PLATTFORM INDUSTRIE 4.0

WORKING PAPER

Relationships between I4.0 Components – Composite Components and Smart Production

Continuation of the Development of the Reference Model for the I4.0 SG Models and Standards

In Cooperation with

ZVEI: Die Elektroindustrie

Imprint

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

Text and editing
Plattform Industrie 4.0
Bertolt-Brecht-Platz 3
10117 Berlin

Design and production
PRpetuum GmbH, Munich

Status
June 2017

Illustrations
Fotohansel – Fotolia (Title)

This brochure is published as part of the public relations work of the Federal Ministry for Economic Affairs and Energy. It is distributed free of charge and is not intended for sale. The distribution of this brochure at campaign events or at information stands run by political parties is prohibited, and political party-related information or advertising shall not be inserted in, printed on, or affixed to this publication.

The Federal Ministry for Economic Affairs and Energy was awarded the audit berufundfamilie® for its family-friendly staff policy. The certificate is granted by berufundfamilie gGmbH, an initiative of the Hertie Foundation.

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 Preliminary remarks

This document was created between May 2016 and May 2017 as part of discussions held by the ZVEI SG Models and Standards group. It was subsequently approved following extensive discussions by working group 1 (AG1) of Plattform Industrie 4.0.

For better readability, the abbreviation 'I4.0' is used throughout to stand for 'Industrie 4.0' in compound terms. As a stand-alone term 'Industrie 4.0' continues to be used.

For the sake of consistency with DIN SPEC 91345, the term 'Asset' is used rather than 'Object' or 'Thing'.

1.1 Objective and methodology of this document

The RAMI4.0 model can be used to describe any I4.0 asset. The I4.0 component allows you to create an information technology link, using the administration shell, between any asset and Industrie 4.0.

This document aims to describe an information technology structure that can be used to interrelate various I4.0 components and then organise these components into composites for specific purposes. Such organisation, including for example modularisation based on the principle of nestability [1, page 59] should be feasible not only at a one-dimensional level, but also according to **various organisational criteria, considerations and engineering disciplines**. For compatibility with the defined specifications, it must also be possible to map this information technology structure as an administration shell.

The observations contained in this document apply in equal measure to the factory automation and process automation industries. Terms such as 'factory', 'production' and 'shop floor' thus also refer to the facilities of the process technology industry.

2 Relevant content from various sources

This section contains important content from previous discussions or from other working groups. It is intended to illustrate and promote links with other topics. No new content is introduced here.

