Results Paper

The Structure of the Administration Shell:

Trilateral perspectives from France, Italy and Germany

Imprint

Published by

France
Ministry of Economy and Finances
139 rue de Bercy
75012 Paris
www.entreprises.gouv.fr

and

Alliance Industrie du Futur
39-41 rue Louis Blanc
92400 Courbevoie
www.industrie-dufutur.org/contact

Germany
Federal Ministry for Economic Affairs and Energy (BMWi)
Public Relations
10119 Berlin
www.bmwi.de

and

Plattform Industrie 4.0
Bertolt-Brecht-Platz 3
10117 Berlin

In cooperation with
WG reference architectures, standards and norms
in the ZVEI – Zentralverband Elektrotechnik- und
Elektronikindustrie e.V.
(German Electrical and Electronics industry e.V.)

Italy
Ministero dello Sviluppo EconomicoVia Veneto 33
00187, Roma
www.sviluppoeconomico.gov.it/index.php/it/industria40

and

Piano Nazionale Impresa 4.0
Direzione generale per la Politica Industriale,
la Competitività e le Piccole e Medie Imprese
Via Molise, 2- 00187 Roma

Design and production
PRpetuum GmbH, Munich

Status
30 March 2018

Print
BMWi

Illustrations
mattjeacock – iStock (Title)

Approved document for final editing
This document was approved for final editing by the
WG Standardization of the trilateral cooperation
between France, Germany and Italy on 2018 March the 29th.

This publication as well as further publications can be obtained from:
Federal Ministry for Economic Affairs
and Energy (BMWi)
Public Relations
E-mail: publikationen@bundesregierung.de
www.bmwi.de

Central procurement service:
Tel.: +49 30 182722721
Fax: +49 30 18102722721

Contents

1. Preamble

The 4th industrial revolution is radically transforming our economies, as innovation and digitization call for a paradigm shift in industrial production and products. Therefore, integrating the digital revolution is the new road ahead for industry.

France, Germany and Italy as important players in the field of digitization in Europe have started initiatives to keep up and improve their position in the manufacturing industry. Alliance Industrie du Futur in France, Plattform Industrie 4.0 in Germany and Piano Industria 4.0 in Italy have agreed to join forces working on a shared action plan towards internationalization as end to end digital continuity and global standardization are of crucial importance for a digitized economy. Therefore, it is important to promote common, consensus-based standardization activities to support thriving economies in the context of the EU digital single market strategy. Shared characteristics of digital manufacturing related Standards are interoperability, openness, scalability, plug&play mechanisms, and security which support a seamless and easy integration of different IoT-solutions.

The trilateral cooperation between France, Germany and Italy is an integral part of the European Multi-Stakeholder-Platform (EU-MSP) and through the EU-MSP, a coordinated European approach is being formed. Europe is home of worldwide leading and innovative manufacturers in all sizes, who want to maintain and even extend their global leading position through this digital transition.

1.1 Elaboration of this document

This document was created by the working group 1 “Standardization and reference architecture” of the trilateral cooperation between Germany, France and Italy. It is based on the document “Structure of the Administration Shell” published in April 2016 by “Plattform Industrie 4.0” that bundled the technical discussions of the working group “Models and Standards” of the ZVEI (Zentralverband Elektrotechnik- und Elektronikindustrie e.V.).

1.2 Objective of this document

This specification shall serve as a leading picture for the necessary harmonized concepts. Existing and future Standards shall be brought into alignment to these concepts. Detailed IT-compliant specification also need to follow-up, in order to realize dependable implementations.

This specification establishes properties as the roots for the constitution of the Administration Shell and how these can be represented.

The objective is not to create a conclusive IT specification or a definitive specification for the implementation of an individual device or system.

The objective is to:

- establish a consensus between the participating organizations;
- allow other stakeholders to make proposals regarding IT structures and IT services;
- allow other stakeholders to develop “submodels”.

1.3 Editorial notes

The Platforms Alliance Industrie du Futur, Plattform Industrie 4.0 and Piano Nazionale Impresa 4.0 have described use cases, where this document “Structure of an Asset Administration Shell” will be a useful contribution for the implementation.

“Reference Architecture Model Industry 4.0 (RAMI4.0)” (see Annex F) and “Smart Manufacturing Standard Landscape” (see Annex G) have already been distributed as contributions to the ISO/IEC Joint Working Group 21 “Smart Manufacturing Reference Models”.

2. Introduction

2.1 Overview

A data-driven economy knows no barriers and the potential for value creation can only be exploited, if all aspects of industrial production are considered. Therefore, the discussion on Smart Manufacturing can be mainly understood as the interaction between four aspects, as illustrated in Figure 1:

- aspect (1) Horizontal integration through value networks
- aspect (2) Vertical integration, e.g. within a factory/or production shop
- aspect (3) Life cycle management, end-to-end engineering
- aspect (4) Human beings orchestrating the value stream

The Asset Administration Shell provides a flexible framework on the information and functions that can be defined and made available to facilitate and promote the Smart Manufacturing aspects above.

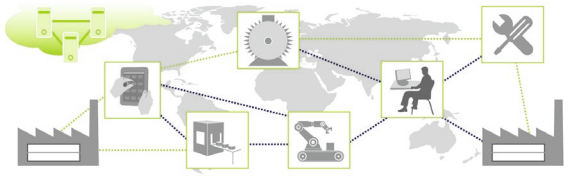
This document addresses all industrial sectors and all types of processes (discrete manufacturing, continous process, hybrid production). It takes into consideration the specificities of different value chains (logistics, procurement, production, outgoing goods, service) and serves the needs of value-added networks involving several partners or even overarching.

Security is a fundamental topic for Smart Manufacturing, a data-driven ecosystem being by nature more exposed to malicious attacks.

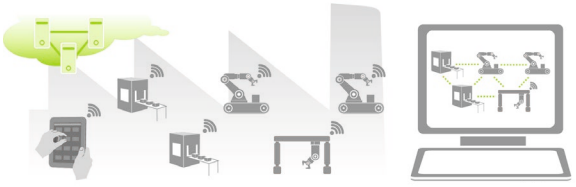
NOTE Security is acheived through guaranteeing the confidentiality, integrity and availability of saved and transferred information. For utilisation of information beyond company boundaries (e.g. through Cloud): pseudonymisation/anonymisation for personal data, cross-company identity and rights management are mandatory. It is also mandatory to guarantee of the confidentiality and integrity of information and functions as well as the maintenance of the availability of the technical functionality and of the functions of the underlying assets.

Figure 1 – Four important aspects of Smart Manufacturing

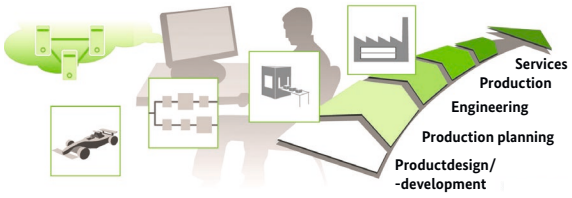
Horizontal integration via value-added networks



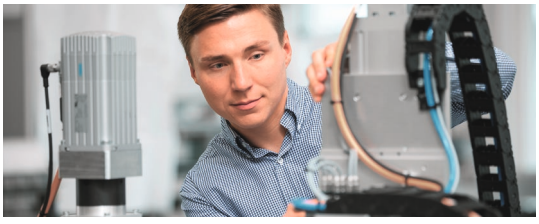
Vertical (integration and networked production system)



Digital consistency for the engineering throughout the whole value-added chain



The human being as a conductor for aded value



Source: Siemens AG/Festo AG&Co KG

2.2 Asset

Standards define “asset” as “physical or logical object owned by or under the custodial duties of an organization, having either a perceived or actual value to the organization” (see the glossary in Annex A).

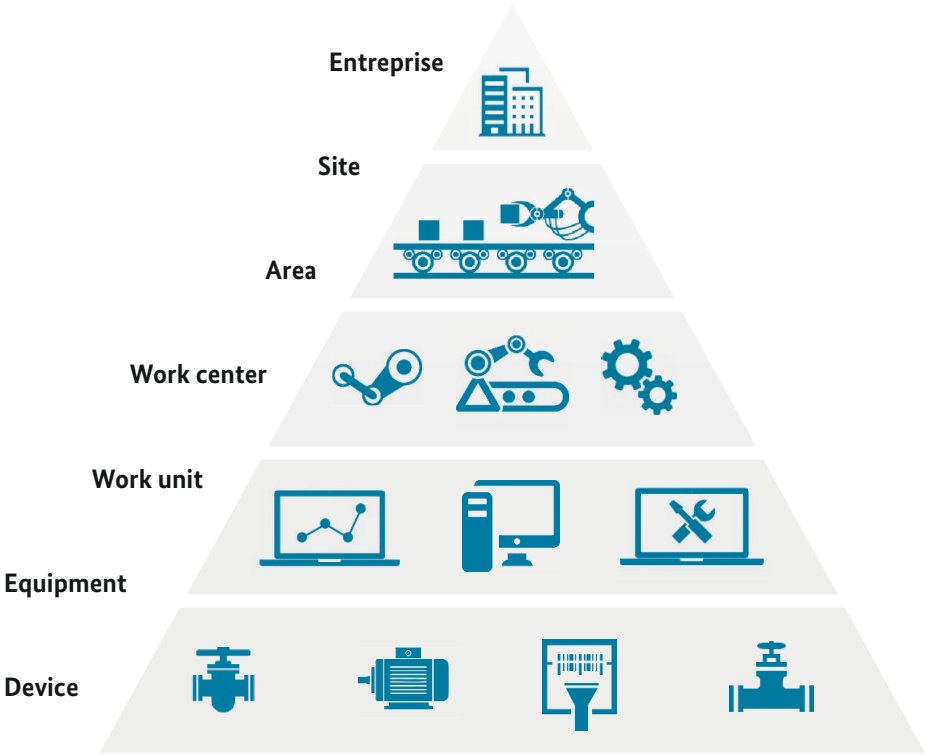
Assets may be material or immaterial, and of various natures such as:

- physical objects, for example, equipment (machine, cabinet, contactor, computer, actuator, cables, connectors, sensors ...), raw material, parts components and pieces (screw, wheel ...), supplies, consumables (paper ...), or products (final or intermediate);
- software (firmware, applications, engineering tools ...);
- documents (data media, life cycle documentation ...)
- immaterial (licence, copyright, idea, plan, process definition, standards, patents, general procedure, recipe, equipment type definition, product/family type definition, production plans, project plans business procedures, actual states ...)
- information
- human (service technician, programmer, operator ...);
- service ...

Depending on these different natures, the aspects of assets to be considered differ (for a piece of equipment, the serial number is relevant; for a person, the gender may be relevant).

In the context of industrial facilities, interoperability is needed at any level of the equipment hierarchy where the production asset is located. In other words, a production

Figure 2 – Equipment hierarchy of an enterprise



asset can be a plant, a machine, a station, an assembly inside a machine, a switch gear, a motor, a tube, etc.
The production asset can be positioned in the equipment hierarchy as shown in Figure 2.

2.3 Interoperability

Interoperability is the key enabler of Smart Manufacturing.

A prominent definition describes interoperability as the ability of two or more objects from the same vendor, or different vendors, to exchange information and use that information for correct cooperation [5].

In other words, two or more components or systems are interoperable if they can perform cooperatively a specific function by using the information that is exchanged. This concept is illustrated in Figure 3.

Assets are used not only in the shop floor but also in the office floor; so, semantics shall be shared on both floors.

Figure 3 – Information exchange in a function

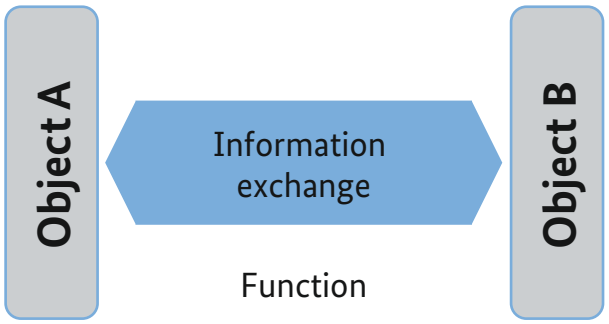
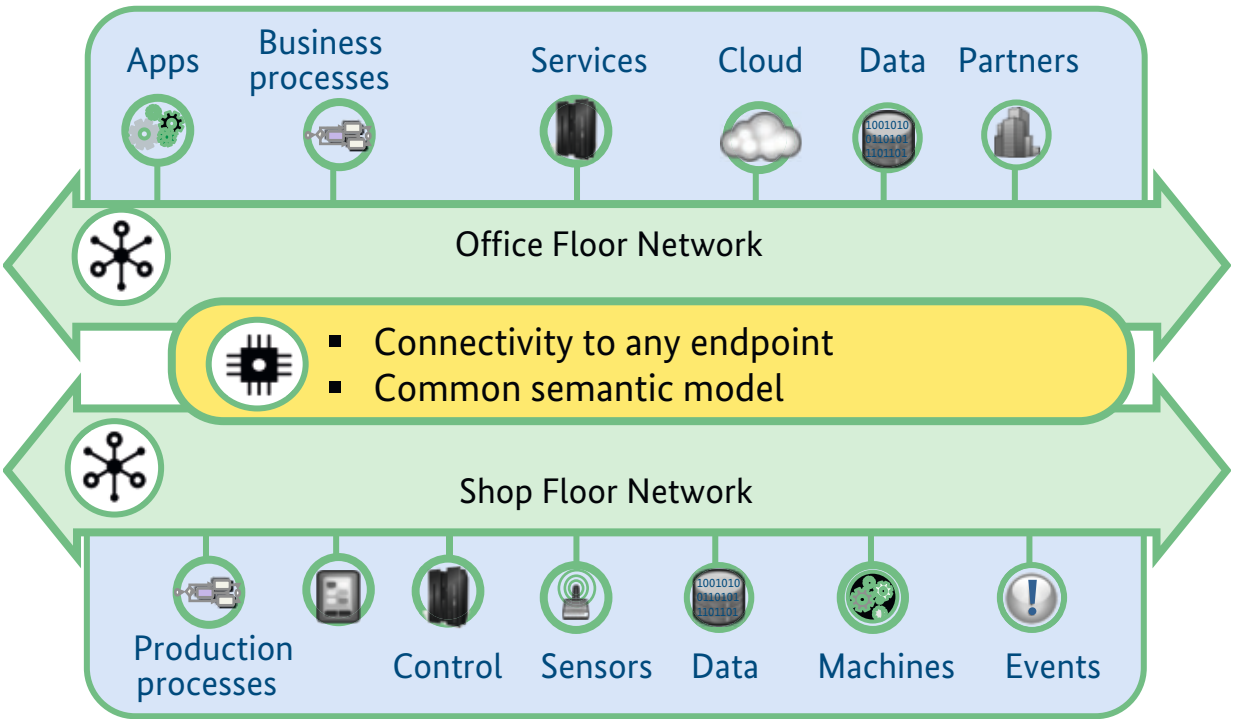


Figure 4 shows the complete enterprise which consists of the shop floor level for production and the office floor level for organization of the company. This is not only the financial and marketing department. It is also the design and development of products that are intended to be manufactured in the shop floor. Both floors work closely together. Therefore, there should be no barriers in terms of semantics otherwise the resulting needed conversions between the semantics of both floors generates costs and disquality (misunterpretation). So, one of the most important subject is a kind of common languages between both floors in order to eliminate the barriers.

Figure 4 – Common semantic in shop floor and office floor



2.4 Administration Shell

The Administration Shell is the standardized digital representation of the asset, corner stone of the interoperability between the applications managing the manufacturing systems.

The Smart Manufacturing component is the combination of the asset and its logical representation, the Administration Shell as illustrated in Figure 5.

The Administration Shell may be the logical representation of a simple component, a machine or a plant at any level of the equipment hierarchy as illustrated in Figure 6.

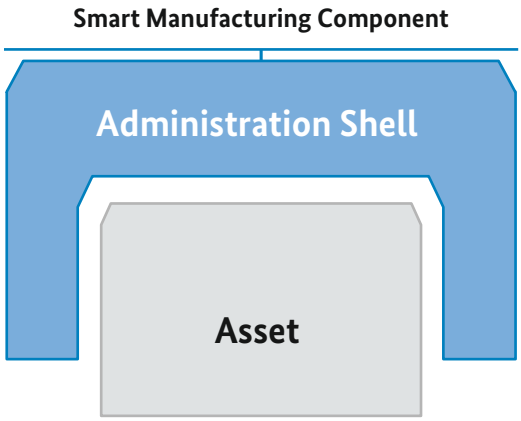
The manufacturer provides the standardized digital representation to his customers, creating both an Asset Administration Shell for the asset type and for asset instance. The system designers, the asset users, the applications, the processes and the asset itself update the information of the Asset Administration Shell during the life of the asset until its disposal.

From the manufacturer point of view the asset is a product. The manufacturer manages different types that have a history with different versions. In parallel, he produces instances of these different types and versions.

These products are incorporated in the production system and as a production asset of an industrial facility as illustrated in Figure 7.

- The product is ordered from the manufacturer and provided to the plant owner or the system integrator for installation, commissioning and use.

Figure 5 – Smart Manufacturing component



- Its Asset Administration Shell will be included in the overall digital descriptions of the manufacturing facility. The type information may be integrated during the plant design.

- The instance properties will start to get valued at the implementation. They will evolve during the rest of the asset life.

The future standard IEC 62890 defines these concepts of product type and instance the asynchronism of their life cycles and the implications to the production system lifecycle.

