP242 Project. *Development of STEP AP 242 ed2 "Managed Model Based 3D Engineering".* Tech. rep. 2014, pp. 1–53. URL: `http://www.asd-ssg.org/c/document_library/get_file?uuid=a0cef789-2f27-45f0-9979-201fced16e24&groupId=11317`.

[74] ISO. *ISO 23247-1 Automation systems and integration — Digital twin framework for manufacturing — Part 1: Overview and general principles.* Tech. rep. 2021. URL: `https://www.iso.org/standard/75066.html`.

[75] Yung-Tsun Tina Lee, Frank H. Riddick, and Björn Johan Ingemar Johansson. "Core Manufacturing Simulation Data – a manufacturing simulation integration standard: overview and case studies". In: *International Journal of Computer Integrated Manufacturing* 24.8 (Aug. 2011), pp. 689–709. ISSN: 0951-192X. DOI: `10.1080/0951192X.2011.574154`. URL: `https://www.nist.gov/publications/cmsd-manufacturing-simulation-integration-standard-overview-and-case-studies%20http://www.tandfonline.com/doi/abs/10.1080/0951192X.2011.574154`.

[76] SISO. *Simulation Interoperability Standards Organization (SISO) Standard for : Core Manufacturing Simulation Data – UML Model.* 2010.

[77] L.C. Pouchard et al. "ISO 18629 PSL : A STANDARDISED LANGUAGE FOR SPECIFYING AND EXCHANGING PROCESS INFORMATION". In: *IFAC Proceedings Volumes* 38.1 (2005). 16th IFAC World Congress, pp. 37–45. DOI: `https://doi.org/10.3182/20050703-6-CZ-1902.01525`.

[78] ISO. *ISO 18629-14 Industrial automation systems and integration — Process specification language — Part 14: Resource theories.* Tech. rep. 2006. URL: `https://www.iso.org/standard/35186.html`.

[79] PM Ferreira, SCY Lu, and X Zhu. "A conceptual model of automated process planning for the machined parts domain". In: *Preprints of the 22nd CIRP International Seminar on Manufacturing Systems.* 1990, pp. 11–12.

[80] W. Eversheim and J. Schneewind. "Computer-aided process planning—State of the art and future development". In: *Robotics and Computer-Integrated Manufacturing* 10.1 (1993), pp. 65–70. ISSN: 0736-5845. DOI: `https://doi.org/10.1016/0736-5845(93)90027-H`.

[81] Laszlo Horvath and Imre J. Rudas. "Knowledge based generation of Petri net representation of manufacturing process model entities". In: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*. Vol. 4. IEEE, 1996. DOI: `10.1109/icsmc.1996.561435`.

[82] Laszlo Horvath and I.J. Rudas. "Generating and evaluation of manufacturing process models". In: *IECON '98. Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (Cat. No.98CH36200)*. Vol. 1. IEEE, 1998, pp. 132–136. DOI: `10.1109/IECON.1998.723958`.

[83] Kendra E. Moore and Surendra M. Gupta. "Petri nets modelingMANUFACTURING SYSTEMS MODELING USING PETRI NETS". In: *Encyclopedia of Production and Manufacturing Management*. Ed. by P. M. Swamidass. Boston, MA: Springer US, 2000, pp. 431–435. ISBN: 978-1-4020-0612-8. DOI: `10.1007/1-4020-0612-8_560`.

[84] Reiner Anderl and Dietmar Trippner. *STEP STandard for the Exchange of Product Model Data*. Vieweg+Teubner Verlag, 2000. DOI: `10.1007/978-3-322-89096-2`.

[85] *Home - ISA 95*. URL: `https://isa-95.com/#home` (visited on 01/04/2022).

[86] Arndt Lüder, Lorenz Hundt, and Andreas Keibel. "Description of manufacturing processes using AutomationML". In: *2010 IEEE 15th Conference on Emerging Technologies Factory Automation (ETFA 2010)*. 2010, pp. 1–8. DOI: `10.1109/ETFA.2010.5641346`.

[87] Rainer Drath. *Datenaustausch in der Anlagenplanung mit AutomationML: Integration von CAEX, PLCopen XML und COLLADA*. Springer-Verlag, 2009.

[88] Steven J Fenves. *A core product model for representing design information*. Tech. rep. Gaithersburg, MD: National Institute of Standards and Technology, 2002. DOI: `10.6028/NIST.IR.6736`. URL: `https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir6736.pdf`.