2.1 Concept of sub models

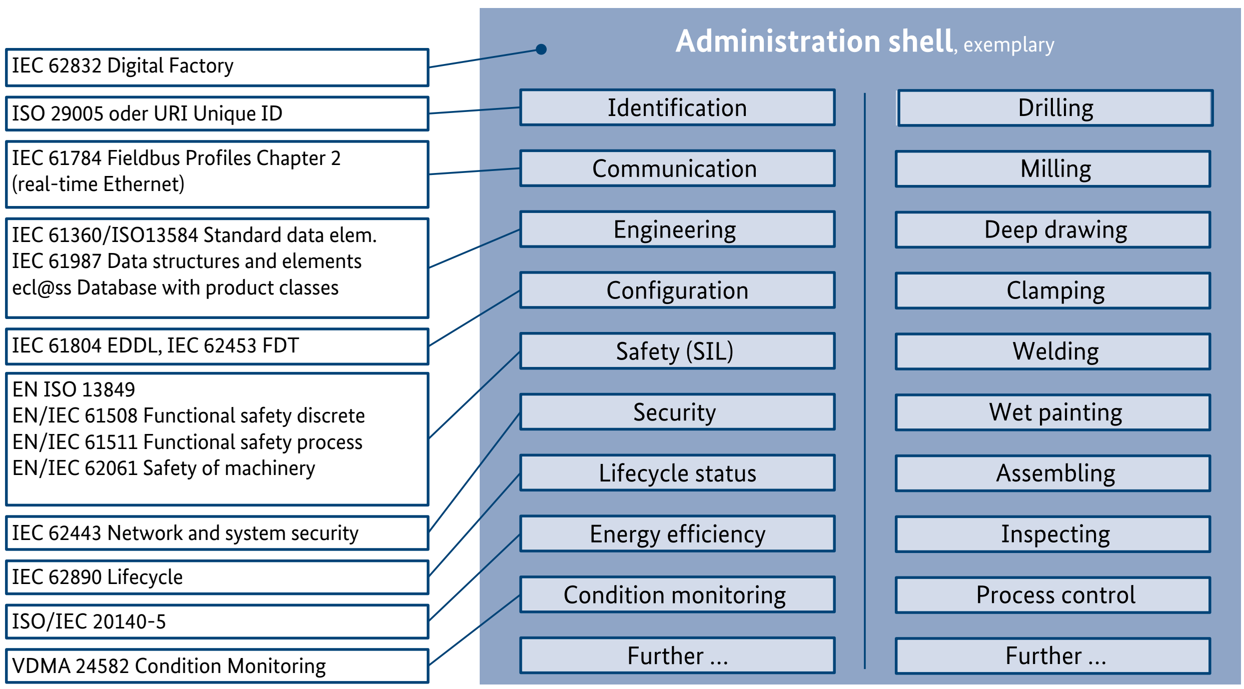
The basic idea behind the I4.0 component is to wrap each I4.0 asset with an 'administration shell', which is designed to provide a minimal but adequate description of each I4.0 asset or application in the information world. It must therefore also be possible to classify existing standards appropriately in the relevant administration shell.

For this reason, each of the administration shells comprises a series of 'sub models', which represent different aspects of the relevant asset. These aspects could describe safety or security, for example, but could also cover various process capabilities such as drilling or assembling.

The aim is to standardise just one sub model for each individual aspect/technical domain. In this way, a drilling machine can be found among many other I4.0 components, because its administration shell has a 'drilling' sub model with appropriate properties. For cooperation between assets, consequently certain properties can then be assumed to exist.

The various sub models complement each other in terms of describing different aspects of the relevant asset. In the drilling machine example, a second sub model 'Energy efficiency' could then describe the ability of the drilling machine to save power during production stoppages.

Figure 1: Administration shell composed of sub models that are relevant to the particular asset



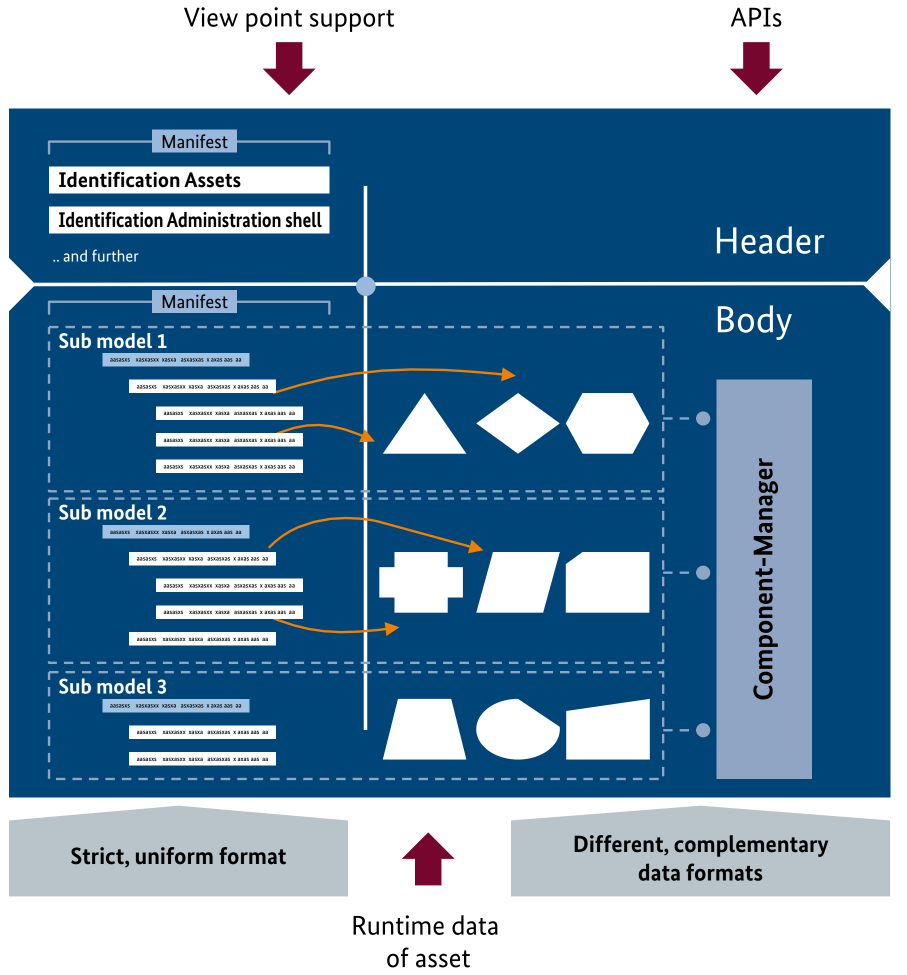
2.2 General structure of the administration shell