Figure 6 – Asset from component to plant

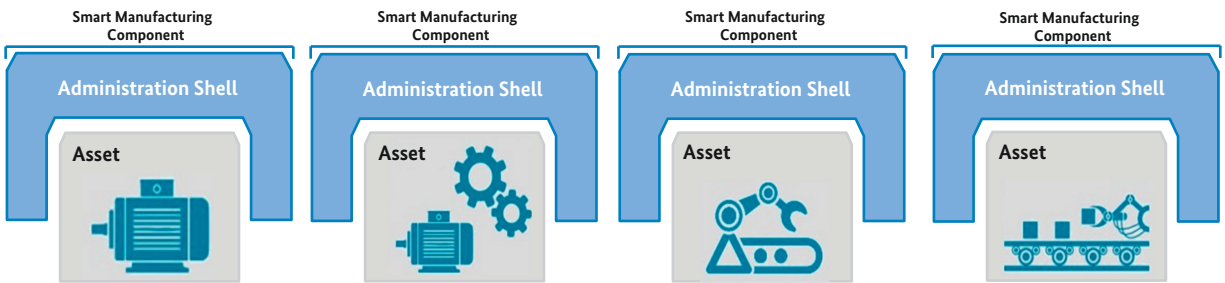
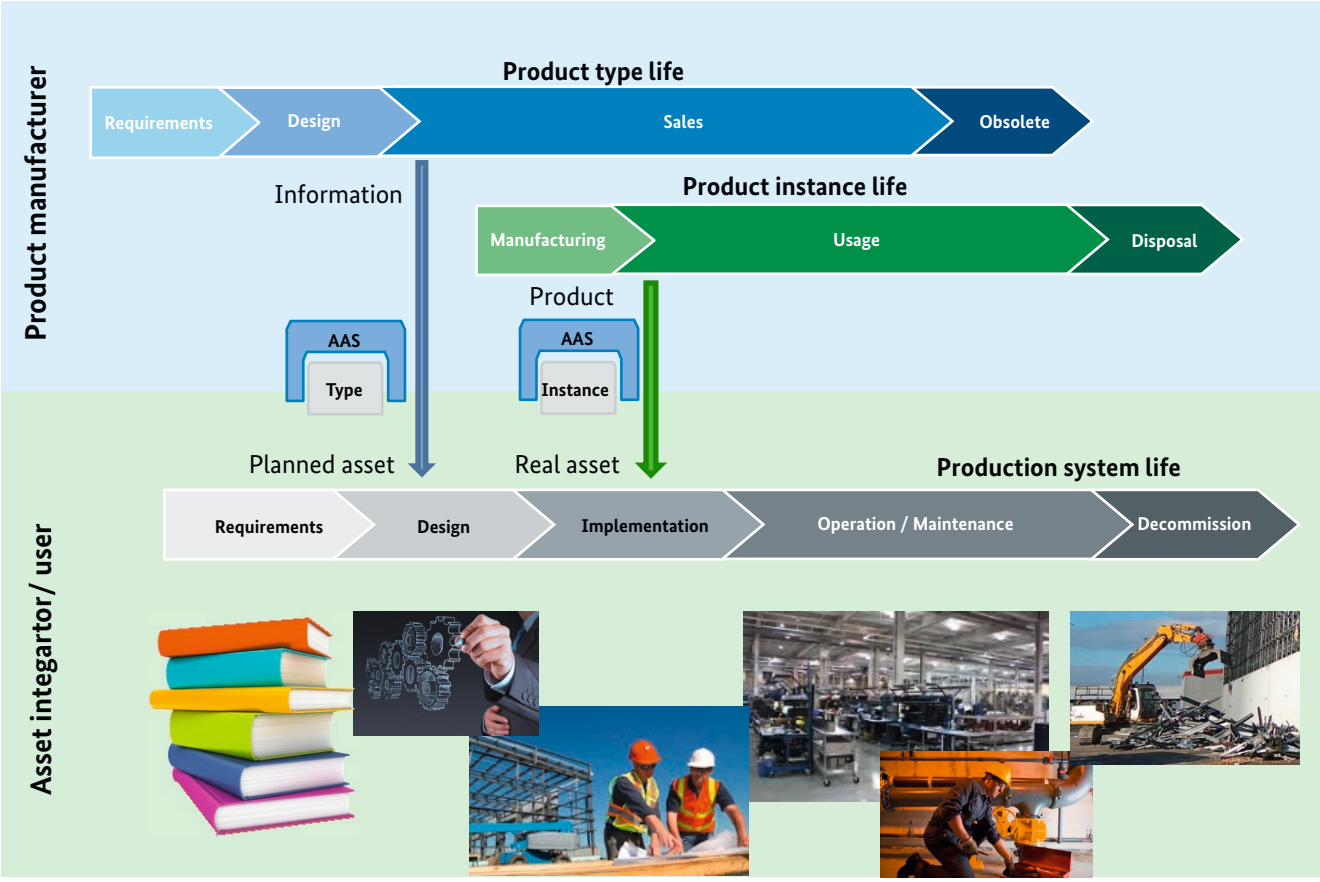


Figure 7 – The Administration Shell in product and sytem life cycles



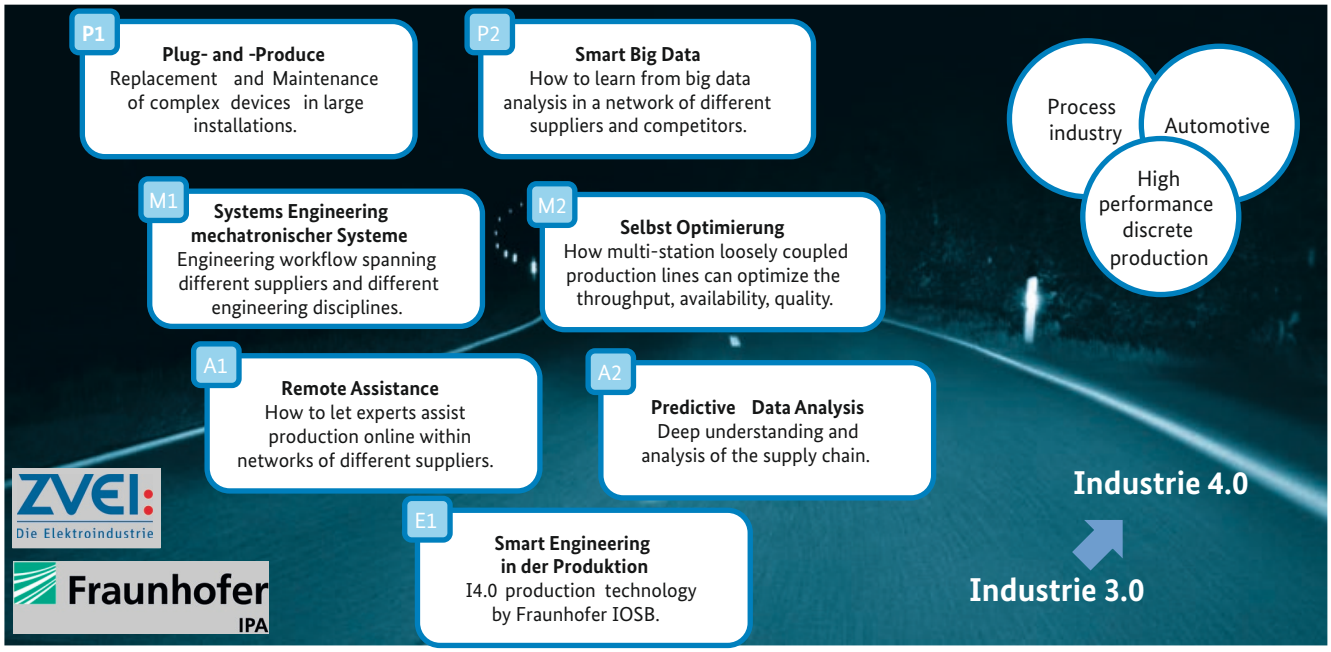
2.5 Evaluation of detailed use cases

The structure of the Administration Shell should be able to support the corresponding use cases of Smart Manufacturing in a suitable manner. Required data, functions and potential security requirements must be identified, and unnecessary additional effort in terms of definitions should be avoided. Some use cases have been defined by ZVEI for example; further use cases have for example already been defined in the publication [4] “Recommendations for implementing the strategic initiative Industrie 4.0”.

2.5.1 Use cases from ZVEI SG “Strategy & Use Cases”

Figure 8 shows the use cases selected by Management Group Industrie 4.0 (SG Strategy and Use Cases) of ZVEI. The description and cross-linking of further use cases are coordinated by the Industrie 4.0 platform and its AG2. The use cases of ZVEI give a detailed description of typical specific applications of Industrie 4.0 in various industries (discrete manufacturing industry, process industry and hybrid production). For each of the use cases several, use cases are currently (August 2015) identified and described in standardised form.

Figure 8 – Use cases of Management Group Industries 4.0 of ZVEI



Source: Festo AG & Co. KG

2.5.2 Use case "self-optimization"

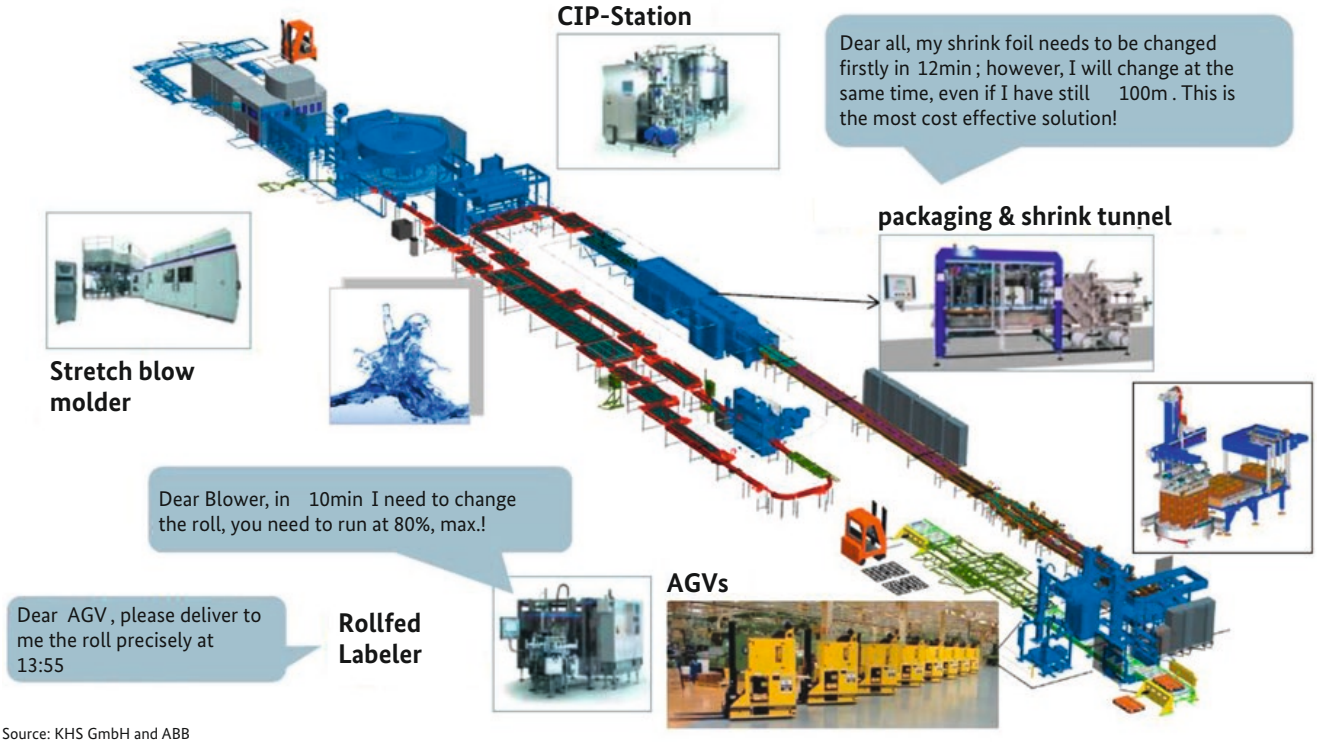
The use case "self-optimization" (also known as "Buffer-free-filling by M2M communication", cf. Plattform I40 AG1/AG2) tackles the problem of misaligned production stations in bottling plants to improve production flexibility and efficiency. The production in existing bottling plants is often directed by a central Manufacturing Execution System that optimizes the production schedule across stations. Involved stations are for example stretch blow molding stations, labelling station, filling station, cleaning-in-place station, packing station, and palletizing stations. Goods are transported between stations using conveyor belts and/or autonomous guided vehicles (AGV).

While in theory the central planning can lead to a globally optimal production schedule, the approach has shown inflexibilities upon certain failure conditions and small batches that require reconfiguration. For example, if a bottle is stuck in a product station essentially stopping the production line, the other production station currently does not react automatically, by for example lowering their product output to avoid larger buffers between stations or by executing scheduled maintenance tasks ahead of schedule during the stoppage. Today, the station operator needs to fix the issue and manually re-configure the production stations.

The buffer-free-filling shall be achieved by moving more intelligence from the MES to the production stations and by increasing the communication among production stations. Each station monitors its own status and provides certain MES KPIs (e.g., utilization rate) to the plant network. Other production stations can subscribe to this status information and draw local conclusions for optimizing their own functionality. For example, if an alarm is raised by a subsequent station in the production line, the current stage can decelerate its production rate to avoid large output buffers. The production stations shall be more aware of their own conditions and learn from past behavior. For example, a production station can predict when its production resources (e.g., labelling paper, bottle raw material) is running out and the autonomously re-order supplies from other stations or transporters. It can prepone scheduled maintenance tasks that require stopping the machines if this suddenly fits into the production schedule.

The use case needs to be completed by assistance systems for service support personnel. It is conceivable to add additional security and safety functions based on the information exchanged between production stations. Supplies can be automatically ordered via an ERP system connected to the Internet.

Figure 9 – Use case buffer-free-filling



Source: KHS GmbH and ABB

If production stations of different vendors shall interact seamlessly, the use of standardized and secure communication protocols, information models, and functional specifications is essential for the implementation of this use case. Candidate standards are OPC UA, ISA-95/88, OMAC PackML, Weihenstephan Standards, and MES KPIs according to ISO 22400. It is however unknown whether these standards are sufficient to implement the use case in a vendor-neutral way.

To support the implementation of this use case, the Administration Shell of Smart Manufacturing Components should be able to carry the (security-) information and provide the services imposed by the previously mentioned standards.

2.6 Examination of various sets of properties

For discussion of the structure of the Administration Shell, it is relevant to determine which sets of properties must be considered, from which sources these originate and how cross-linking between them can be achieved.

2.6.1 Many matters characterise properties even now

International standards and consortia specifications specify classes, features and information that can be portrayed in properties and that are crucial for the concept of Administration Shell. Each of these sources of properties belongs to a certain technical discipline (technical domain) that may constitute a submodel.

The following tables shows some examples:

Examples of standards and specifications providing actual submodels:

Specification/standard Number	Title	Submodel
ecl@ss	ecl@ss classification and product description	Several
IEC 62683	Low-voltage switchgear and control gear • Product data and properties for information exchange	Low-voltage Switchgear and control gear
IEC 61987	Industrial-process measurement and control • Data structures and elements in process equipment catalogues.	Process control field devices

Examples of standards and specifications providing potential sources of submodels:

Specification/standard Number	Title	Submodel
IEC 61784	Industrial communication networks • Profiles	Industrial network
ISO 12100	Safety of machinery • General principles for design • Risk assessment and risk reduction	Safety
ISO 13849-1	Safety of machinery • Safety-related parts of control systems • Part 1: General principles for design	
IEC 62061	Safety of machinery • Functional safety of safety-related electrical, electronic and programmable electronic control systems	
IEC 61511	Functional safety – Safety instrumented systems for the process industry sector	
IEC 61508	Functional safety of electrical/electronic/programmable electronic safety-related systems	
IEC 62443	Industrial communication networks • Network and system security	Security
ISO 20140-5	Automation systems and integration • Evaluating energy efficiency and other factors of manufacturing systems that influence the environment • Part 5: Environmental influence evaluation data	Energy efficiency
65E/482/NP	Industrial-process measurement, control and automation – Uniform representation of condition monitoring functions	Condition monitoring
IEC 61915	Low-voltage switchgear and controlgear • Device profiles for networked industrial devices	Low-voltage Switchgear and controlgear
IEC 61800-7	Adjustable speed electrical power drive systems • Generic interface and use of profiles for power drive systems	Motion & drives

Examples of standards and specifications providing reference models for submodels:

Specification/standard		Submodel
Number	Title	
IEC 62714	Engineering data exchange format for use in industrial automation systems engineering • Automation markup language (AutomationML)	Administration Shell
IEC 62541	OPC Unified Architecture	
IEC TS 62832	Industrial-process measurement, control and automation • Digital Factory framework	
IEC 61360	Standard data elements types with associated classification scheme for electric items	
ISO 13584-42	Industrial automation systems and integration • Parts library • Part 42: Description methodology: Methodology for structuring parts families	
ISO/IEC 6523	Information technology • Structure for the identification of organizations and organization parts	Identification
ISO/IEC 11179	Information technology • Metadata registries (MDR)	
ISO TS 29002-5	Industrial automation systems and integration • Exchange of characteristic data • Part 5: Identification scheme	
IEC 61804	Function blocks (FB) for process control (EDD and EDDL)	Configuration
IEC 62453	Field device tool (FDT) interface specification	
ISO 15745	Industrial automation systems and integration • Open systems application integration framework	
IEC CDV 62890	Life-cycle management for systems and products used in industrial-process measurement, control and automation	Life cycle

Requirement # 1

The Administration Shell shall accept properties from different technical domains in mutually distinct submodels that can be version-controlled and maintained independently of each other.

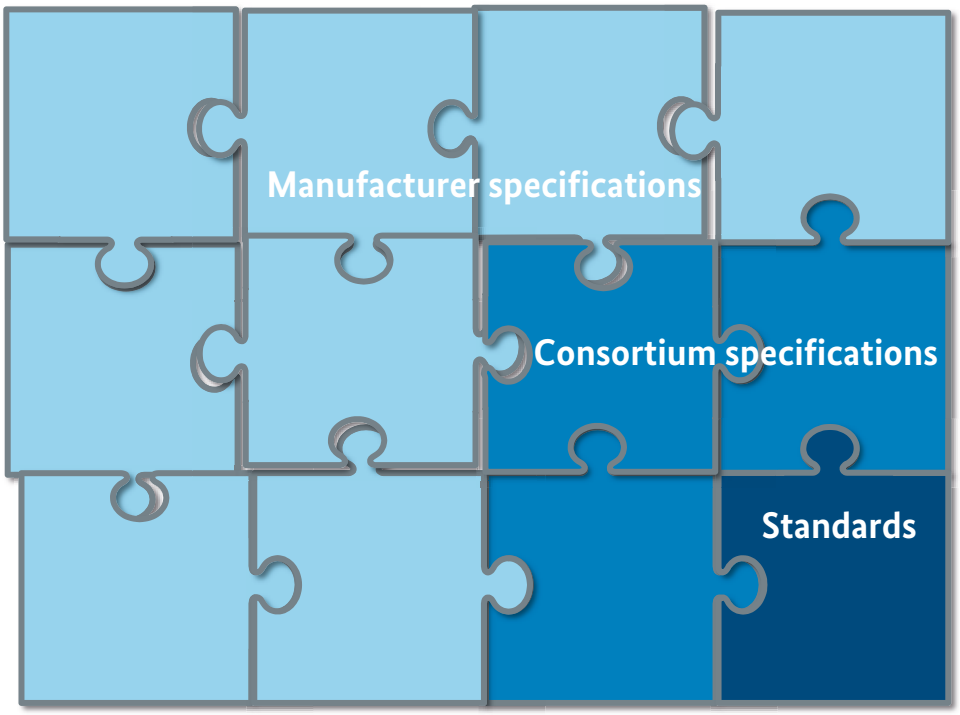
2.6.2 Addressing different sets of properties

In accordance with the Smart Manufacturing requirements and the required suitability for a wide range of use cases, an Administration Shell shall be capable of holding information and functions for different technical disciplines, different life cycle phases and different application and analysis scenarios. The quantities of different properties to support should be correspondingly high and broad-based.

Properties may be defined from three different categories of specifications and standards:

- Standards
- Consortium specifications
- Manufacturer specifications.

Figure 10 – Different sets of property definitions



They are characterized below.

Standards:

Standards are designed and published by national or international standards developing organizations (SDOs) such as ISO, IEC, AFNOR, DIN, DKE, UNI, CEI, ANSI, ISA ... Participation in the design of standards is opened to any stakeholder in his country generally not free of charge. Access to the publications is possible to any individual or organization generally for a fee.

The typical lead time for creating or amending a standard is three years.

Standardized properties are published in different domains under the responsibility of the relevant technical committees of these SDOs. They might be limited to a reduced set. In IEC, access to the property data base (Component Data Dictionary "IEC CDD") is free of charge.

The quality of the output is guaranteed by the participation of the best worldwide experts and a sharp process involving the choice of the stakeholders of participating national bodies.

The IEC CDD is a standard in data base format managed by the change request procedure, which reduces the duration of amendments and can address a unique property".

Consortium specifications:

Consortium specifications are designed and published by associations of stakeholders. Participation in the consortium is reserved to stakeholders agreed by the board against a participation fee. Access to the publications is reserved to the members although some documents may be accessible freely.

For classes and properties, the corresponding organizations are called "product classification organization" such as eCl@ss, ETIM, GS1, ECCMA, UNSPSC, idea, Jeita, ECCENTER.

Due a more limited number of stakeholders and simplified adoption process, the consensus is quicker than for standardization.

The coverage of domains is wide. For example eCl@ss covers nearly 700 domains while IEC CDD only 3.

Manufacturer specifications:

Manufacturers own many private properties; more can be formed rapidly and can also address specific asset description.

Private classes and properties for specific assets that cannot be developed in conjunction with other stakeholders may be defined as part of a manufacturer dictionary, and exposed by some product classification organization such as ECCMA.