[89] Steven J Fenves et al. *CPM 2: A REVISED CORE PRODUCT MODEL FOR REPRESENTING DESIGN INFORMATION*. Tech. rep. Gaithersburg, MD: National Institute of Standards and Technology, 2004. DOI: `10.6028/NIST.IR.7185`. URL: `https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir7185.pdf`.

[90] Shaw C. Feng and Eugene Y. Song. "A manufacturing process information model for design and process planning integration". In: *Journal of Manufacturing Systems* 22.1 (Jan. 2003), pp. 1–15. ISSN: 02786125. DOI: `10.1016/S0278-6125(03)90001-X`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S027861250390001X`.

[91] ISO/IEC. *ISO 16100-6:2011(E): Industrial Automation Systems and Integration—Manufacturing Software Capability Profiling for Interoperability*. 2015.

[92] X. G. Ming, K. L. Mak, and J. Q. Yan. "A PDES/STEP-based information model for computer-aided process planning". In: *Robotics and Computer-Integrated Manufacturing* 14.5-6 (Oct. 1998), pp. 347–361. ISSN: 07365845. DOI: `10.1016/s0736-5845(98)00023-4`.

[93] Heng Zhang et al. "Development and utilization of a Process-oriented Information Model for sustainable manufacturing". en. In: *Journal of Manufacturing Systems* 37 (Oct. 2015), pp. 459–466. ISSN: 0278-6125. DOI: `10.1016/j.jmsy.2015.05.003`. URL: `https://www.sciencedirect.com/science/article/pii/S0278612515000515` (visited on 12/20/2021).

[94] Hao Tang et al. “A Reconfigurable Method for Intelligent Manufacturing Based on Industrial Cloud and Edge Intelligence”. In: *IEEE Internet of Things Journal* 7.5 (May 2020). Conference Name: IEEE Internet of Things Journal, pp. 4248–4259. ISSN: 2327-4662. DOI: `10.1109/JIOT.2019.2950048`.

[95] Johan Nielssen. “Information modeling of manufacturing processes”. eng. In: (2003). Publisher: Industriell produktion. URL: `http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-3628` (visited on 12/20/2021).

[96] Bin Yang et al. “Manufacturing process information modeling using a metamodeling approach”. In: *The International Journal of Advanced Manufacturing Technology* 94.5-8 (Feb. 2018), pp. 1579–1596. DOI: `10.1007/s00170-016-9979-0`.

[97] Johan Nielsen. “Information modeling of manufacturing processes: Information requirements for process planning in a concurrent engineering environment”. In: *Department of Production Engineering, Royal Institute of Technology* (2003).

[98] Kurt Kosanke. “ISO Standards for Interoperability: a comparison”. In: *Interoperability of enterprise software and applications*. Springer, 2006, pp. 55–64.

[99] Parisa Ghodous and Denis Vandorpe. “An Integrated Product and Process Data Model Using Step”. In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 80494. American Society of Mechanical Engineers. 1996, V001T01A062.

[100] Sebastian Schmied et al. “Information modelling with focus on existing manufacturing systems”. In: *Annual Reviews in Control* 51 (Jan. 2021), pp. 392–400. DOI: `10.1016/J.ARCONTROL.2021.04.010`.

[101] Ahmed Ismail, Hong Linh Truong, and Wolfgang Kastner. “Manufacturing process data analysis pipelines: a requirements analysis and survey”. In: *Journal of Big Data* 6.1 (2019), pp. 1–26. DOI: `10.1186/s40537-018-0162-3`.

[102] Philipp Gölzer, Patrick Cato, and Michael Amberg. “Data Processing Requirements of Industry 4 . 0 –”. In: *European Conference on Information Systems ECIS* 0.March 2019 (2015), pp. 1–13.

[103] Oliver J Fisher et al. “Considerations, challenges and opportunities when developing data-driven models for process manufacturing systems”. In: *Computers & Chemical Engineering* 140 (2020), p. 106881.

[104] ISO. *ISO 10303-41:2019 - Industrial automation systems and integration — Product data representation and exchange — Part 41: Integrated generic resource: Fundamentals of product description and support*. 2019. URL: `https://www.iso.org/standard/78578.html` (visited on 03/23/2022).

[105] ISO. *ISO 10303-43:2018 - Industrial automation systems and integration — Product data representation and exchange — Part 43: Integrated generic resource: Representation structures*. 2018. URL: `https://www.iso.org/standard/76129.html` (visited on 03/23/2022).