The last published document on the administration shell [2] presented a general, logical view of its structure. Marked in blue in the illustration below, the administration shell comprises a 'Header' and 'Body'. The 'Header' contains identifying details about the administration shell and the represented assets. The 'Body' contains a large number of sub models in order to shape the administration shell according to specific assets.

Each sub model contains a structured quantity of properties that might refer to data and functions. A standardised format, which is based on IEC 61360, is provided to describe the properties. Data and functions may be available in a variety of different, complementary data formats.

The properties of all sub models thus always form a legible table of contents or manifest of the administration shell and therefore of the I4.0 component. As a prerequisite for binding semantics, administration shells, assets, sub models and properties must each be uniquely identified globally. Authorised 'global identifiers' are ISO 29002-5 (used for example for eCl@ss and IEC Common Data Dictionary) and URIs (Unique Resource Identifiers, for example for RDF ontologies) [2].

Figure 2: General structure of the administration shell



3 Relationships

This section defines the concept of relationships between individual elements in Industrie I4.0 and the associated advantages.

3.1 General information on types of cooperation

In engineering, combining individual components results in a new, higher functionality compared to that offered by the individual components of a system. This means that the individual components, e.g. machines, cooperate with each other during operation. We are of course already familiar with "Human to Human" cooperation. The idea of human-machine cooperation has also been around since 1844 at least, the time of the Weaver Revolt. This type of cooperation is important, but just one of the aspects we must consider in relation to Industrie 4.0, which has also seen the emergence of machine-machine type systematic cooperation. We must examine this type in more detail since this is what transforms human-machine cooperation into human-machine-machine cooperation.

3.2 Structured asset description

In Industrie 4.0, only objects of value are considered and designated as "assets", irrespective of their form of appearance. The concept of the I4.0 component links the asset with the information world. This duality is also important for the basic concepts of relationships.

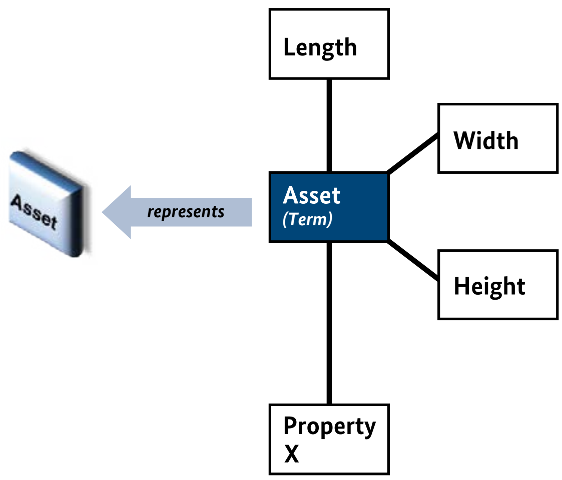
3.2.1 Terms

The physical world consists of objects that are objects of value, or assets, in Industrie 4.0. An asset can be understood in Industrie 4.0 once it has been assigned a name or term that can be characterised by means of properties. A term is characterised by the following features:

- Identifier
- Term designation (name, e.g. asset name) is often just called a "term". This document uses "designation" for clearer differentiation.
- Term definition
- Characteristic(s)

An asset from the physical world is represented by a term and characterised by well defined properties e.g. "length", "width", "height" , "colour" etc. (Figure 3).