None of these sets of property can be excluded a priori for the Administration Shell; by contrast it must be taken into account:

Requirement # 2

The Administration Shell should be capable of including properties from a wide range of technical domains and of identify which domain they derive from.

Requirement # 3

For finding definitions within each relevant technical domain, different procedural models should be allowed that respectively meet the requirements of standards, consortium specifications and manufacturer specifications sets.

3. Structure of the Administration Shell

3.1 General structure of the Administration Shell

Applying the requirements (see 3.2 and 3.3), a general structure for the Administration Shell can be developed. It should be based on established concepts in both automation and ICT technologies and be equipped for future developments regarding the relevant Smart Manufacturing aspects (Horizontal Integration, Vertical Integration, Integrated Engineering and Interaction with Humans). An ICT-compliant specification needs to be developped additionally.

As shown in Figure 11, the Administration Shell is structured in **"header"** and **"body"**.

The **"header"** provides identification and designation of the asset(s) and of the Administration Shell.

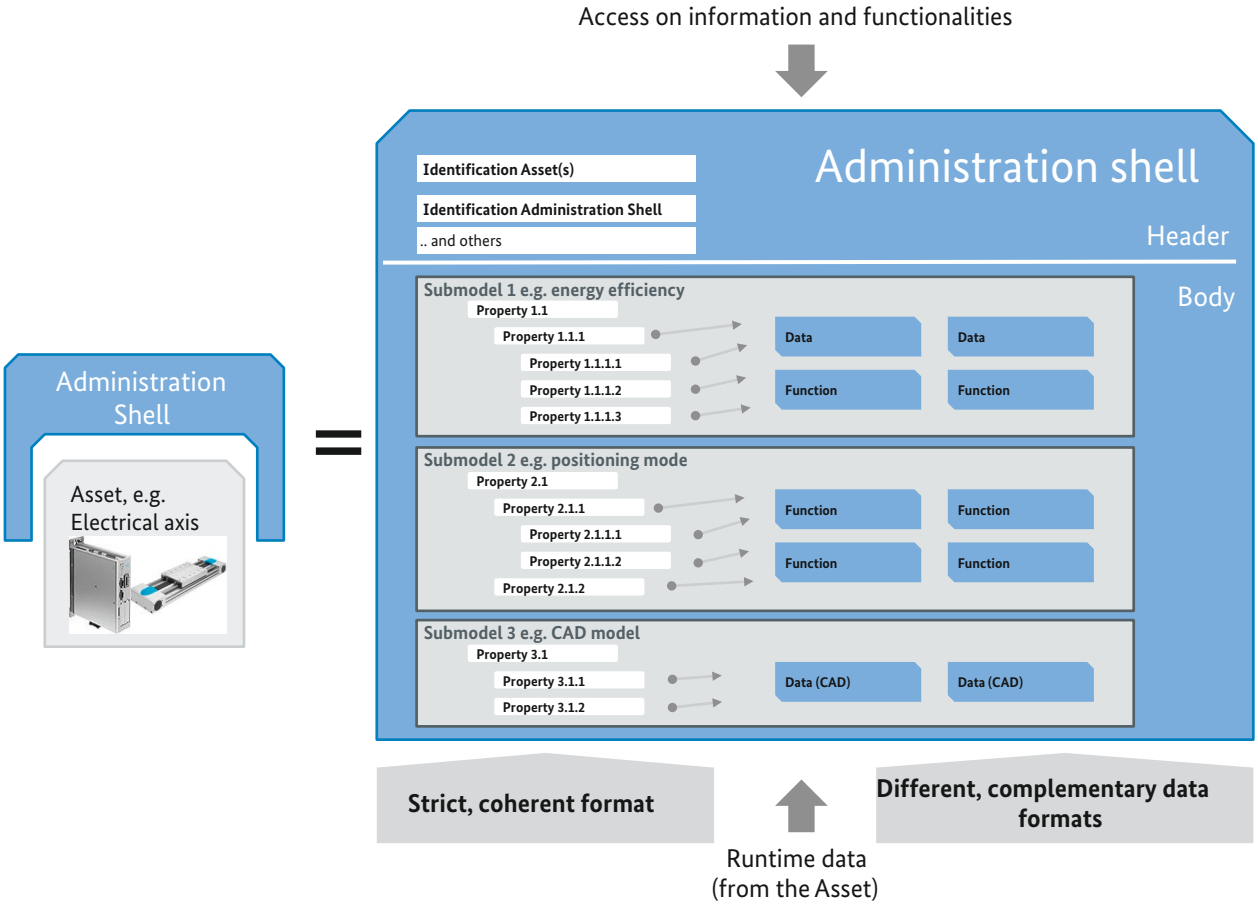
The **"body"** administers individual submodels within the Administration Shell. Each submodel has hierarchically organised properties, which refer to individual information and functions (depicted by the blue geometric elements in Figure 11). The mutual referencing and the creation of views is possible but not shown.

The header and body have to be safeguarded by provisions for information security adequate to the industrial application of the involved assets.

The respective property structures are arranged in a strict, standardised format (based on IEC 61360, see B.1), while for the different information and functions various complementary data formats and methods of access are possible.

The Administration Shell can record and depict runtime information from the asset (e.g. the actual position and actual currents for a servo amplifier).

Figure 11 – General structure of an Administration Shell



A service oriented API (Application Programmer's Interface) should make the services of the Administration Shell available externally, including:

- lifelong maintenance of the properties,
- information and functions within the Administration Shell,
- addressing and identification of Administration Shells and assets,
- efficient search mechanism for properties and referenced information and functions.

3.2 Interlinking between different Administration Shells

- An asset can have several Administration Shells

Example 1 (see Figure 12)

The manufacturer of a servo amplifier can maintain an Administration Shell under "Type/Development" (see Annex F) for his internal purposes, and file his internal development information there.

For the purposes of its customers this manufacturer can provide an external Administration Shell of the model series under "Type/Usage".

For each instance delivered, the respective user can derive and continue to maintain an Administration Shell under "Instance/Usage".

Example 2 (see Figure 13)

When a system integrator designs a production system for a customer in the phase "Design" (see Annex G), he creates an Administration Shell for this complex asset. He keeps it in his files the status after the "Implementation".

During "Usage", the customer manages an Administration Shell. At the same time, an external service provider also manages a local Administration Shell for maintenance. The mutual update of the user's and partner's Administration Shells shall be considered.

The above examples raise the possibility of a requirement for automatic referencing and matching of individual elements of Administration Shells between themselves, for example by being able to execute updating of an Administration Shell of a "Type" into an "Instance" (see Requirement # 16).

Figure 12 – Administration Shells of manufacturer and customer

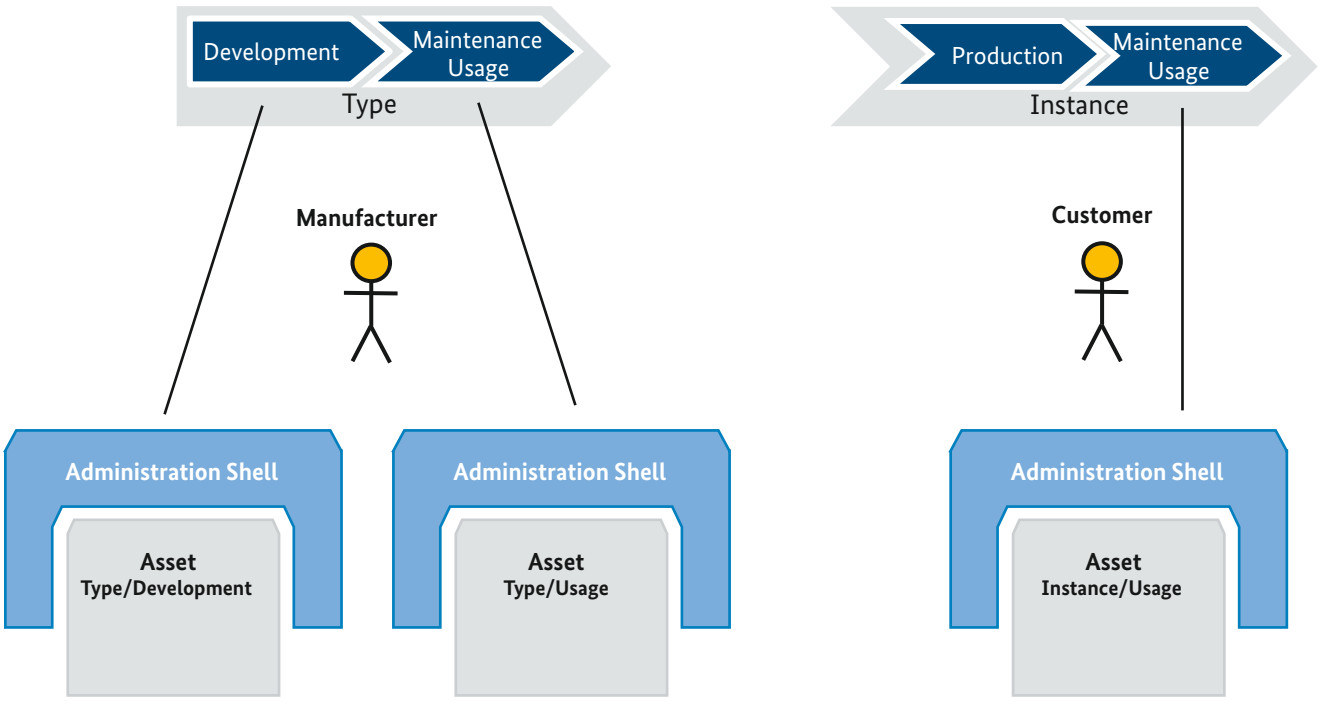
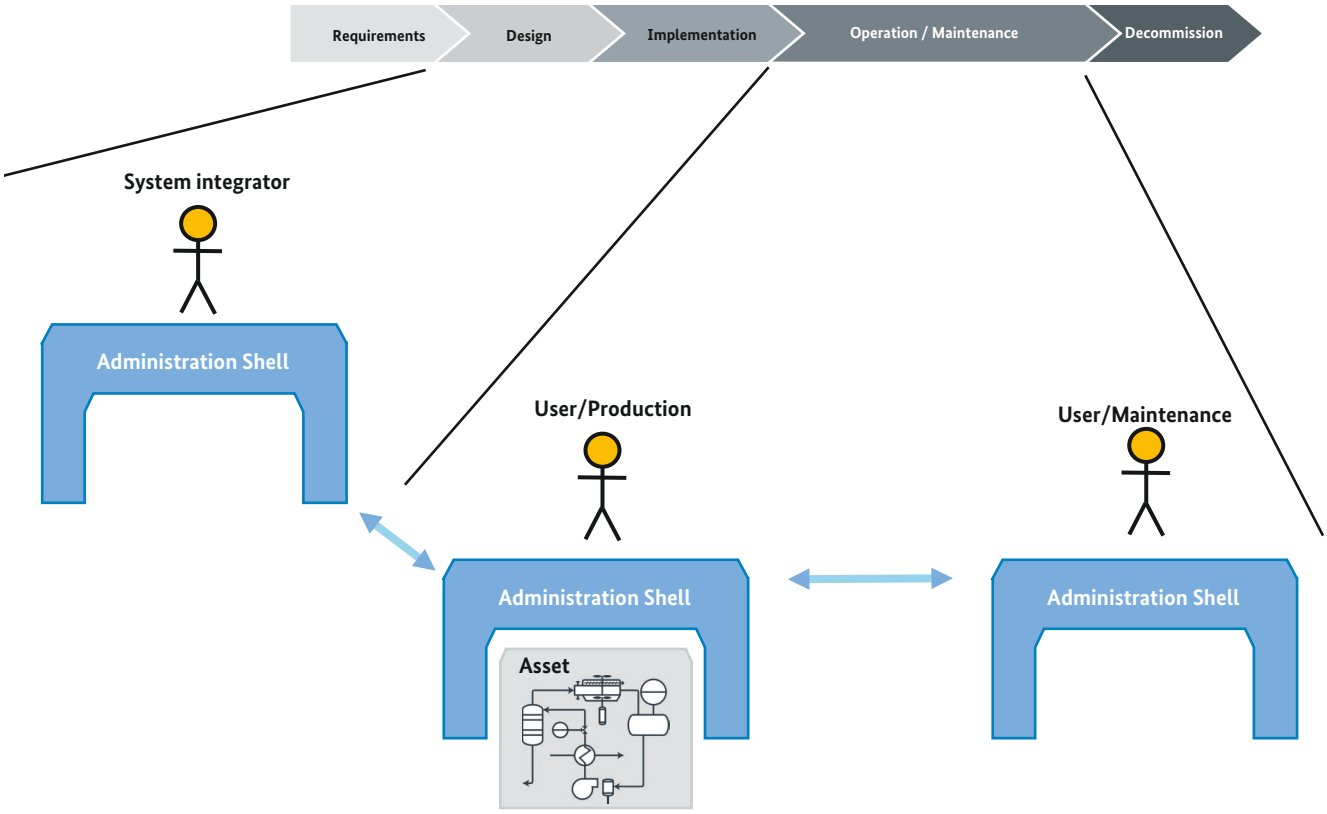


Figure 13 – Administration Shells of complex asset



Requirement # 4

Different Administration Shells in respect of an asset must be capable of referencing each other.
In particular, elements of an Administration Shell should be able to play the role of a "copy" of the corresponding components from another Administration Shell.

- **One or more assets can be portrayed in an Administration Shell**

Example (see Figure 14)

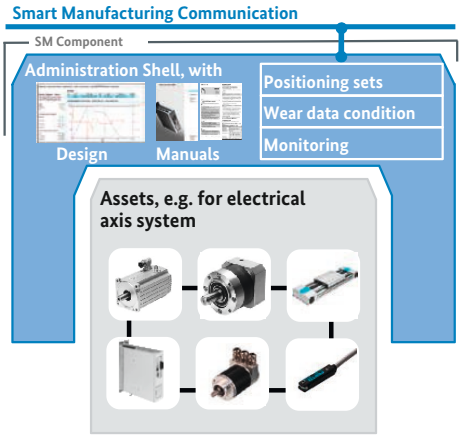
A mechanical axis, motor, servo amplifier and additional assets constitute an "encapsulate-capable" Smart Manufacturing Component.

The Administration Shells of several individual assets that a manufacturer brings onto the market individually, are consolidated into one Administration Shell, if this manufacturer also sells a whole axis system.

Requirement # 5

Individual Administration Shells should, while retaining their structure, be combined into an overall Administration Shell.

Figure 14 – Administration Shells of an electrical axis system



3.3 Identifiers

3.3.1 Needs

Various requirements demand the availability and clarity of identifiers (IDs) for Administration Shells, properties, asset instances, asset types and relationships.

Discussion of different aspects of the provision of services (production, logistics and operation), of different industry sectors and different regions suggests that exclusive commitment to a system of identification that must also offer global uniqueness is difficult to achieve. Instead identifiers should be found that indirectly allow association of further identifiers (dereferencing), which therefore allows the number of variants accepted for Smart Manufacturing to be kept small.

Requirement # 6

Identification of assets, Administration Shells, properties and relationships shall be achieved using a limited set of identifiers (IRDI, URI and GUID), providing as far as possible offer global uniqueness.

3.3.2 Determination of Identifiers

Thus, for the following international and freely available standards and specifications, appropriate variants of identifiers are determined:

- **Global identifiers** are recognised as allowing the interaction of the Administration Shell or its elements with the Administration Shells of other partners in the value-added networks. For these identifiers global uniqueness is required and IRDI or URI are specified as variants:

- **Identification by IRDIs**

The internatinal standards ISO/IEC 11179-6 and ISO/IEC 6523 establishes an identification scheme for globally unique identifiers as displayed in Figure 15:

- ISO/IEC 11179-6 defines international registration data identifier (IRDI);

- ISO/IEC 6523 defines registration authority identifier (RAI).

The technical specification ISO TS 29005-2 defines specific syntax and requirements for concept identifiers based on IRDI.

Other standards also define this syntax such as ISO 13584-26.

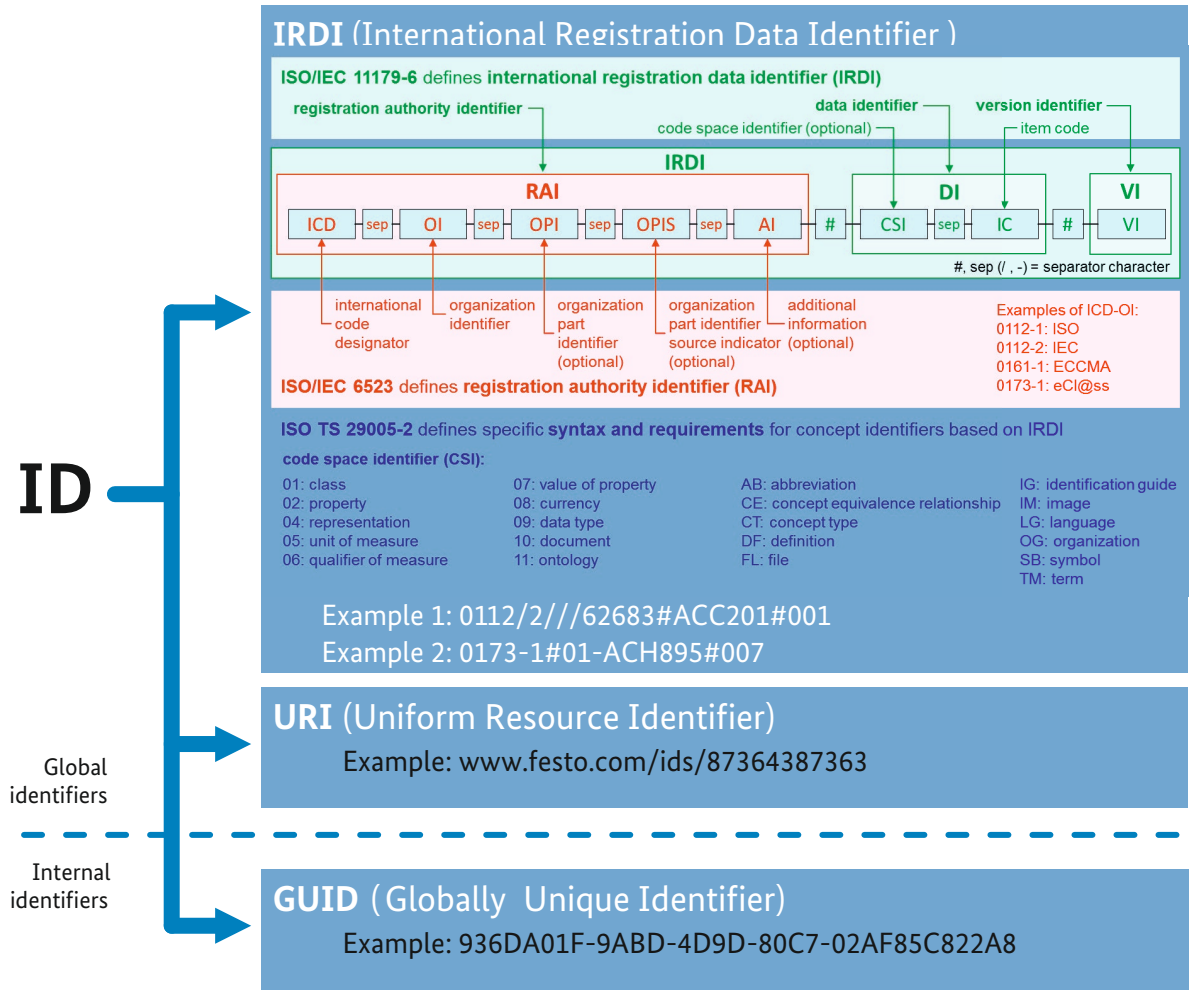
These standards are appropriate for the identification of standards and consortium specifications properties and classes.