Figure 3: Properties are assigned in order to characterise an asset in the physical world.



Before they can be reflected in the information world, the assets must be characterised by means of "terms". The asset represented by the term and its characteristics must be displayed in the information world in terms of data. The methodology used to characterise an asset by means of properties is called the property principle in Industrie 4.0.

3.2.2 Properties

A property is characterised by the following features:

- Term designation (name) is often just called a "term". This document uses "designation" for clearer differentiation.
- Identifier (code),
- Term definition

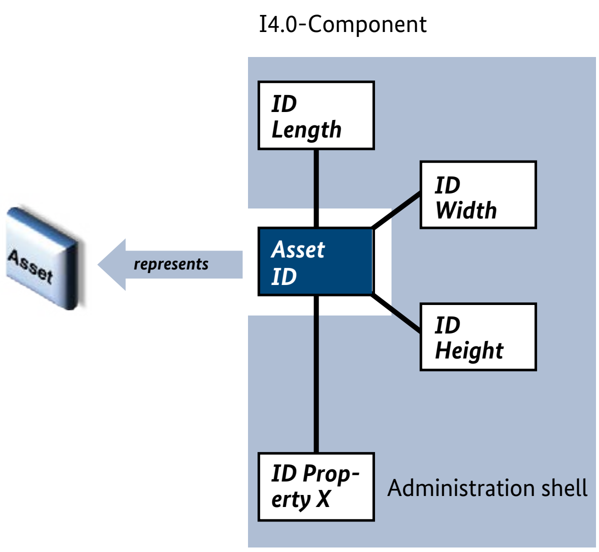
- (Binary) data-related representation of the features (s) with attributes and references

For the property definition (property type) the "Value" attribute is not assigned. The "Value" attribute is populated for the property instance.

As a result, a property is created in the information world comprising at least one term (-designator) with an identifier.

Before an object can be used as an asset in Industrie 4.0, it must be described in the information world using machine-processable properties, as displayed in Figure 4.

Figure 4: The properties from the physical world that characterise an asset are stored in the administration shell of the I4.0 component together with their identifiers (ID).



Properties are generated from terms as follows:

1. The relevant characteristics of an asset are listed for the I4.0 application

2. Properties and identifiers are generated from the characteristics with or without value assignments, whereby properties without a value assignment represent "Property types" and properties with value assignments represent "Property instances". Functions can also represent a property in this process.

Therefore initially the characteristics of a real asset are described with terms. A property within the meaning of Industrie 4.0 is created when a term is specified digitally as a property according to IEC 61360 or ISO 13584-42. It thus describes a specific feature of an asset from the physical world in the information world. If a term is allowed to have only one semantic occurrence in the domain of Industrie 4.0, semantic ambiguity is avoided using the identifier. A production unit in Industrie 4.0 therefore knows that the term "Jaguar" refers to the car being manufactured, not the animal.

Properties that meet these requirements include properties from eCl@ss, from the IEC 61360 Common Data Dictionary (IEC 61360 CDD) and some other sources. An exhaustive treatment of forming properties is beyond the scope of this paper. However, in order to understand the technology of Industrie 4.0, it is necessary to consider the formation rules of the data model according to IEC 61360/ISO 13584-42.

3.3 Cooperation between assets

According to the German definition in Wikipedia [Kooperation], cooperation is the process of two or more organisms, persons or systems working or acting together for mutual benefit. This definition contains all aspects that are relevant to relationships in Industrie 4.0: People (human) and systems (assets and composites of assets). To gain a better understanding of the type of cooperation to be achieved in I4.0 applications, we must examine the application scenarios in more detail.

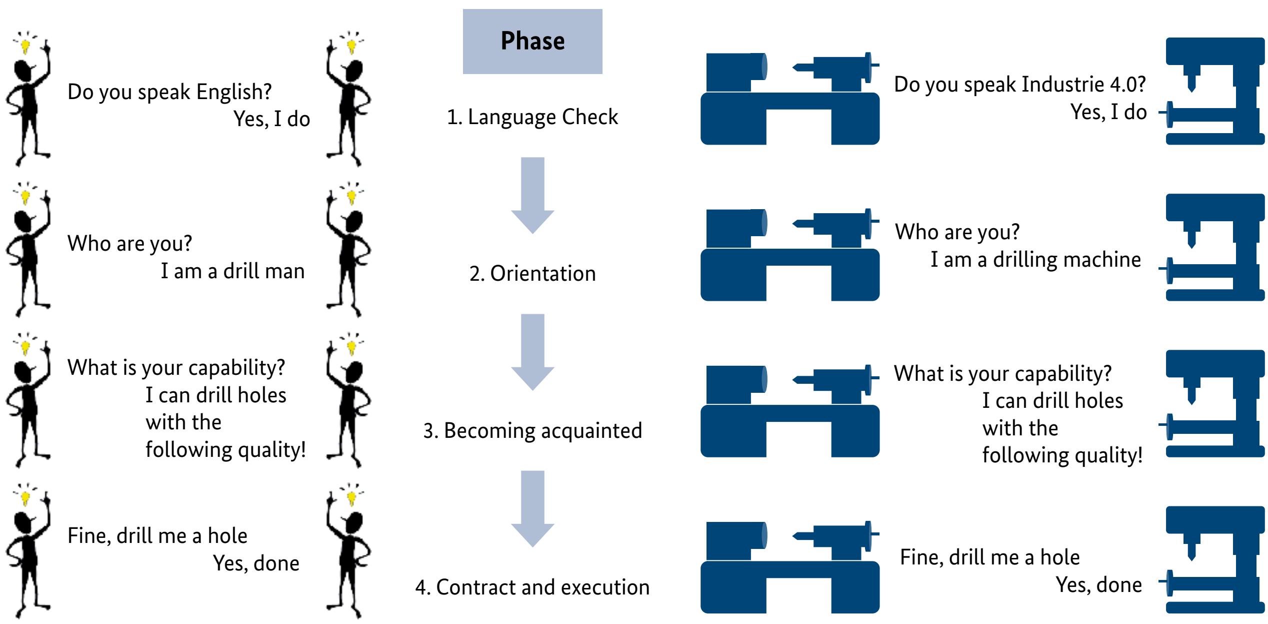
For this purpose, we must first envisage the main process involved in building relationships between assets, using two machines as example.

Figure 5 shows, by means of two machines (assets), the main procedure for generating types of cooperation (right side) using services as the equivalent of setting up cooperative relationships between people (left side). During a "becoming acquainted" phase, tests are run to check how and whether the parties can communicate (1). This includes establishing the communication link and agreeing on the language necessary for communication. The capabilities of the counterpart are then queried (2). If these meet the other party's requirements, the familiarisation phase is completed (3), the order issued and executed and its completion is reported (4).

3.3.1 Key basic services for checking the abilities of assets

The mechanism described is so general that it could apply to all machines in a machine pool that are intended to form a temporary production line to manufacture a particular product. Depending on the order, this line should create the optimum configuration to manufacture a particular product, which is automatically manufactured by connecting the machines required in each case (Production Units PU) under the control of a Production Manager (PM) mainly executing MES functions. Essentially this process consists of automatically creating cooperative relationships between suitable I4.0 components with the subsequent automatic execution of functions.