- **Identification by URIs**

The ICT technologies have similarly developed and approved a global standard for identifiers (W3C). By means of URI Schema, domain and path a type-secure, globally unique and distributed volume of identifiers can be realised (see Annex E).

Internal identifiers are valid if they do not need to be accessible to other partners in the value-added networks. These can be identifiers of manufacturers internal data elements, for example.

Figure 15 – Identifiers



3.3.3 Secure added-value networks

Specific security requirements may require secure identities. A description of identifiers, identities, unambiguous identities and secure identities, including a matrix chart, needs to be worked out by the workgroup “secure identies” and is not part of this specification (see [1]).

3.3.4 Association of further identifiers

In order to meet the requirement for different identifiers for various aspects of the provision of services (production, logistics and operation), it makes sense to place in the Administration Shell further properties which hold further indirect identifiers, such as a GS1 identifier, locally.

Requirement # 7

The Administration Shell should allow retrieval of alternative identifiers such as a GS1 and GTIN identifier in return to asset ID (deferencing).

3.3.5 Best practices

As far as identification of assets and Administration Shells is concerned, there are numerous overlaps with further interests in companies. Annex D gives further information on how these overlaps can be resolved in execution.

3.4 Requirements regarding the Administration Shell

Requirement # 8

The Administration Shell consists of header and body.

Requirement # 9

The header contains information about the identification.

The header contains minimal information about identification. It uniquely identifies the Administration Shell. This identification can therefore also serve as a root entry point for an application programming interface (API) to browse for information and functionalities. The header contains also the identification of one or multiple assets that are described by the Administration Shell. The header also indicates if these assets are asset types or asset instances.

Requirement # 10

The body contains information about the respective asset(s).

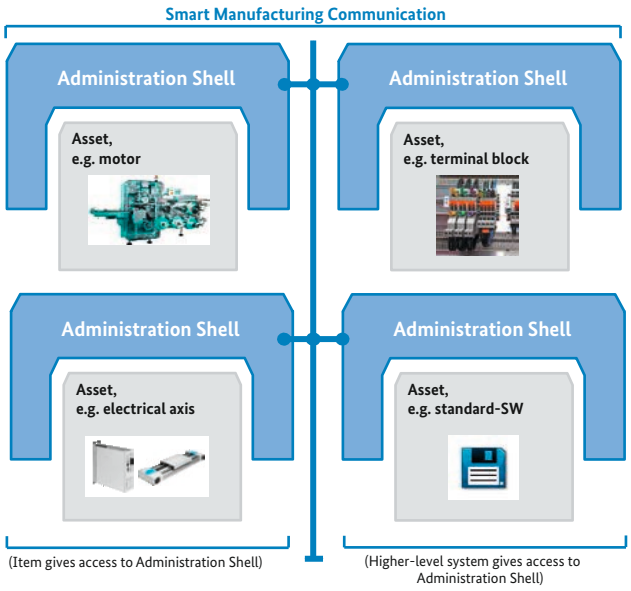
The body contains information about the asset(s) and describes functionalities that are associated with the asset(s) or the Administration Shell. The information can concern asset type(s) and/ or asset instance(s). Thus, the body serves as the actual carrier of information and functionality.

Requirement # 11

The information and functionality in the Administration Shell is accessible by means of a standardised application programming interface (API).

Figure 16 - Smart Manufacturing compliant communication, which provides access to a wide range of Administration Shells

Figure 16 – Smart Manufacturing compliant communication, which provides access to a wide range of Administration Shells



The Administration Shell shall represent information and functionality associated with the respective assets. By this principle, the virtual representation of a production system is always congruent with the actual one.

To access these entities, a standardised application programming interface (API) is required. This API can also be used for a lifelong maintenance of the information in the respective Administration Shells and can provide efficient retrieval options.

The entirety of the API invocations forms the “smart manufacturing compliant communication”, which allows for abstracting over different deployment options for Administration Shells (see Figure 16), taking into account corresponding security requirements (including access rights).

Requirement # 12

The Administration Shell has a unique ID.

Requirement # 13

The asset has a unique ID.

It should be ensured that the link between assets and Administration Shells does not break, even if they are saved in digital repositories or saved in a manner which spans all value-added partners.

Requirement # 14

An industrial facility is also an asset, it has an Administration Shell and is accessible by means of ID.

The concepts of the Administration Shell shall be applicable on all hierarchy levels of an industrial facility, such as factories/plants, production lines, stations, controls and field devices.

Requirement # 15

Types and instances must be identified as such.

Administration Shells can be formulated for both types and instances of assets. It must be possible to differentiate between these. Ideally, a information relationship will also be established between component producers and the system integrator that, where required, allows updated developments regarding asset types to be communicated to the system integrator and conversely feedback to be transmitted to the component producer about the component use.

Requirement # 16

The Administration Shell can include references to other Administration Shells or Smart Manufacturing information.

For the cross-linking of information to knowledge it is important that this can also take place on an over-arching basis. Thus, for example, a component can model the dependencies on other components or can contain a circuit diagram which refers to other components.

Requirement # 17

Additional properties, e.g. manufacturer specific, must be possible.

The Smart Manufacturing component can only meet future requirements if, in addition to the information content stipulated by standards, consortia and manufacturer properties can also be quickly agreed and processed. The Administration Shell should therefore support this consortia and proprietary information content and, associated accordingly, necessary collaboration processes.

Requirement # 18

A reliable minimum number of properties must be defined for each Administration Shell.

Administration Shells shall be a reliable source of information to other Administration Shells or other systems. To do so, it shall be possible to define for each asset class a minimum set of properties and value statements which can be relied upon.

3.5 Classes of Property

From Requirement # 13 and Requirement # 15 result four classes of property:

- **Basic properties:** Properties that are mandatory and standardised for all Administration Shells.
- **Mandatory properties:** Properties that are mandatory and standardised for submodels of Administration Shells.
- **Optional properties:** Properties that are standardised but non-mandatory for submodels of Administration Shells.
- **Free properties**, e. g. manufacturer specific properties.

3.6 Submodels

Submodels are used to structure the information and functionality of an Administration Shell into distinguishable parts. Each submodel refers to a well-defined domain or subject matter. Submodels can become standardized and thus become submodels types. An Administration Shell of an asset can refer to a submodel type, causing this submodel instance to feature a guaranteed set of properties and functions.

NOTE: A later document will define how submodels will be standardized on European level.

3.7 Requirements regarding individual elements of information

The following requirements are applicable to properties of an Administration Shell; the properties are structured by submodels. Standardized submodels types can require the presence of properties in submodel instances.

Requirement # 19

The properties and other elements of information in the Administration Shell must be suitable for types and instances.

Administration Shells can be formulated for both types and instances of assets; thus, properties need to be able to describe particularities of on asset type and, may be in addition, the asset instance. An Administration Shell of an asset instance shall also feature the properties of the Administration Shell of the respective asset type, as long as these properties were not overridden.

NOTE: This can for example also mean, that the descriptions of an asset type are extended over the life-time or, for an asset instance, properties are added, amended or deleted depending on (maintenance) activities of the respective asset.

Requirement # 20

There must be a capability of hierarchical and countable structuring of the properties.

The volume of properties to be organised is rather large and it is anticipated that it will steadily increase in the progress of Smart Manufacturing. The means should therefore ensure that these quantities remain manageable for human and machine. It is thus necessary to be able to organize properties using combinations of structures and arrays.

Requirement # 21

Properties shall be able to reference other properties, even in other Administration Shells.

Properties referring to other properties allow to express dependencies on values contained in other Administration Shells. Also, knowledge can be modelled by interrelating two properties by a predicated relationship (equivalent to a semantic triple).

Requirement # 22

Properties must be able to reference information and functions of the Administration Shell.

The structure of submodels and properties serves as a clearly defined “table of contents” for all information and functions within the Administration Shell. Properties are of uniform structure, they are standardised and they are thus providing a very stable source of information.

Complex data (digital models) and functions on the other hand can have a large variance and can be very complex in structure. Therefore, properties shall be able to refer to these complex data and functions in order to provide an anchor point for these entities in the above “table of contents”.

NOTE: This concept relies on an extended understanding of a IEC 61360 property concept.

Annex

Annex A

Glossary

Forword notice

Definition of terms are only valid in a certain context. The current glossary applies to the context of this document.

asset
physical or logical object owned by or under the custodial duties of an organization, having either a perceived or actual value to the organization

NOTE 1 to entry: In the case of industrial automation and control systems the physical assets that have the largest directly measurable value may be the equipment under control.

→ [SOURCE: IEC TS 62443-1-1:2009, 3.2.6]

property
defined parameter suitable for the description and differentiation of products
→ [SOURCE: ISO/IEC Guide 77-2]

NOTE:

The concept of type and instance applies to properties.

The property types are defined in dictionaries (like IEC component Data dictionary or ecl@ss), they do not have a value.
The property type is also called data element type in some standards.

The property instances have a value and they provided by the manufacturers.
A property instance is also called property-value pair in certain standards.

attribute
data element for the computer-sensible description of a property, a relation or a class
→ [SOURCE: ISO/IEC Guide 77-2]

type
hardware or software element which specifies the common attributes shared by all instances of the type
→ [SOURCE: IEC TR 62390:2005-01, 3.1.25]

instance
concrete, clearly identifiable component of a certain type

NOTE 1 to entry: It becomes an individual entity of a type, for example a device, by defining specific property values.
Note 2 to entry: In an object oriented view, an instance denotes an object of a class (of a type).

→ [SOURCE: IEC 62890:2016, 3.1.16] 65/617/CDV

application
software functional element specific to the solution of a problem in industrial-process measurement and control

NOTE: An application may be distributed among resources, and may communicate with other applications.

→ [SOURCE: IEC TR 62390:2005-01, 3.1.2]

component
product used as a constituent in an assembled product, system or plant
→ [SOURCE: IEC 61666:2010, 3.6)

system
interacting, interrelated, or interdependent elements forming a complex whole
→ [SOURCE: IEC TS 62443-1-1:2009, 3.2.123]

device
independent physical entity capable of performing one or more specified functions in a particular context and delimited by its interfaces
→ [SOURCE: IEC 61499-1:2012, 3.29]

class
description of a set of objects that share the same attributes, operations, methods, relationships, and semantics
→ [SOURCE: IEC TR 62390:2005-01, 3.1.4]

sector
grouping based on similar production processes, similar products, similar activities or similar behaviour in financial markets

EXAMPLES: Health care technology, Environment-Health protection-Safety, Metrology and measurement-Physical phenomena, Testing, Mechanical systems and components for general use, Fluid systems and components for general use, Manufacturing engineering, Energy and heat transfer engineering, Electrical engineering, Electronics, Telecommunications-Audio and video engineering, Information technology, Image technology, Precision mechanics-Jewellery, Road vehicles engineering, Railway engineering, Shipbuilding and marine structures, Aircraft and space vehicle engineering, Materials handling equipment, Packaging and distribution of goods, Textile and leather technology, Clothing industry, Agriculture, Food technology, Chemical technology, Mining and minerals, Petroleum and related technologies, Metallurgy, Wood technology, Glass and ceramics industries, Rubber and plastic industries, Paper technology, Paint and colour industries, Construction materials and building, Civil engineering, Military affairs-Military engineering-Weapons, Domestic and commercial equipment-Entertainment-Sports.

→ [SOURCE: Alliance pour l'Industrie du Futur – Smart Manufatcuring Standards Landscape]

unit with passive communication ability
physical unit with a passive data medium that can be read (RFID, barcode, etc.)
→ [VDI/VDE GMA FA 7.21]

technical functionality
functionality of the Administration Shell that is exposed by an application programming interface (API) and that is creating added value to the respective assets(s)

virtual representation
entirety of information of the Administration Shell, such as submodels, properties and complex data objects, covering digital models for the respective asset(s) for all applicable life-cycle phases

dereferencing
accessing an alternative identifier from a regular ID

Annex B

Relevant standards

B.1 IEC 61360 dictionaries, classes and properties

The IEC 61360 series provides a framework and an information model for product dictionaries.

The concept of product type is represented by “classes” and the product characteristics are represented by “properties”.

B.1.1 Classes

A hierarchy of classes is a structure of classes according to the commonalities of the properties in the classes (“taxonomy”). Properties associated with classes can be inherited.

B.1.2 Properties

A property, when used, has an associated value, such as a numerical value together with a unit, or a qualitative value taken from a predefined list.

The information model divides the attributes of properties into four main groups:

- identifying attributes (code, version number, revision number, preferred name, synomymous name, short name ...),
- semantic attributes (definition, note, remark, source document of definition ...),
- value attributes (data type, value format, unit of measure, alternative unit of measure, references class identifier, number of significant digits) and
- relationship attributes.

Figure B.1 – Example: Representation of a class tree in IEC CDD

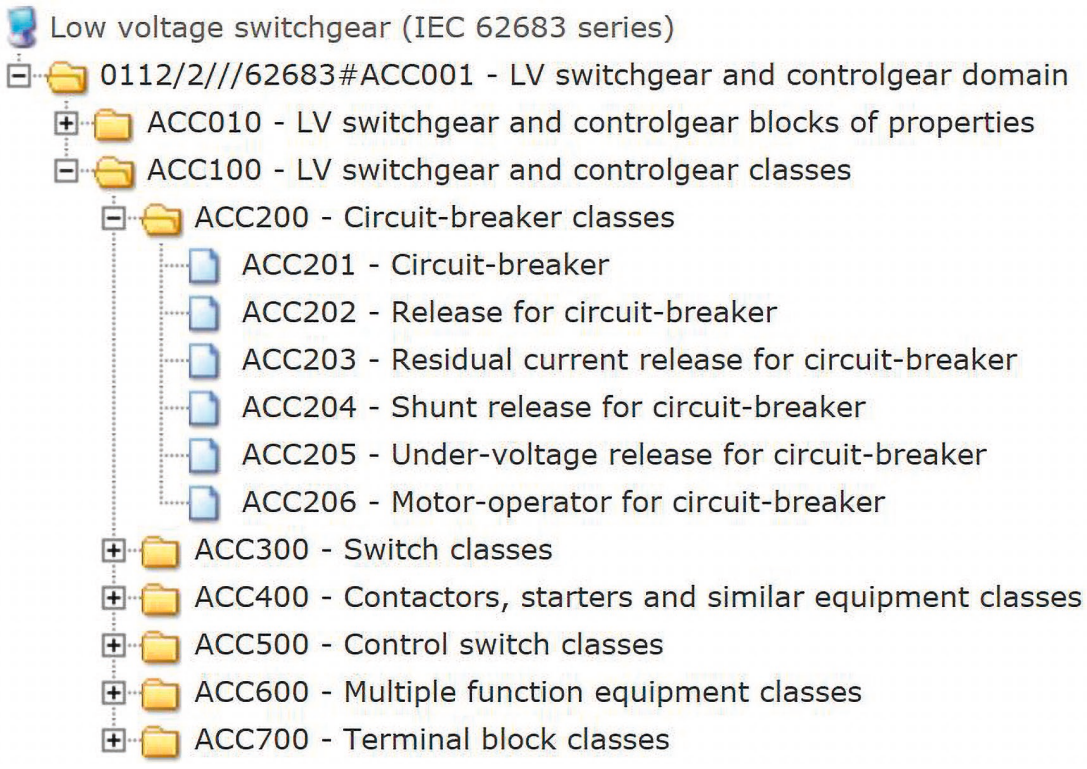


Figure B.2 – Example of representation of a property with some attributes in IEC CDD

Code:	0112/2///62683#ACE424
Version:	001
Revision:	01
IRDI:	0112/2///62683#ACE424#001
Preferred name:	rated current
Synonymous name:	
Symbol:	In
Synonymous symbol:	
Short name:	
Definition:	maximum uninterrupted current equal to the conventional free-air thermal current (Ith)
Note:	
Remark:	
Primary unit:	A
Alternative units:	
Level:	
Data type:	LEVEL(MAX) OF REAL_MEASURE_TYPE

B.1.3 IEC Common Data Dictionary (IEC CDD)

Based on IEC 61360, classes and property dictionaries (domain ontologies) for the description of production resources can be created.

IEC CDD – based on IEC 61360 - may be used as templates or structures for defining company specific data collections for various applications, such as catalogues, databases, and master-data repositories. Such consistent data collections are indispensable for data integrity and efficient management of corporate product and service portfolios including data sharing.

The standard in data base format IEC Common Data Dictionary (IEC CDD) is a common repository of concepts for all electrotechnical domains based on the methodology and the information model of IEC 61360 series, and provides:

- unambiguous identification of classes and properties, and their relations;
- commonly accepted terminology and definitions based on accepted sources such as IEC International Standards, other International Standards, industry standards, or public authorities;
- hierarchies of concepts enabling users to appropriately characterize their products and services;
- relevant conditions and constraints if necessary on possible values of characteristics;
- technical representation of concepts including units and data types and their identification.

IEC CDD offers free access to it's dictionary and the license conditions allow the use of the classes and properties in business, engineering and service/maintenance processes and related applications.

There are various organizations which maintain dictionaries for ontologies and for example:

- IEC CDD:
  - provided by IEC,
  - based on IEC 61360 Standard,
  - provides ontologies for various technical areas and units based on ISO and IEC (product) standards.

- eCl@ss:
    - provided by eCl@ss e.V. consortium.,
    - based on ISO 13584 (similar to IEC 61360),
    - provides ontologies for various technical areas based on consortia standards.

B.2 Digital Factory (IEC TS 62832)

B.2.1 Introduction to Digital Factory

The standard for the Digital Factory, IEC TS 62832-1, defines a framework of abstract definitions for:

- Assets of production systems
- Structural and behavioural relationships
- Feature (property) management
- Hierarchical relationships
- Technical aspects

As asset is understood as a physical or logical object which is owned or managed by an organisation and which has an actual or perceived value for the organisation. The standard addresses in particular the utilisation phase of such assets and the related production facilities, their design, construction, commissioning, operation and maintenance. For the identification of concepts the standard uses identifiers according to ISO 29002-5. The assets under consideration can also be identified by means of other identifiers (e.g. URI).

Assets ("PS asset", real or logical items) are described by means of asset descriptions ("DF asset", Virtual Representation). Classes of assets are modelled by means of so-called asset classes and thus stand respectively for one or more assets which share the same set of features. The features of the assets are described by means of data elements. If the described assets have a modular structure, the corresponding asset descriptions (asset classes) can similarly describe a modular structure. In this case the asset class describes the modules as assets and represents the links between the assets.

Most important for the description of an asset is the breakdown of the description into "header" and "body". The "header" contains information for identification and designation of the specific asset in the respective production system and stipulates aspects for administration of the asset.