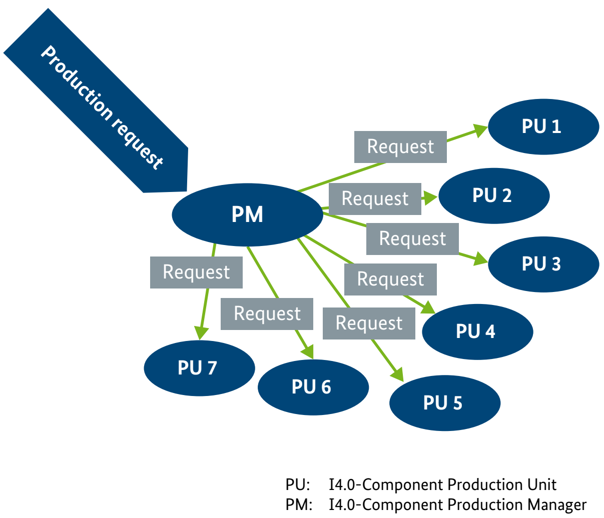
Figure 5: "Becoming acquainted" principle with subsequent team activity for people (left) and its equivalent for assets (right)



3.3.2 Scenario example of an flexible production line

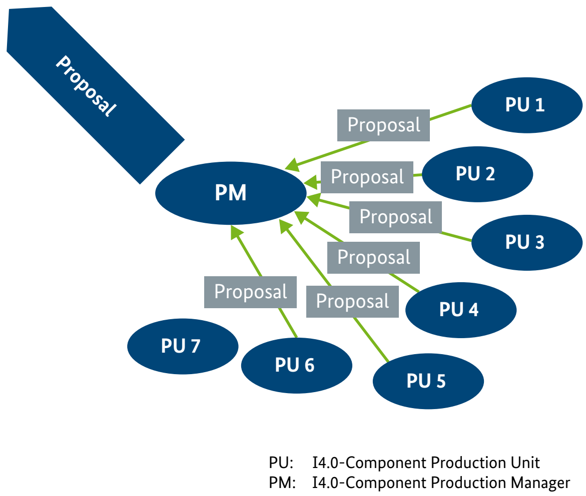
Based on [1], the figures 6 to 9 illustrate the principle of creating, configuring and displaying the decomposition of a flexible production line using the requirements to manufacture a particular product. The production line consists of a series of production units (PU) and a production manager (PM). Both instances are I4.0 components with information and features structured according to RAMI4.0. A PM's function comprises parts of an ERP system and parts of an MES system, the parts of which can reside in the PM and in the PUs.

Figure 6: Production request and checking the production resources



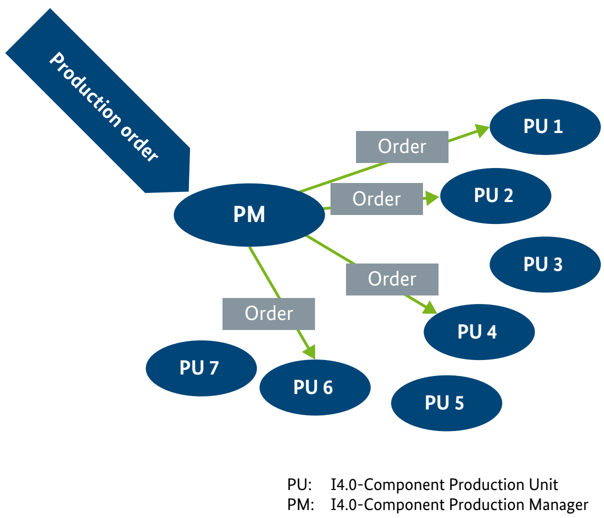
According to Figure 6, the Production Manager accepts a production request and, based on the services displayed in Figure 3, right side, checks the production options with the abilities of the PUs available in the pool using requests to the PUs. This check based on the PU quote includes the availability of the PUs and the price of each production step to enable an order price to be determined (Figure 7).