Figure B.3 – Structured asset class in the Digital Factory

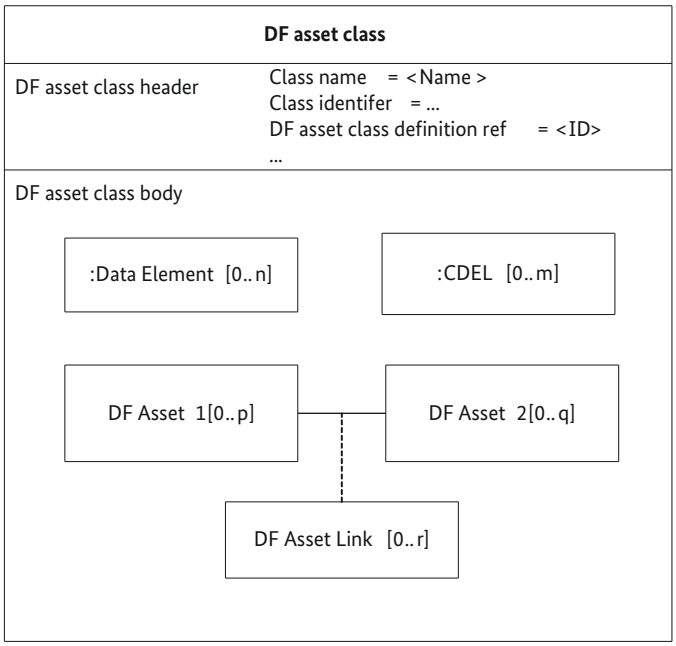
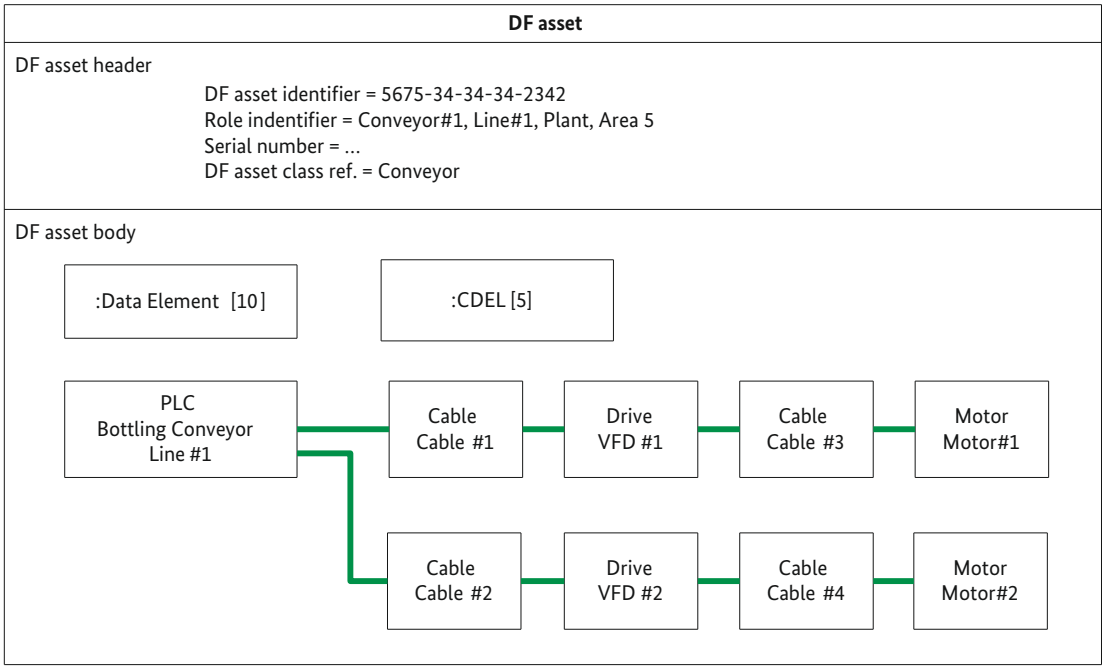


Figure B.4 – Description of a structured asset that is compposed of several assets



The "body" contains data for description of the features of the asset class with their respective disposition (i.e. data element values) for the specific asset. For the specification of data elements the Digital Factory makes reference to the IEC 61360 properties (see Figure B.1) and ISO 22745.

The main contribution of the Digital Factory Framework is the clear and unambiguous separation of data domains and their respective concepts:

- Dictionary domain with dictionaries for providing DF asset class definitions (i.e. templates for asset descriptions).
- Library domain with libraries for providing DF asset classes (i.e. asset type descriptions).
- Digital Factory domain providing DF assets (i.e. description of the real or logical assets).

This separation allows to define clear rules within the different domains (e.g. regarding the usage of identifier types), allows for automated processing of the data and enables the usage of property-based data for engineering use cases (e.g. for supporting design decisions based on library data or for support of design validation based on Digital Factory data).

These concepts may be applied in conjunction with other design languages (e.g. Automation ML and SysML).

B.2.2 Compatibility of the Digital Factory with the concept of Administration Shell

The objective of the work in the IEC is to create a standard (known as "Digital Factory") for electronic, machine-readable representations of production systems which can be used in all life cycle phases of the production systems. At the same time exchange of information between the different participants in the design, construction and operation of a production system should be supported. An important area of focus is the semantic clarity of the information, which is why the standard "Digital Factory" is based on property descriptions of the components of the production system. The utilisation of property descriptions for the components and component groups allows automated processing of the description of the production system – e.g. in validation of the system design. The standard "Digital Factory" makes no assumption in regard to the management of the data in the description of the production system (i.e. no central repository is defined).

Figure B.5 – Comparison of the Administration Shell concepts and the Digital Factory concepts

Similarities in the structure of the Administration Shell and the Digital Factory, as exemplified by an electrical axis system

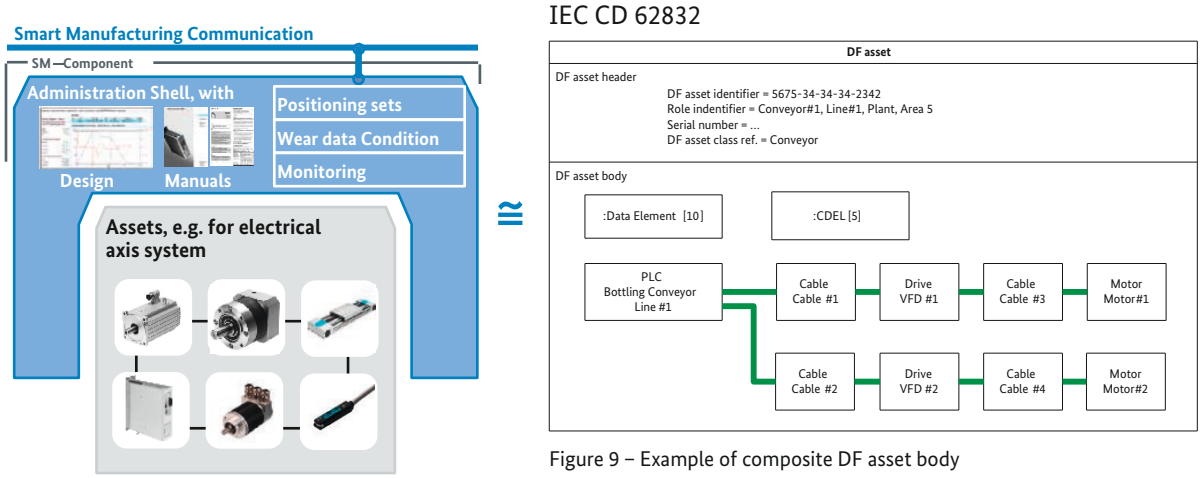


Figure 9 – Example of composite DF asset body

Source: IEC 62832

An objective of Smart Manufacturing is automated configuration of a production system in accordance with varying production orders. This objective can only be achieved based on a machine-processible description of the production system and the products to be manufactured. The information model for Industrie 4.0 is therefore based on the standard "Digital Factory" and extends it to include further required information. In order to achieve the required flexibility for the automated configuration of a production system the production resources are not only represented by passive data elements, but also by independent I4.0 Components, which are not only centrally administered, but which can also be understood as being a distributed repository. The data elements of the Digital Factory (referred to there as DF assets) at the same time form the basis for the Administration Shells of the corresponding I4.0 Components.

B.3 AutomationML (IEC 62714)

B.3.1 AutomationML Overview

AutomationML (AML) is a scalable, open, neutral, XML based, extendable, and free data exchange format, standardized in IEC 62714. It has been developed in order to support the data exchange in a heterogeneous engineering tools landscape by enabling a discipline and company crossing transfer of engineering data between software tools – in a consistent manner and without data loss.

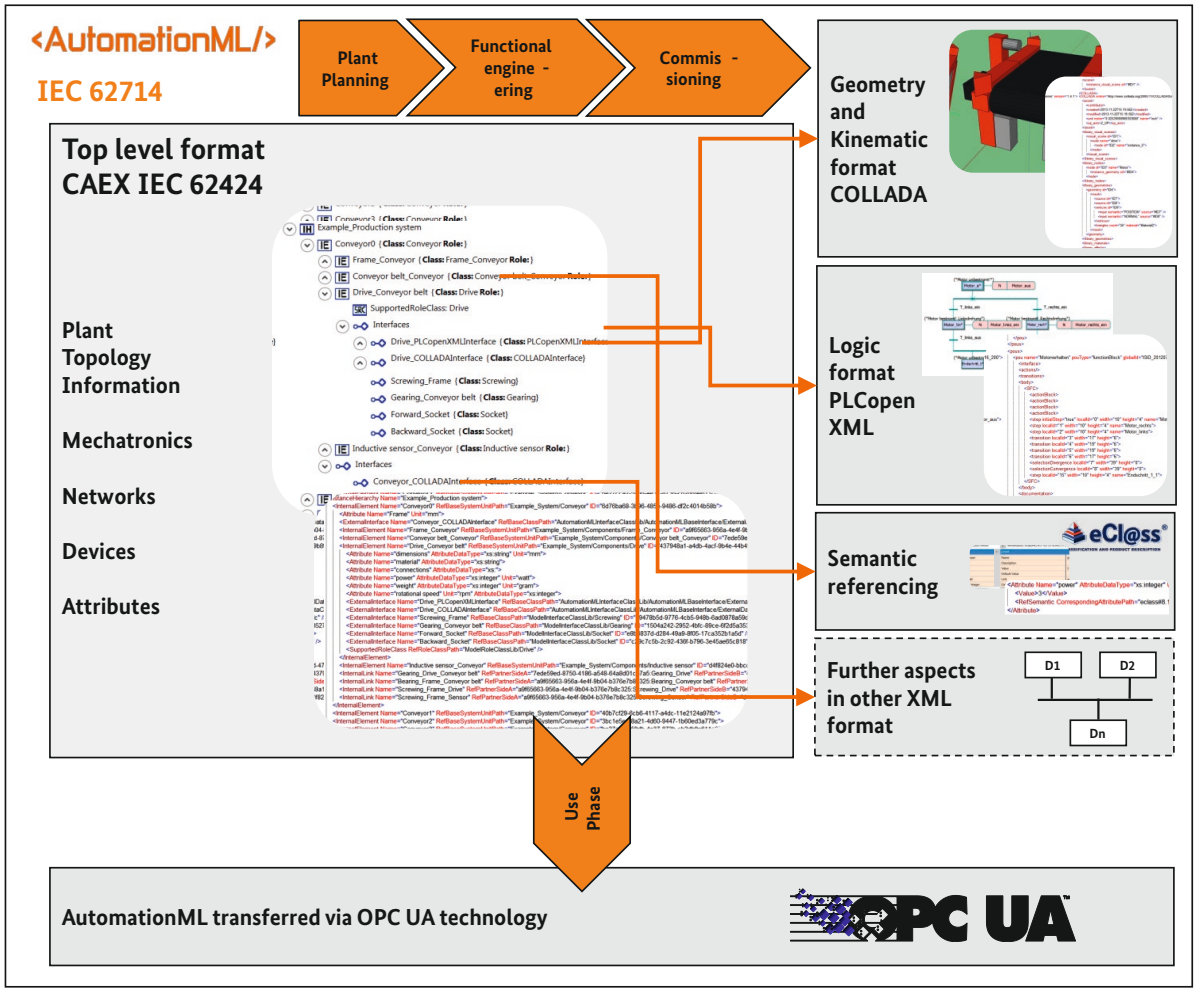
With the meta-model, that AML provides, production system structures can be modelled expressed as a hierarchy of AutomationML objects. They can contain geometry and kinematics, logic data (behavior and sequencing), and relations among AutomationML objects as well as references to information that is stored in external documents. Following this object-oriented paradigm, each AutomationML object can integrate different information with different semantics related to different disciplines provides an overview of the distributed document architecture of AML.

CAEX is the top level format of AML and is characterized by the four following concepts (see Figure B.6), where the AML objects are represented as InternalElements (IE)

- By using RoleClasses (RC), semantics for AML objects can be modeled. RoleClasses describe functions of a physical or logical object independent of a technical implementation. They offer a possibility to specify an object in an abstract way and independent of the manufacturer. They are stored in RoleClass libraries.

- By using InterfaceClasses (IC), the semantics for the interfaces of AML objects can be modeled. InterfaceClasses describe relations of physical or logical objects or references to documents independent of a technical implementation. They are stored in interface class libraries.

Figure B.6 – Architecture of AutomationML



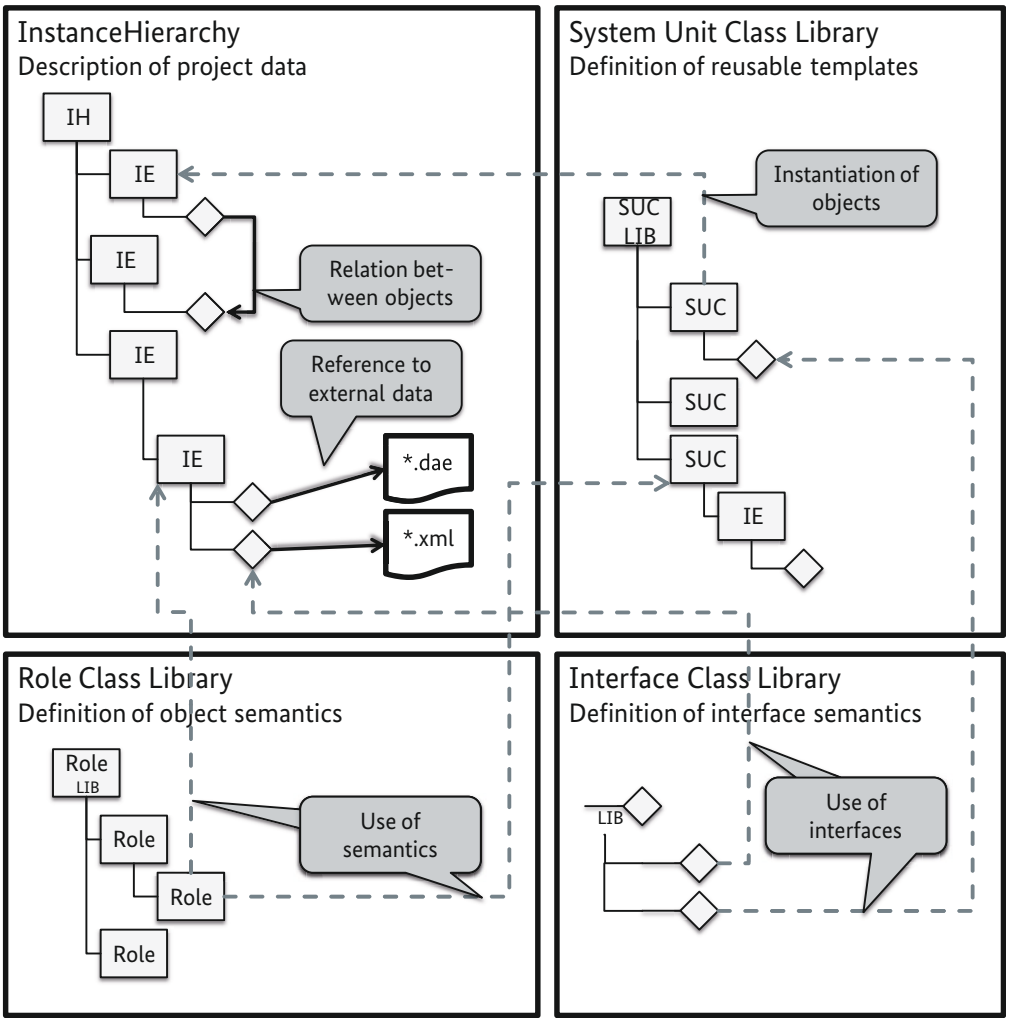
- By using SystemUnitClasses (SUC), types or templates of AML objects can be modeled. They are stored in system unit class libraries.

- The InstanceHierachy (IH) stores the actual project.

For a consistent data exchange, an object identification concept is essential. The concept, AML provides, specifies that every object shall have an UUID (Universally Unique Identifier), which shall be kept as long as the object exists. With this ID persistence, it is possible to identify a certain AML object throughout its entire life cycle.

AML mechanism to reference external documents enables the integration of further information aspects to AML objects. Therefore, it provides a special set of interface classes and role classes being used to uniquely identify the document type and the semantics of attached information. In case the referenced document has a unique identification concept for its content, also a referencing into the referenced document is possible. Besides the AML meta-model, there is the specification for the data exchange format available as well. Since AML is an XML based format, XML specific features are available, like XML encryption, XML signature, or XML serialization.

Figure B.7 – CAEX concepts

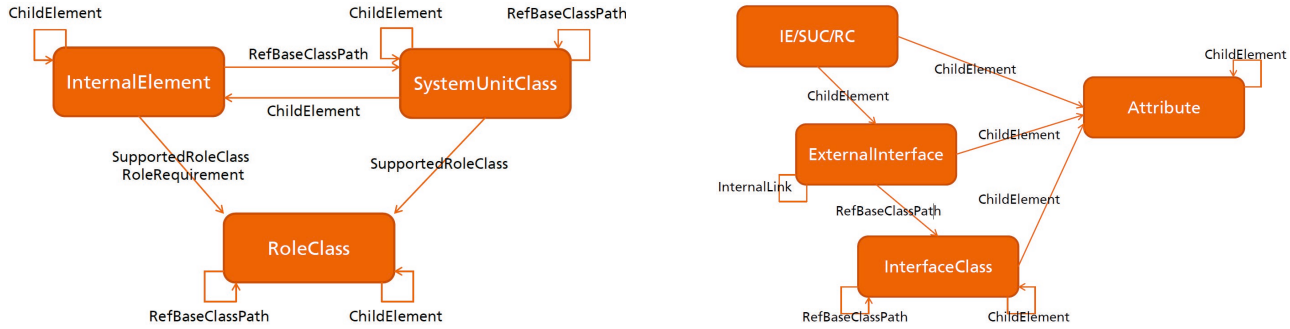


B.3.2 AutomationML Modeling Concepts

The main AutomationML elements and all possible relations between them are shown in Figure B.7. The organization in a hierarchical tree involves that every element type can have child elements of the same type (see ChildElement relations). SystemUnitClasses and RoleClasses are defined in an inheritance structure; the RefBaseClassPath relations define such inheritance relations.

An InternalElement shall be inherited from a role class and can inherit from a SystemUnitClass. In the scope of AML, SystemUnitClasses are only templates and can be changed after instantiation. This allows a modification of the AML object regarding instance specific needs, e.g. deleting, editing, or adding attributes and interfaces. However, it is also possible to model AML objects directly into the InstanceHierarchy – without having instantiated a SystemUnitClass. The backward relation indicates that a SystemUnitClass can define sub-elements in terms of InternalElements. These sub-elements (InternalElements) should be included on instantiation. RoleClass, SystemUnitClass, or InternalElement can contain arbitrary nested Attributes and Interfaces (see ChildElement relations). The Interfaces are called ExternalInterfaces and shall inherit from an InterfaceClass. They can be connected to other ExternalElements via InternalLinks. The InterfaceClasses are stored hierarchically and shall have independent inheritance relations. InterfaceClasses and certainly ExternalInterfaces may also have arbitrary nested attributes.

Figure B.8 – Relations between AutomationML elements



RoleClasses can be assigned to SystemUnitClasses and InternalElements. In the latter case there are two different ways to make the RoleClass assignment using SupportedRoleClasses or the RoleRequirement. The remaining arrows in the left diagram of Figure 3 describe the possible relations between InternalElements and SystemUnitClasses.

When RoleClasses are assigned to SystemUnitClasses or InternalElements or when InterfaceClasses are assigned to ExternalInterfaces, their attributes can be copied, manipulated, or deleted. No strict inheritance relation is applied here, either.

B.3.3 Interoperability of Administration Shell supported by AutomationML

With respect to the structure of the Administration Shell, AML provides a number of concepts that complies with the concepts of the structure of the Administration Shell as well as with many of the relevant standards described in this annex. Due to the strong focus of AML in the field of manufacturing engineering, many engineering tools that support AML could provide structured information of assets, e.g. automation system components, already. These could be directly consumed for the descriptive part of the header and the individual sub models in the body of the Administration Shell.

Concerning the different life cycle phases that need to be supported from the Administration Shell, AML further complies with this concept with its following AutomationML main elements:

- **RoleClasses**, which represents semantics of objects (e.g. parallel gripper)
- **SystemUnitClasses**, which represent object types, which can be linked to role classes (e.g. specific product family of parallel gripper from a certain manufacturer)
- **InternalElements** that represent the object (type) instances, which can be linked to role classes and/or System Unit Classes (Actual parallel gripper with an individual serial number)
- **InterfaceClasses** allows semantic relations between the AutomationML objects or references to documents (e.g. mechanical connection between parallel gripper with robot or reference to 3D-CAD representation)

Though AML follows the object–oriented paradigm, it is not as strict as it is defined in object-oriented programming, as described in section B.3.2. This allows a more flexible representation of instances in a data model, by changing the attributes of an InternalElement independent from the SystemUnitClass. Even inconsistent data models are possible to be represented by AML by design, which addresses the real life situation of incomplete models during the first life cycle phases.

The standardized meta-model of AML makes no assumptions on the actual application domain. Hence, it can be used to represent the self-sufficient description of arbitrary data structures for the header of the Administration Shell. Further, it can be used for the individual description of the sub-model data structures representing the possible different aspects of the asset to be represented in the body of the Administration Shell.

This self-sufficient describing capability through AML is an important pre-requisite for realizing interoperability between arbitrary types of assets (physical, logical, software, etc.) represented by an Administration Shell.

B.3.4 Connectivity by means of OPC UA supported by AutomationML

The existing OPC companion specification OPC UA Information Model for AutomationML as well as the DIN Spec 16592: Combining OPC Unified Architecture and Automation Markup Language further describe a formal mapping between AML and OPC UA elements. In addition, AML enables the modeling of runtime data sources especially within further OPC UA servers or other runtime communication systems. This allows a formal transition between the data model described with AML in the Administration Shell and the relevant information model represented by OPC UA.

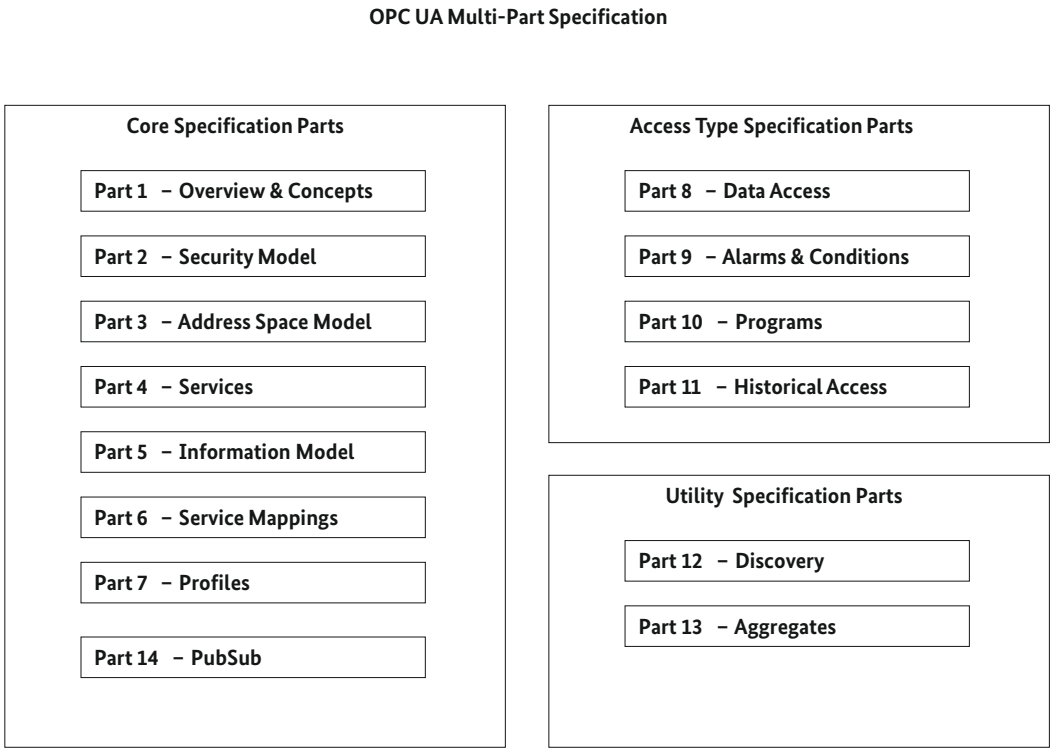
This allows to communicate and operationalize data models described with AML by means of OPC UA, which fulfills the requirement on smart manufacturing (Industry 4.0) conformal communication between multiple Administration Shells. Thus, AML can become an important enabler for interoperability in the smart manufacturing domain in the same way as OPC UA is for connectivity in that domain.

B.4 OPC-UA

B.4.1 OPC-UA Overview

OPC Unified Architecture (UA) is a series of standards that supports the vertical and horizontal connectivity between components, applications, and functions. OPC UA is applicable to components in all industrial domains, such as industrial sensors and actuators, control systems, Manufacturing Execution Systems (MES) and Enterprise Resource Planning (ERP) Systems, etc. OPC UA defines a common infrastructure model to facilitate an information exchange OPC UA specifies the following:

Figure B.9 – OPC UA Specification Organization



- The information model to represent structure, behaviour and semantics.
- The interface model to interact between applications.
- The communication model to transfer the data between end-points.
- The conformance model to guarantee interoperability between systems.

OPC UA series is organized as a multi-part specification, as illustrated in Figure B.8.

Parts 1 through 7 and Part 14 specify the core capabilities of OPC-UA. These core capabilities define the structure of the OPC *AddressSpace* and the *Services* that operate on it. IEC 62541-14 defines an OPC-UA publish subscribe pattern in addition to the *Client Server* pattern defined by the Services in IEC 62541-4. Parts 8 through 11 apply these core capabilities to specific types of access previously addressed by separate OPC COM specifications, such as Data Access (DA), Alarms and Events (A&E) and Historical Data Access (HDA). IEC 62541-4 describes *Discovery* mechanisms for OPC UA and IEC 62541-13describes ways of aggregating data.

Readers are encouraged to read Parts 1 through 5 of the core specifications before reading Parts 8 through 14. For example, a reader interested in UA Data Access should read Parts 1 through 5 and 8. References in IEC 62541-8 may direct the reader to other parts of this specification.

The 4th industrial revolution (Industrie 4.0) is driven by advanced information and communication technologies (ICT), which are becoming increasingly prevalent in industrial automation. In distributed, intelligent systems physical, real systems and virtual, digital data merge into cyber physical systems (CPS).

B.4.1.1 Secure, reliable and platform independent exchange of information

OPC-UA is the latest technology generation from the OPC Foundation for the secure, reliable and vendor-independent transport of raw data and pre-processed information from sensor and field level, through the control system level up to and into enterprise level with e.g. ERM, MOM, and PLM systems. With OPC-UA every type of information is available anytime and anywhere for every authorized use and to every authorized person.

B.4.1.2 Platform and vendor-independent

OPC-UA is independent of the vendor or system supplier that produces or supplies the respective application. The communication is independent of the programming language in which the respective software was programmed and it is independent of the operating system on which the application runs. It is an open standard without any dependence on, or bind to proprietary technologies or individual vendors.

B.4.1.3 Standardized communication via internet & firewalls

OPC-UA extends the preceding OPC industry standard by several important functions such as platform independence, scalability, high availability and Internet capability. OPC-UA is very simple to adapt. It connects the enterprise level down to the embedded systems of the automation components – independent of operating systems. OPC-UA uses TCP for an optimized, binary protocol. Web service and HTTP are optionally supported. Additional protocol bindings can be integrated easily. The integrated encryption mechanisms ensure secure communication over the Internet.

B.4.1.4 Service-oriented architecture

OPC-UA defines generic services and in doing so follows the design paradigm of service-oriented architecture (SOA), with which a service provider receives requests, processes them and sends the results back with the response.

B.4.1.5 Protection against unauthorized access

OPC-UA technology uses proven security concepts that offer protection against unauthorized access, against sabotage, the modification of data and against careless operation. The OPC-UA security concepts contain user and application authentication, the signing of messages and the encryption of the transmitted data itself. OPC-UA security is based on recognized standards that are also used for secure communication in the Internet, such as SSL, TLS and AES.

B.4.1.6 Accessibility and reliability

OPC-UA defines a robust architecture with reliable communication mechanisms, configurable timeouts and automatic error detection.

B.4.2 OPC UA Information Models

The core of OPC UA is specified by part 1 to 14. Especially part 5 “Information model” defines the basic rules for the address space. The OPC-UA address space is structured hierarchically, to foster the interoperability of clients and servers. The top levels are standardized for all servers. All nodes in the address space can be reached via the hierarchy. They can have additional references among each other, so that the address space forms a cohesive network of nodes. The OPC-UA address space not only contains instances (instance space), but also the instance types (type space).

OPC UA for devices (part 100) specifies in addition standardized structuring rules for component. Within components parameters (IEC 61360 properties) and methods can be hierarchically organized.

This part defines in addition modelling rules for composed components and network structures.

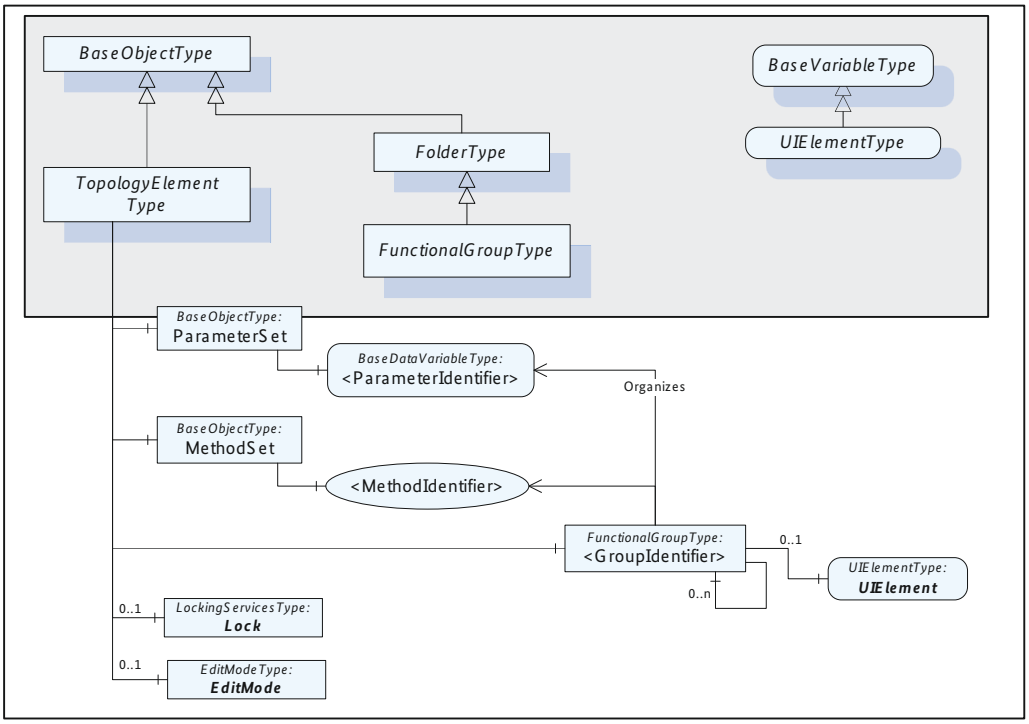
B.4.3 Usage of OPC UA to implement the Administration Shell

OPC-UA allows separating instance and type data in the AddressSpace. The usage of this concept enables components to add semantic rich self-describing information. The OPC-UA for devices information model defines how properties should be organized in components. IEC 61360 defines attributes that a component property should contain.

Submodels of components can be represented according to OPC UA for devices with OPC-UA folders (FunctionalGroupType). A component can have multiple submodels. Methods can be also defined in submodels.

An implementation of an OPC-UA information model shall only permit access to properties and methods for authorized users.

Figure B.10 – Component internal base mode



Annex C

Views

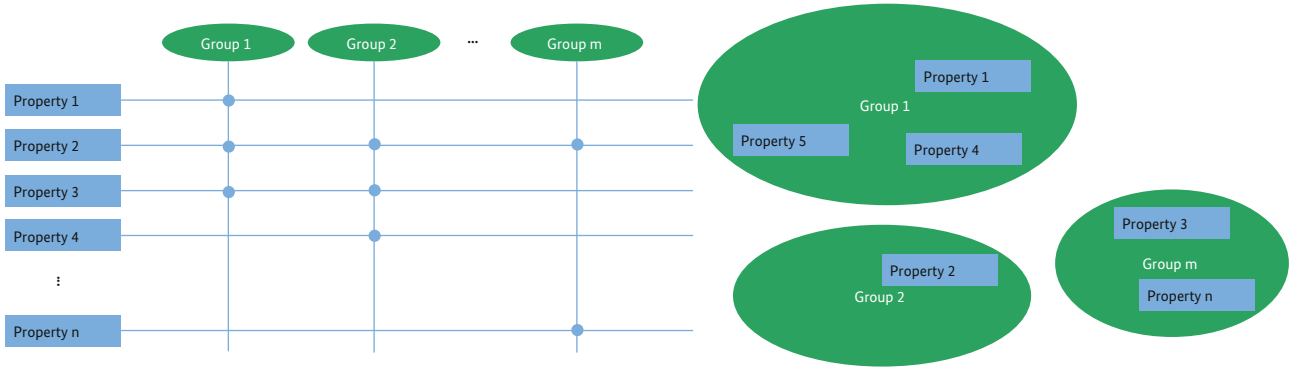
C.1 General

Assets or system of assets have many properties and it is necessary to gather them into different groups to facilate the understanding to the information.

A classification of asset properties into groups is generally provided by the asset manufacturer in its digital device description.

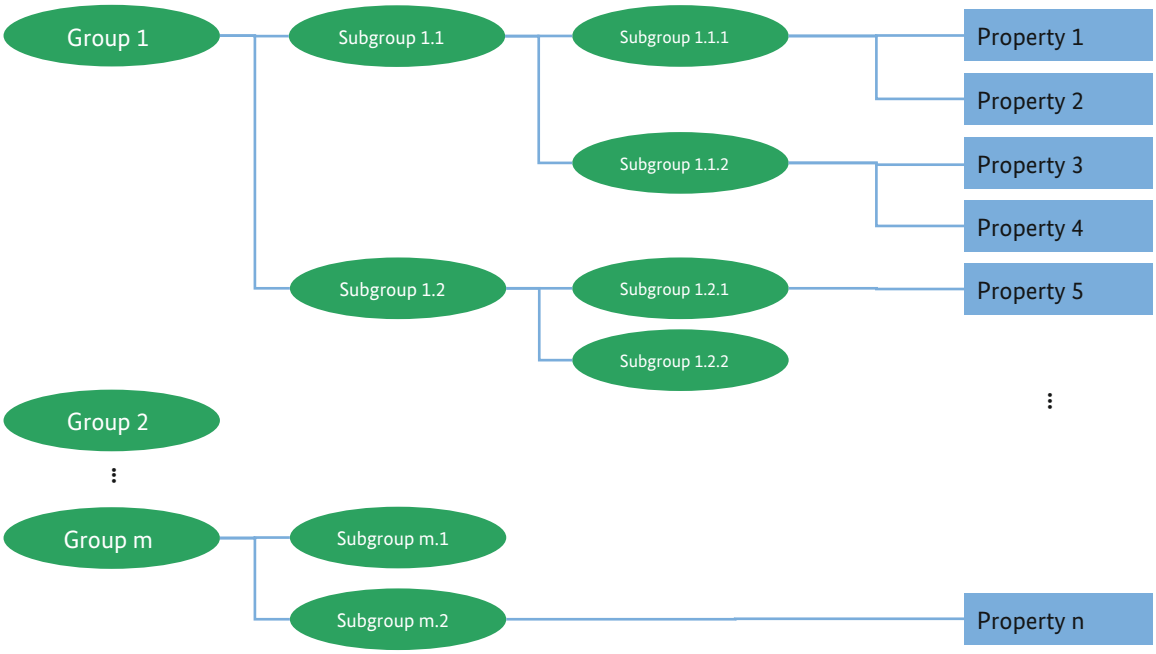
Properties may belong to several groups.

Figure C.1 – Properties belonging to several groups



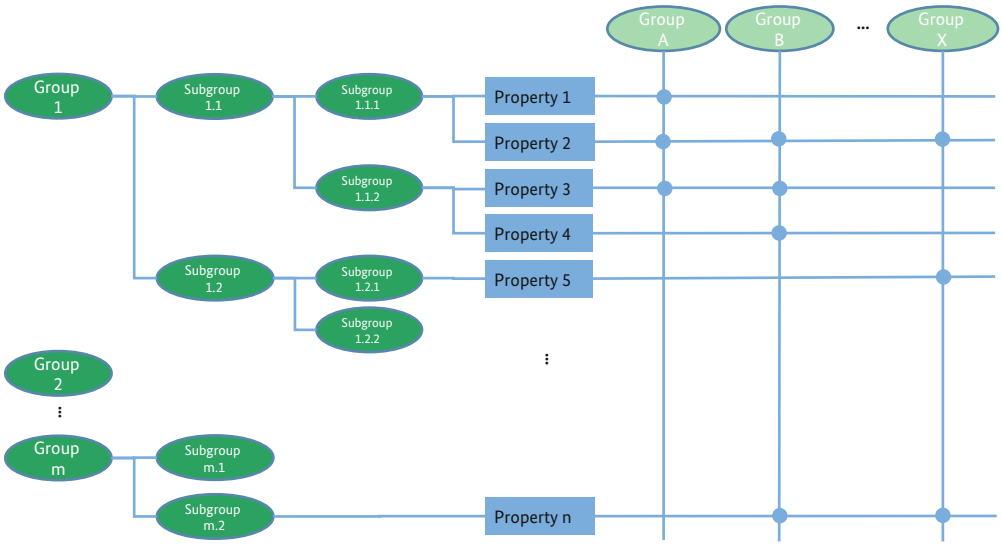
Groups may be organized in hierarchies. In this case, a property may only belong to one group.

Figure C.2 – Properties organized in hierarchies



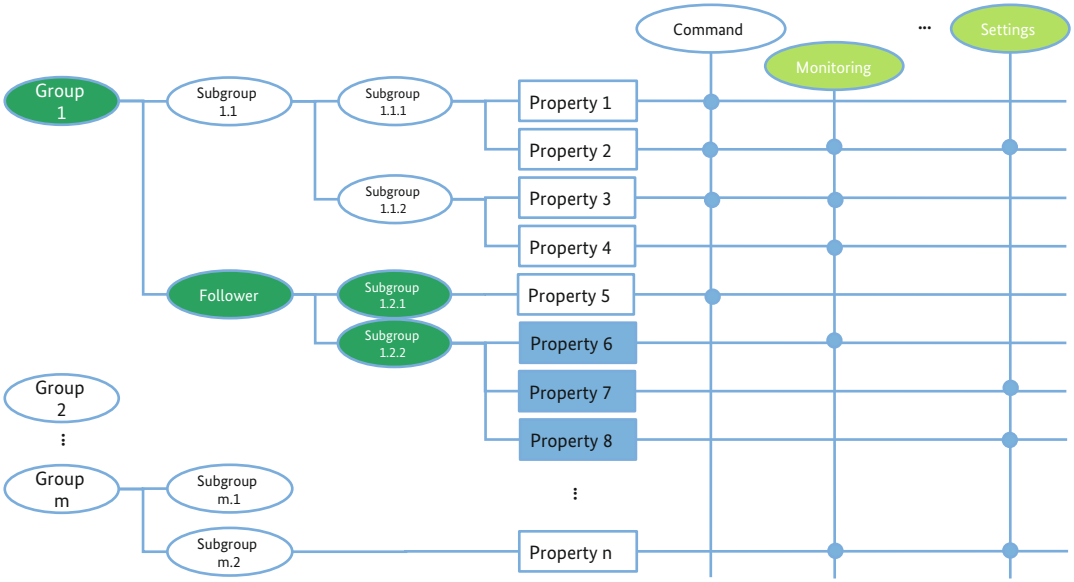
The purpose is to reduce the complexity for the human users of the engineering tools. The groups are used for either organize or filter the information or both.