Figure 7: Manufacturing quote of the PUs. Not all PUs have to submit a quote



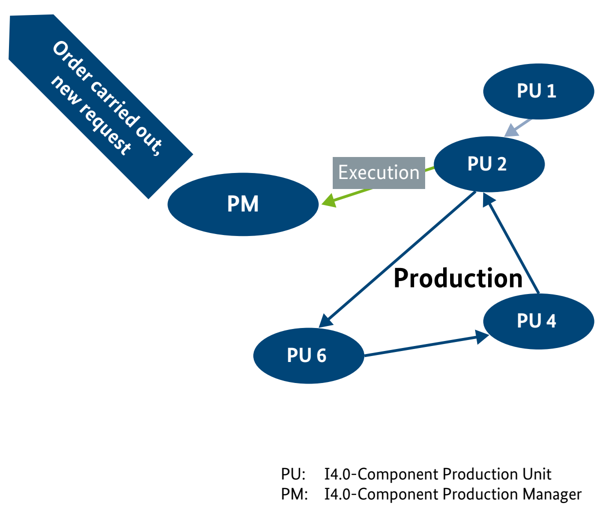
Not all PUs have to submit a quote in this process. After (automatic) clarification of all business conditions with the customer based on the issued quote, the customer awards the order (Figure 8), the PM sends an (electronic) order confirmation to the customer and then occupies the reserved PUs with corresponding orders. The part to be manufactured can also itself intervene in this process.

Figure 8: Awarding the production order



Once all orders have been completed by the PUs, the PM reports completion and is ready for a new request (Figure 9). If the setup is appropriate, the PM can of course also process a new order while the previous order is being executed.

Figure 9: Production process with completion report



The essential elements of this broad scenario can be distilled as follows:

First, the PM is as an instance that accepts a digital request after production of a product under human control e.g. in a lot size of one. This means that a computer operating as a customer uses a request service of the PM. The PM's algorithm determines the necessary production functions and uses requests to scour the PU pool *(production resources)* of the *"shop floor"* for suitable and available PUs. If the PM is successful and if certain basic conditions are fulfilled, e.g. availability of the PUs and their price per manufacturing step, and if approval from the responsible person in the relevant department is available, the PM sends the customer a digital quote. If the customer accepts this quote, the PUs are organised into a production line. They manufacture the product and report completion to the PM. The PM then triggers the commercial processes of the business at the *"Office Floor"* company level, which then result in delivery of the manufactured product and collection of payment.

Elaborate processes are clearly required to achieve a collaborative human-machine-machine platform.