Figure C.3 – Component internal base mode



For example, a motion specialist needs to display the settings and the monitoring information form the follower. Only property 6, property 7 and property 8 will be displayed.

Figure C.4 – Example of filtering



Different types of groups are defined in existing standards:

- Block,
- Basic view,
- Compatibility view,
- Technical discipline view,
- Parameter category.

They described in the following paragraphs.

C.2 Block

The concept of "block" is defined in IEC 61987-11 "Industrial-process measurement and control - Data structures and elements in process equipment catalogues - Part 11: List of Properties (LOP) of measuring equipment for electronic data exchange - Generic structures".

Manufacturers of assets, standards for classes and properties (LoPs) and classification organizations provide such organization of properties.

For example IEC 62683 " Low-voltage switchgear and controlgear – Product data and properties for information exchange", for the class "Circuit-breaker" defines the following blocks:

- Identification,
- General technical data,
- Main circuit
- Short-circuit
- Over-current release,
- Control and auxiliary circuits,
- Data communication,
- Installation, mounting and dimensions,
- Connection facilities,
- Product certificates and standards.

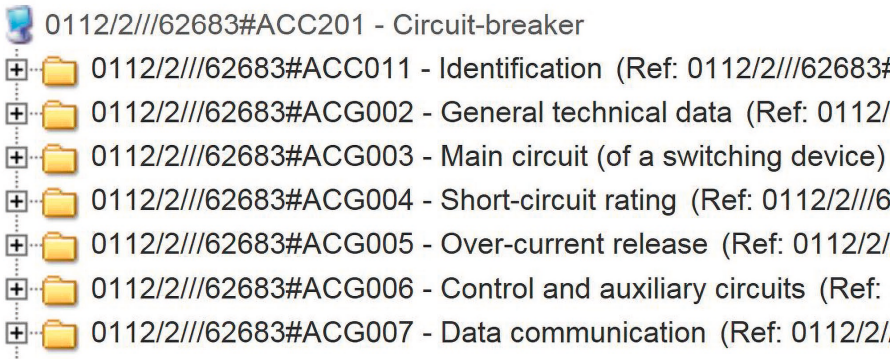
The organization of the blocks is generally functionally oriented.

C.3 Basic views

IEC CDV 62890 "Industrial-process measurement, control and automation – Life-Cycle-Management for systems and components" defines five typical groups:

- business,
- construction,
- function,
- performance,
- location.

Figure C.5 – Organization in blocks



That can also be named basic views and are defined in the following table.

Name	Description and examples
Business	Allows judging on the business suitability and performance of an asset. Examples: Manufacturer name, prices, terms of delivery, order codes."
Construction	Applies to the constructive deployment of the asset for selection, design of building structure or assemblies ... Contains a structure classification system pursuant to IEC 81346. Applies to properties in respect of: • physical dimensions and regarding start, processing and output values of the component. • modular view of subcomponents or a asset structure. • inputs and outputs of different signal types.
Function	Makes statements on the function pursuant to IEC 81346 and on the function of the asset components. Includes skills such as interpretation (?), commissioning, calculation (?) or diagnosis functions of the asset. Examples: current imbalance detection ability
Performance	Describes performance and behavioural characteristics. Examples: To allow a performance assessment and virtual commissioning of an overall system. Example: instantaneous short-circuit current setting
Location	Describes the capabilities of the asset regarding environmental conditions and positions of the asset. Describes the relationships capabilities of the asset or its parts or inputs and outputs with the linked assets. Examples: Degree of protection, working temperature, EMC characteristics, rated voltage, rated ultimate short-circuit breaking capacity.

C.4 Compatibity views

For assessing compatibility, IEC CDV 62890 "Industrial-process measurement, control and automation – Life-Cycle-Management for systems and components" also states that two other views shall be defined and maintained:

- capability view,
- requirements view.

C.5 Technical disciplines views

Filtering information according to the technical discipline of the tools user is very important. The table below describes the technical discipline views:

Name	Description and examples
Security	Can identify a property as security relevant. This property should be taken into account for an assessment of security.
Network	Makes statements in respect of electrical, fluidic, materials flow related and logical cross-linking of the asset.
Electrical	
Motion	
Automation	
Sensor	
Safety	
Thermal	
...	

C.6 Parameters categories

IEC 61915-1 “Low-voltage switchgear and controlgear – Device profiles for networked industrial devices – Part 1: General rules for the development of device profiles” defines the following classification of parameters also applicable to properties.

- Control parameters
  - Commands
    - Switching commands
    - Fault reset
    - Operating mode selection
  - Monitoring information
    - Operational signals
    - Fault signals
    - Warning signals
    - Maintenance signals
    - Network errors
  - Measurements
    - Operational measurements
    - Maintenance measurements
- Management parameters
  - Configuration
    - Operational levels
    - Fault levels
    - Warning levels
    - Maintenance levels
    - I/O assignment
    - Network configuration
  - Diagnostics
    - Operational diagnostics
    - Fault diagnostics
    - Warning diagnostics
    - Maintenance diagnostics
    - Network counters
    - Counters reset
  - Identification
    - Product identification
    - Manufacturer identification

Annex D

Best practices for identifiers for assets and Administration Shells

D.1 Introduction

Identification is one of the key topics for Smart Manufacturing. The designation and identification of assets and their Administration Shells shall be unambiguous and globally unique.

This annex describes the identification in general terms, the prerequisites and principles, and presents some examples of execution in the Plattform Industrie 4.0.

D.2 Who needs identification?

Identification is necessary for various processes within Smart Manufacturing, but the various processes require different methods of execution. The following paragraphs describe the practical needs for some processes.

D.2.1 Technical process

For unambiguous classification of assets in Smart Manufacturing, each asset requires a unique identification (see Requirement # 1).

Each piece of information must also be clearly discernible or identifiable. The unique identification of the functions and software components is an additional issue.

D.2.2 Logistics

Identification throughout the movement of goods continues to play an important role. This concerns the entire logistics chain such as products, and also packaging, transport vehicles, containers, warehousing, etc. Vendor parts are also affected, although not themselves being sold, are included in a product or system. Several identification marks are merged in a device or a system and the device/system similarly receives an identification mark. The relationships between the individual parts and thus also the identification marks are the responsibility of the manufacturers.

D.2.3 Distribution, after sales, marketing

Customers and prospects need quick and simple access to information about the product, the support (commissioning) or service (e. g. replacement parts, service engineers, remote maintenance, etc.).

Globally unique identification is similarly necessary for these processes.

Identification as related to different industrial processes

	Production	Supply chain	Sales, marketing, service
Target	Simple, fast and error free identification in production	Simple, fast and error free identification in material flow	Customer access/loyalty, extended services: e.g. autotuning, commissioning, etc.
Customer	Internal processes	Internal processes, suppliers & customer processes	Potential buyers, customers
Technology	PLC interface, industrial readers	Backend systems (ERP), industrial readers	Standard smartphone, no specific apps
Contents	Material numbers, serial number	Material numbers, serial number, quantities & additions	Catalogue information, spare parts, sales contacts, support, CAD, etc.
Further comment	Secure labeling, attached to product or carrier	Secure standardized labeling, nearly all products, utilities like trays, package units, etc.	Additional labeling, selected products, simple and fast
Market usage	Barcode, Data Matrix, RFID, QR-Code	Barcode, Data Matrix, RFID, QR-Code	QR-Code, Data Matrix

D.3 Principles

- As few characters as possible in the identification, properties, features and further information can be retrieved under the identification.
- ID should be as short as possible in order also to be able to mark small parts that have little space for a code.
- The object should be identifiable throughout the world by means of one identification (ID), one ID string.
- The ID string consists of ASCII characters, only a few special characters should be used (see below, “use of characters”).
- The ID is always globally unique (“my ID only exists 1x, I am an individual”).
- Without overlaps with other users, i.e. inclusion of the manufacturer/company name.
- Companies internally ensure non-overlapping of their IDs.
- If codes are used then the issuing entity of the codes must also be shown.
- Technical data, properties, features, etc. are not components of the ID.
- The ID can include descriptive elements such as material number, product name and type.
- ID is independent of technical realization; it should be realizable at least with Quick Response Code (QR Code), Data Matrix Code (DMC), Near Field Communication (NFC) and RFID (Radio-Frequency Identification). There may be further technical realizations in the future.

D.4 Information for identification

In the manufacturing sector, various classification and identification systems are used in order to unambiguously identify parts, products, people, software and services and thus also to make them technically workable.

Organisation name

Each organisation (company or manufacturer) internally ensures non-overlapping issue of identification marks. If the identification is stated together with the organisation name, unique identification can take place. A prerequisite is that the organisation name is also globally unique.

Product (type) name

A product is a usable and saleable asset resulting from production. Examples of superordinate product categories are services (e.g. development), software (e.g. computer program), hardware (e. g. valve) and process engineering products (e.g. lubricants).

Product (type) names are mostly identical across all manufacturers and identify the same class or category of product (for example servomotor, control unit, pressure sensor). In addition, manufacturers have specific additional abbreviations that specify the product (type) name even more precisely. They belong to the product (type) name (for example servo motor MKS140).

In addition to the product name and its abbreviation, each possible version (e. g. with options) receives an exact product and material designation. This is still a type whose version is precisely designated, however. It is identified by a unique material number.

Serial number (instance)

The serial number is a unique combination of ASCII letters, numbers and/or characters issued on a single occasion by the manufacturer.

It intends to distinguish a product instance from other product instances of the same product type.

For many asset types, no serial number is allocated (for example. screws, cables). Here, the naming of the manufacturer and product name is sufficient for identification.

D.5 Use of characters

1) Permitted characters

- Permitted: numerical (0...9) and alphanumeric (letters A...Z and a...z)
- Not permitted: national characters (e.g. ä, â, ...)
- To be avoided: blank, space, ! # $ % & ‘ ( ) * + _ ~ . , / : ; = ? @ [ ]

Certain characters identify and separate the individual segments of a URL and facilitate its processing:

- A question mark (?) introduces the specific path specifications (query string) of the URL
- An ampersand (&) acts as a separator for the parameter (data field)
- An equals sign (=) stands between the name of a parameter (= data field) and its value (= data content)

NOTE: When used, these characters must be previously encoded (see also RFC 3986).

2) String lengths

Some devices and software packages make it possible to transfer an almost infinite volume of characters. However, in transmission, some are limited to 245 characters.

To guarantee safe transfer, therefore, a maximum of 245 encoded characters should be transferred in a string.

D.6 ID implementation

Example of sample values:

Organisation with issuing entity:	7777777 (GS1)
	123456789 (DUNS)
	http://Firmaxy.com (DNS)
Product name:	Servomotor_MKS140
Material number:	1122334455
Serial number:	667788990012345

1) Typical implementation with URL

The issuing company is responsible for ensuring that information on the product is also retrievable under the Web address.

a) Example: Access to a unique serial number in the event of a service call

http://Firmaxy.com/667788990012345

http://Firmaxy.com/ID/667788990012345

http://Firmaxy.com?s=667788990012345

b) Example: Access to development documentation of a product/material

http://Firmaxy.com/Servomotor_MKS140/1122334455

http://Firmaxy.com/1122334455

http://Firmaxy.com?m=1122334455

2) Typical execution with ISO

The ISO/IEC schema allows a certain amount of freedom in selection of the Code Schema

Issuing entity:	e.g. UN for Dun & Bradstreet Application Family
Identifiers (AFI):	Object class, A1 for product identification
Data Identifier:	Structure of the ID String, 25S or 37S recommended

a) Example with Issuing Entity = UN, AFI = A1 and Data Identifier = 25S executed as RFID code

→ RFID, URI: urn:iso:id:obj:25SUN123456789Servomotor_MKS1401122334455667788990012345

b) Example with Issuing Entity = UN, AFI = A1 and Data Identifier = 37S executed as RFID code
→ RFID, URI: urn:iso:id:obj:37SUN.123456789.Servomotor_MKS1401 122334455+667788990012345

D.7 Technical realisations

The technical realisation of a code relating to the products can vary to a very great extent. Below are practical examples of implementation by companies participating in the Plattform Industrie 4.0. The examples show how today a link can be made from a QR code for an article to the data in the eShop.

Company A:

http://dc-qr.com?m=R911345469&t=HMU05.1N-F0140- 0350-N-A4-D7-N1N-NNN-N&s=7260403890047

Company B:

www.phoenixcontact.net/product/2832632

The seven-digit article number is allocated in the master data record upon creation of the article. In the PLM system this article number is supplemented by a revision number from the product documentation.

Company C:

Version 1: URL+“Enter“+serial number.
http://shop.murrelektronik.de/en/7000-14041-0000000
(Enter)
123456789123456789

Version 2: URL+“Enter“+serial number.

http://me23.me/ 7000-14041-0000000/123456789123456789
Version2 (24px=1,4cm)

Current (22px=1,3cm)

Company D:

http://go2se.com/Referenz
reference for an object. Multiple sections of data and requests per “/” with appropriate keywords for an object.

A Smart Manufacturing Component can therefore be simply linked, like our products by means of ref = ID or sn = ID

e.g. http://go2se.com/ref=TM241CE24T

http://go2se.com/ref=TSCEGWB13FA0

Or as part of an object data (though “data=” can also be replaced by other abbreviations).

(notional link)
http://go2se.com/sn=TM241CE24T/Data=(I4.0ID)

Company E:

Valve terminal VTUG

http://pk.festo.com/3s7pl9xL2QK

Proportional pressure control valve – VPPM

http://pk.festo.com/3S7PL9JS583

MS6-SV soft-start and exhaust valve

http://pk.festo.com/3S7PL810PFQ

Annex E

Representation in Semantic Technologies

For realisation of the global project "Semantic Web", the international ICT community and the World Wide Web Consortium (W3C) created a whole series of complementary standards and technologies. These are organised in the so-called "Semantic Web Stack"12F:

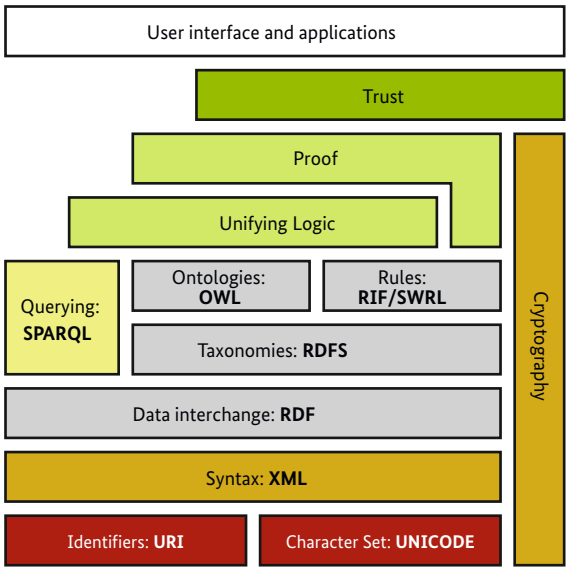
Some of the terms from Figure E.1 may be briefly explained:

URIs serve as globally unique identifiers of concepts. They are generally known in their special type, URL.

URL, e. g. http://www.zvei.org/Themen/Industrie40/Seiten/default. aspx.

- "http://" refers to the so-called URI Schema, so to speak the nature of the resource,
- "www.zvei.org/" refers to the general domain which is allocated on a globally unique basis by a recognised authority (e. g. Internet Corporation for Assigned Names and Numbers (ICANN)), and
- "Themen/Industrie40/Seiten/default.aspx" to the part of the resource/path that can be independently administered within the domain of an organisation.

Figure E.1 – Semantic Web Stack



Source: https://en.wikipedia.org/wiki/Semantic_Web_Stack

This example shows how globally unique identifiers can be easily created and how ICT services can be provided by the same means (e. g. via REST interface).

XML is used as syntax for data representation, thus a format, such that characters from a defined character set, e.g. UNICODE, which can be formed into valid data units and structures of these.

RDF, the "Resource Description Framework", is used for the formulation of logical statements regarding resources. It breaks each logical statement down into one or more "triplets" that correspond to the form "subject – predicate– object", each individual item being represented by a single URI for unambiguous ICT processing. With RDF, statements about individuals can be made, as for example in the table below about an item NBB1-3M22-E2:

Subject	Predicate	Object
NBB1-3M22-E2	isA	proximity sensor (AAA110)
NBB1-3M22-E2	isProducedBy	Pepperl und Fuchs
NBB1-3M22-E2	hasOutputCurrent	0.1 A
NBB1-3M22-E2	hasOutputDiameter	3 mm

The mapping of IEC 61360 properties in RDF is trivial and also possible in both directions as shown in the example below:

Subject	Predicate	Object
AAE867	hasPreferredName	output current
AAE867	hasSymbol	lopen
AAE867	hasPrimaryUnit	A
AAE867	hasDefinition	maximum dc output current of a semiconductor inductive proximity sensor at specified supply voltage
AAE867	hasDataType	LEVEL(MAX) OF REAL_MESURE_ TYPE
AAE867	hasFormat	NR2 S..3.3

RDFS, the "Resource Description Framework Schema", allows as a taxonomy definition as to how logical statements should be formulated for a certain knowledge domain by means of RDF. It depicts relationships and structures of knowledge, and together with RDF allows the formulation of simple ontologies. With RDFS it is possible to make statements about classes.

OWL, the "Web Ontology Language", lifts these definitions to the next level and allows the formulation of more complex relationships, structures and conditions for the formulation of more complex ontologies.

SPARQL, the "SPARQL Protocol And RDF Query Language" defines a language for the retrieval of content from ontologies, e. g. in RDF format. For example a request to list possible proximity sensors and their output currents that meet a size range, can be stated as follows:

Typical SPARQL request

```
PREFIX abc: <http://example.com/exampleOntology#>
SELECT ?current ?company
Where {
  ?x a abc:ProximitySensor .
  ?x abc:hasOutputDiameter ?y .
  ?x abc:isProducedBy ?company .
  ?x abc:hasOutputCurrent ?current .
  FILTER ( ?y < 4 )
}
```

Source: ZVEI SG Modelle und Standards

These standards and technologies constitute an ICT toolbox for which exist many different implementations and established processes for knowledge generation already (see for example: http://www.w3.org/wiki/SparqlImplementations, http://lod-cloud.net/, http://schema.org/).

Annex F

Reference Architecture Model Industry 4.0 (RAMI4.0)

F.1 General

The reference architecture model Industrie 4.0 (RAMI4.0) provides a structured view of the main elements of an asset using a level model consisting of three axes, as shown in Figure F.1. Complex interrelationships can thus be broken down into smaller, more manageable sections by combining all three axes at each point in the asset's life to represent each relevant aspect.

The three axes are:

- The architecture axis ("Layers"), with six layers to represent the information that is relevant to the role of the asset;
- The "Life cycle & value stream" axis to represent the lifetime of an asset and the value-added process, based on IEC 62890;
- The "Hierarchy levels" axis for assigning functional models to specific levels of hierarchy based on the IEC 62264-1 and IEC 61512-1 standards. The aim of the reference architecture model Industrie 4.0 (RAMI4.0) is to describe assets and combinations of assets with sufficient precision. The description in RAMI4.0 is a pure logical one.

Security and safety are elementary aspects. Therefore, they are included in RAMI4.0 in all layers and at any time. Therefore, they shall always be included in the description of each section of the three axes.

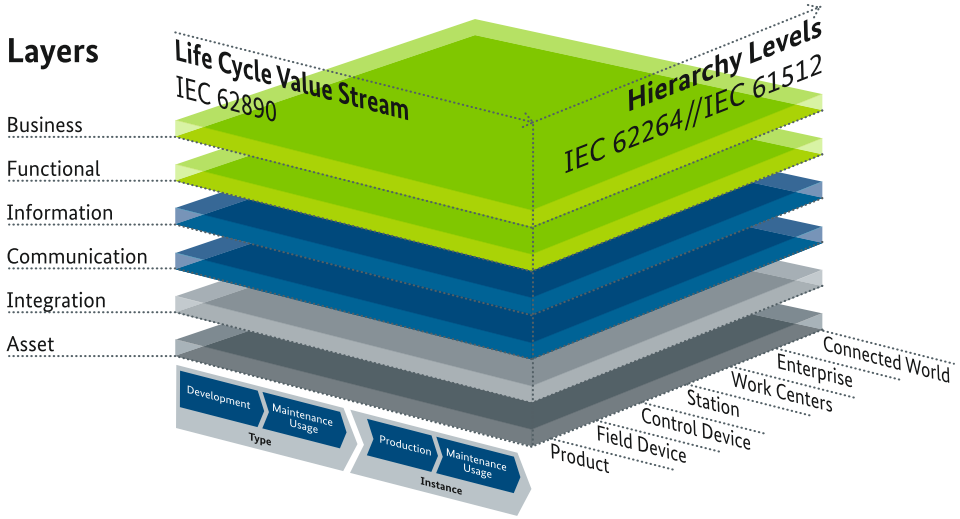
F.2 Architecture axis ("Layers")

F.2.1 Overview

The vertical axis describes the architecture in terms of properties and system structures with their functions and function-specific data in the form of layers with their outside behaviour.

Layers do not always have to have content. Layers can also be ignored if they are not relevant. So the term "layer" shall not be interpreted as a layer of the ISO/OSI Layer Model. They just represent a certain part of the asset behaviour. So there is a loose connection between the layers. Interactions may only take place between two adjacent layers or within a layer. Layers are never skipped but interactions may be passed through.

Figure F.1 - Reference architecture model Industry 4.0 (RAMI4.0)



F.2.2 Business layer

The Business Layer describes the relevant business processes with their framework requirements. This includes regulation and legislative requirements. It describes also the business related properties of the asset including contracting, licensing etc.

F.2.3 Functional layer

All (logical) functions and services of an asset (technical functionality) are assigned to the "Functional" Layer which follow the rules of Industrie 4.0. These functions obtain information from the data of the information layer and deposit them back to the information layer.

F.2.4 Information layer

Der Information Layer represents the information of an asset which is important to its function. Instead of considering data as integral part of functions in the past data are considered separately as information in data driven systems The Information layer contains all data associated to the functions of an asset including their storage location (i.e. a Cloud).

F.2.5 Communication layer

The Communication layer contains communication interfaces according to the rules of Industrie 4.0. It distributes information to other Industrie 4.0 component and gets information from them back. The inner part of the communication layer certainly follows the ISO/OSI 7-layer stack.

F.2.6 Asset and Integration layer

The Asset Layer represents the lowest level of RAMI 4.0. It reflects the physical world which has to be reflected in the information world by the upper 4 layers. The Integration Layer is a kind of adaption or transition layer. For example, it converts physical values into electrical one and in consequence into computer numbers according to a certain format. It also can have subroutines in order to send and receive information of a drive i.e.to control the speed of a drive (sub function).

Remark: any relevant changes in the physical world must be reflected in the information world which the Integration layer carries out. As an example, for such an event may be the change of an aggregate change of a mass flow (gas to liquid).

F.3 Life cycle & value stream axis

The "Life cycle & value stream" axis is used to describe an asset at a particular point in time during its lifetime from its production and value-added use right up to its disposal.

On this axis, the asset is characterized by its state at a particular time at a particular location.

F.4 Hierarchy axis

The "Hierarchy" axis is based on the reference architecture model for a factory along the lines of IEC 62264 1 and IEC 61512 1, the standards for integrating enterprise IT and control systems. To ensure a consistent consideration across as many sectors as possible from factory automation to the process industry, the terms "enterprise", "work centres", "station" and "control device" have been taken from the above-mentioned standards. The following hierarchy levels have been modified and supplemented to reflect the needs of Industrie 4.0:

- "Connected world" describes the relationship between an asset or combination of assets (such as an installation or company) and another asset or combination of assets (another installation or company), in other words, for example, a network of factories;
- "Field device" has been added as a hierarchical level;
- "Product" denotes the cooperating or collaborating product to be manufactured as an integral part of an Industrie 4.0 value-added process.

NOTE: For the asset "production plant" for example, the reference architecture model Industrie 4.0 thus enables a homogeneous consideration of the product to be manufactured and the production plant, with all their dependencies and relationships.

Annex G

Smart Manufacturing Standards Landscape

The Smart Manufacturing Standards Landscape project aims at building a common standards map where all relevant aspects for Smart Manufacturing are addressed.

Today standards are mostly classified by subject with their detailed information developed from dedicated committees or consortia.

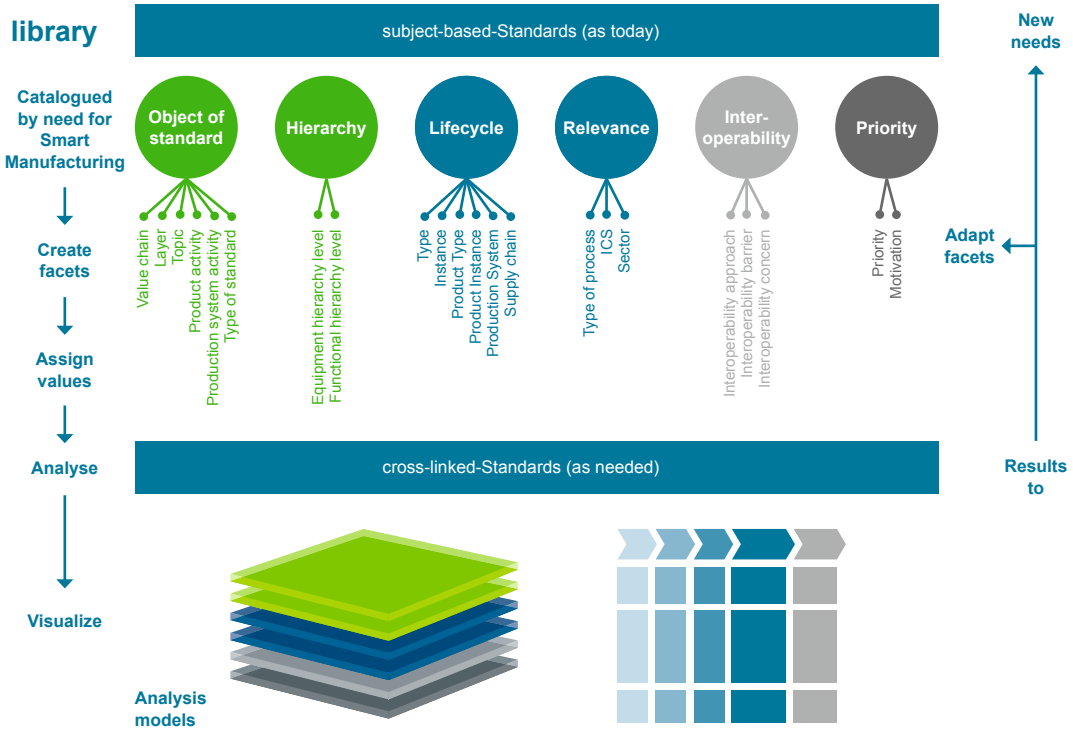
The framework developed for the Smart Manufacturing Standards Landscape enables the standardisation stakeholders to identify, for each standard publication or project, the characteristics related to its role and impact in the industry.

The framework consists of a standards library, a glossary of facets for the characterisation of these standards, and graphical representations for analysis as illustrated in Figure G.1.

The core item is the standards library, filled with standards from IEC, ISO, other SDOs[1], and specifications from consortia. By structuring the standards library with facets, the dependency between the standards is highlighted. This allows identification of gaps and redundancies between standards and facilitates decisions concerning standardisation projects and the use of standards. Any new insight may lead to adopt new facets which are possibly not known yet.

Different graphical representations can be generated from the structured standards library. The type of graphical representation depends on the view point that is needed for the investigation as shown in the following three examples.

Figure G.1 – Smart Manufacturing Standards Landscape framework



1 SDOs: Standards Developing Organisations

Example 1

The 2D-representation of Figure G.2 supports the analysis of the standards landscape in the context of the “product value chain” according to different “activities”.

Example 2

The RAMI 4.0 model may be useful to analyse the standard landscape according to layers, life cycle and hierarchy levels in a 3D-representation (Figure G.3). The following figure shows the positioning of IEC 61131-3 “Programmable controllers - Part 3: Programming languages” in this model.

Figure G.2 – 2D representation

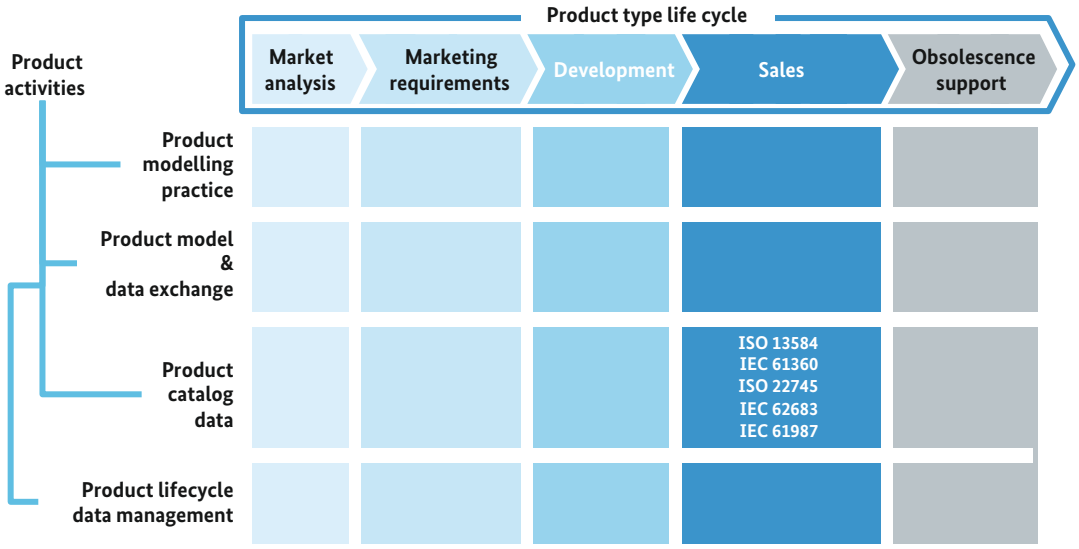
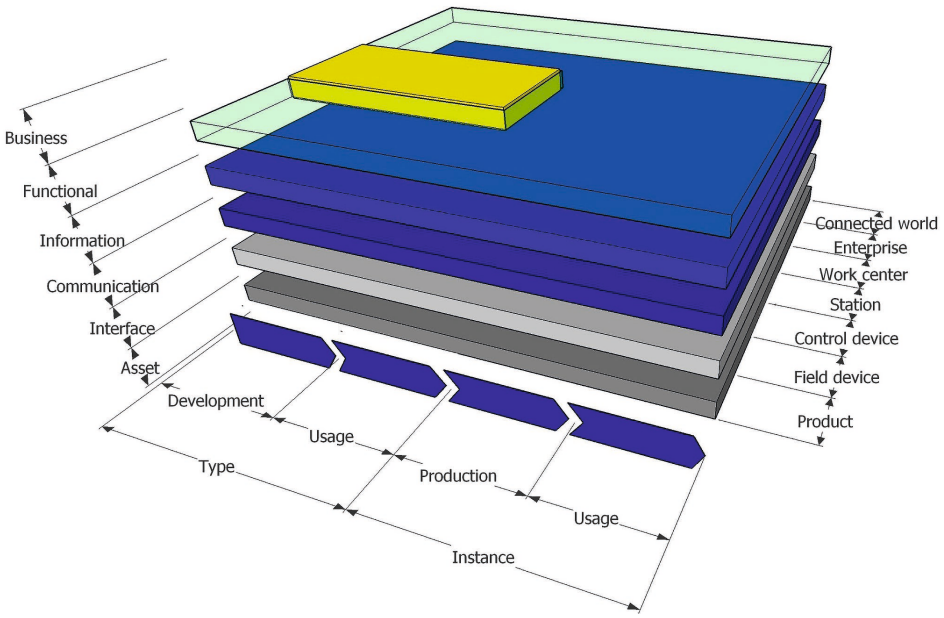


Figure G.3 – 3D RAMI representation

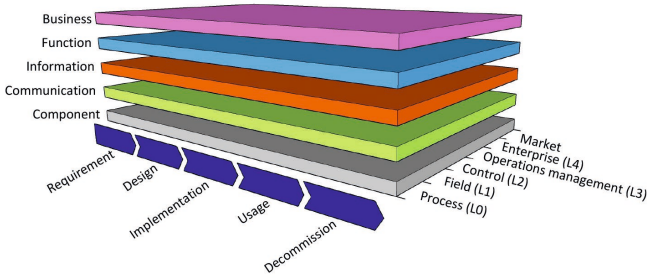


Example 3

The 3D-representation of Figure G.4 below relies on a model for analysis in the context of a "production system value chain". The three axis are:

- Production system life cycle (ISO 15704),
- Functional hierarchy levels (IEC 62264),
- Layers (SGAM[2]).

Figure G.4 – 3D production system representation



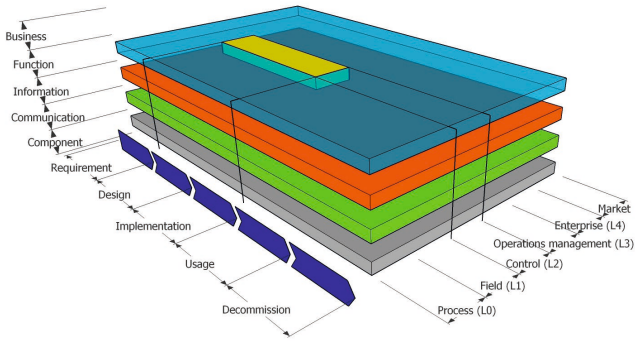
The following Figure G.5 shows the positioning of IEC 61131-3 "Programmable controllers – Part 3: Programming languages" in this model.

An initial version of the glossary of facets was agreed in 2017 by both Alliance Industrie du Futur and Industrie 4.0; and a first version of the standards library was developed accordingly. The 2017 edition merges the models developed in ISO (ISO/TC 184: "The Big Picture of Standards"), in Germany (RAMI 4.0) and in the USA (NIST, "Smart Manufacturing Standards Landscape").

The glossary of facets and the standards library have been initially proposed by France at ISO-IEC level together with a working process to maintain the standards library. The project has been accepted both by ISO and IEC. Since March 2018, it is being managed by the ISO IEC Smart Manufacturing Standards Map Task Force (SM2TF) under the leadership of France and Germany.

NOTE: The joint ISO-IEC/SM2TF is currently hosted by IEC/SEG 7 "Smart Manufacturing", and is intended to be transferred to its successor, the IEC Systems Committee (SyC) for Smart Manufacturing.

Figure G.5 – 3D production system representation of IEC 61131-3



2 SGAM: Smart Grid Reference Architecture (CEN-CENELEC-ETSI Smart Grid Coordination Group, November 2012)

Annex H

Bibliography

1. Technical overview: Secure identities, Publication of the Plattform Industrie 4.0, April 2016
2. https://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/secure-identities.pdf?__blob=publicationFile&v=7
3. Smart Grid Reference Architecture, CEN-CENELEC-ETSI Smart Grid Coordination Group, November 2012
4. Lightweight approach to the standardisation of vocabularies for semantic interoperability, Dr. Gökhan Coskun, Prof. Sören Auer, Fraunhofer IAIS
5. "Recommendations for implementing the strategic initiative Industrie 4.0" (Acatech) http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Material_fuer_Sonderseiten/Industrie_4.0/Final_report/Industrie_4.0_accessible.pdf
6. IEC 61850 Communication networks and systems for power utility automation
7. IEC TR 62390: 2005 Common automation device - Profile guideline
8. IEC PAS 63088:2017 Smart manufacturing - Reference architecture model industry 4.0 (RAMI4.0)
9. ISO TR 23087:2018 Automation systems and integration - The Big Picture of standards
10. Smart Manufacturing Standards Landscape, Alliance pour l'Industrie du Futur, 2017
11. Current Standards Landscape for Smart Manufacturing Systems, National Institute of Standards and Technology, U.S. Department of Commerce, 2016
12. IEC 61499-1:2012 Function blocks - Part 1: Architecture
13. IEC CDV 62890:2016 Industrial-process measurement, control and automation – Life-Cycle Management for systems and components

AUTHORS

Authors of edition 1 of Plattform Industrie 4.0

Dr.-Ing. Peter Adolphs, Pepperl + Fuchs GmbH | Prof. Dr. Sören Auer, IAIS (Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme) | Dr. Heinz Bedenbender, VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (GMA) | Meik Billmann, ZVEI (Zentralverband Elektrotechnik- und Elektronikindustrie e.V.) | Martin Hankel, Bosch Rexroth AG | Roland Heidel, Roland Heidel Kommunikationslösungen e.K. | Dr.-Ing. Michael Hoffmeister, Festo AG & Co. KG | Haimo Huhle, ZVEI (Zentralverband Elektrotechnik- und Elektronikindustrie e.V.) | Michael Jochem, Robert Bosch GmbH | Markus Kiele-Dunsche, Lenze SE | Gunther Koschnick, ZVEI (Zentralverband Elektrotechnik- und Elektronikindustrie e.V.) | Dr. Heiko Koziolek, ABB AG | Lukas Linke, ZVEI (Zentralverband Elektrotechnik- und Elektronikindustrie e.V.) | Reinhold Pichler, DKE (Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE) | Frank Schewe, PHOENIX CONTACT Electronics GmbH | Karsten Schneider, Siemens AG | Bernd Waser, Murrelelektronik GmbH

Authors of edition 2 of the trilateral cooperation between France, Germany and Italy

Nikolai D'Agostino, CENIT AG | Enrico Annacondia, UCIMU (UCIMU-SISTEMI PER PRODURRE, Italian machine tool, robots, automation systems and ancillary products (NC, tools, components, accessories) manufacturers' association) | Alexander Bentkus, DKE (Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE) | Giacomo Bianchi, ITIA- CNR (Istituto di Tecnologie Industriali e Automatione, institute of the National Research Council) | Joseph Briant, Schneider Electric | Roland Heidel, Roland Heidel Kommunikationslösungen e.K. | Dr.-Ing. Michael Hoffmeister, Festo AG & Co. KG | Dr. Patrick Lamboley, Schneider Electric | Ruggero Lensi, UNI (Ente Italiano di Normazione) | Gernot Rossi, Siemens AG | Dr. Marco Mendes, Schneider Electric Automation GmbH | Domenico Squillace, UNINFO (Ente di normazione Federato all'UNI) | Hadrien Szigeti, Dassault Systèmes | Nicole Schmidt, Daimler Protics GmbH | Philippe Tailhades, Gimélec | Ingo Weber, Siemens AG

www.plattform-i40.de

www.industrie-dufutur.org/contact

www.sviluppoeconomico.gov.it/index.php/it/industria40