3.4 Composites of assets and complex of relationships

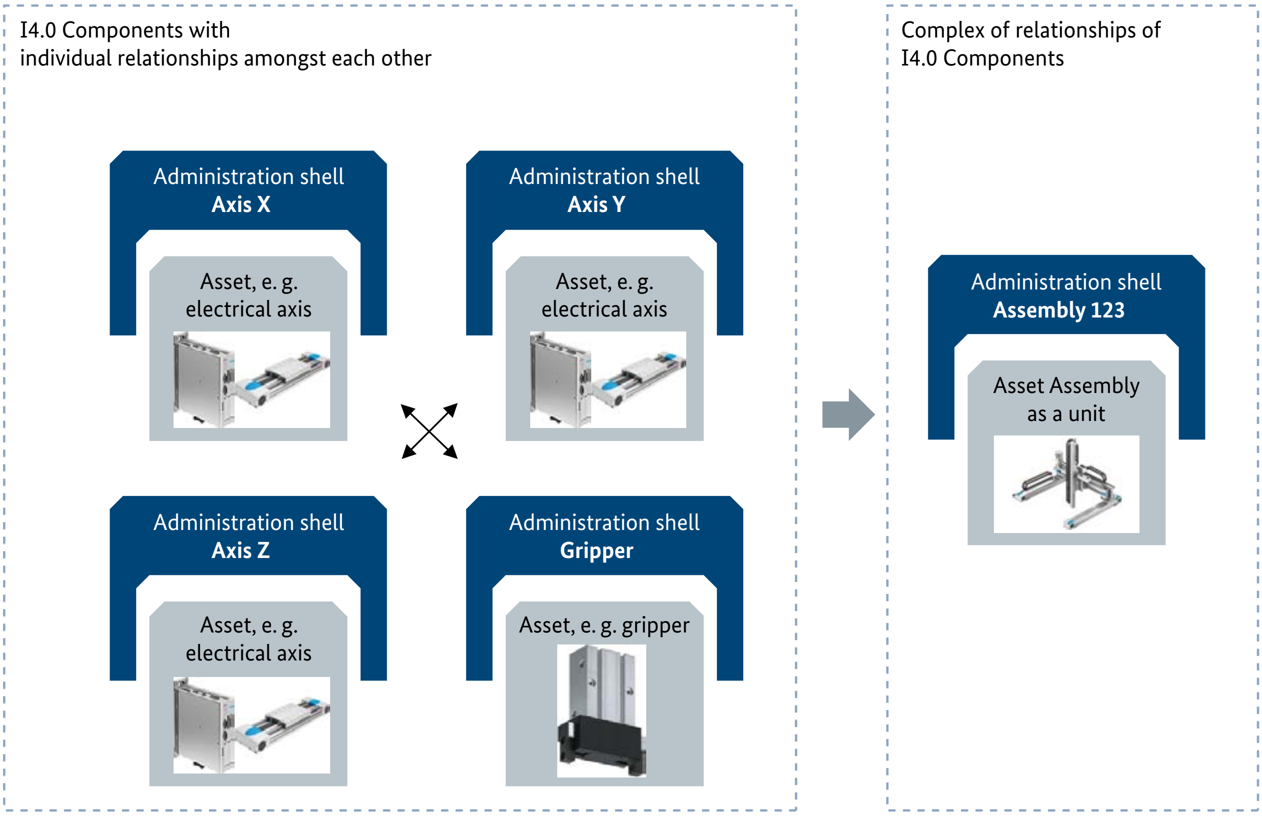
If assets are to cooperate with each other (see Section 3.3), they must be connected. Such composites of assets nowadays are generated using a standard engineering process. However, from an I4.0 perspective, a composite created in this way is considered to be static. This is because the dynamic behaviour of Industrie 4.0, which enables assets to self-locate and to self-organize, cannot be achieved with today's engineering methods.

To create the "team" mentioned in the previous section that is necessary for cooperation, we use the connections between assets in which people are incorporated as assets: that is we group assets together to form a composite asset. Since a composite asset itself represents a separate asset, it is subject to the same mapping rules as each individual asset. Composite assets therefore automatically exhibit the same data-related structure with unified semantics of the information as the individual assets. They are therefore also I4.0 components.

A composite asset in the physical world thus in the information world represents a complex of relationships made from I4.0 components. Establishing relationships between the I4.0 components in this way results in a complex of relationships with new, higher functionality. In other words, a complex of relationships forms its own I4.0 component with its own administration shell. Figure 10 shows this situation.

The new abilities of this type of composite of assets are consequently represented by appropriate sub models.

Figure 10: Several I4.0 components with relationships constitute, as a complex of relationships, a new I4.0 component with an administration shell



3.4.1 Asset hierarchies

Since the features of an asset can be described in any level of granularity, the engineering task can be started as soon as a minimum quantity of information is available (Figure 11). Subsequently, the modelling can be extended to include parts of the asset, or assets can be aggregated (treated as one). This allows for a highly flexible system design.

Figure 12 shows the methodology based on the recursive description of assets with RAMI 4.0. If assets are hierarchically arranged, that is, if an asset is formed from several subassets (detailed composition) for example, the arrangement remains consistent due to the uniform description methodology. Similarly, assets can be grouped together and represented by a higher-level asset (aggregation). In this case, the sum of the features of two interconnected assets is greater than the sum of their individual features.

The following applies: In Industrie 4.0, combining two or more assets results in a new asset with a corresponding administration shell. Thus a new I4.0 component is created.

Figure 11: Work on system design can be started at any level of granularity, structured according to RAMI 4.0[1]

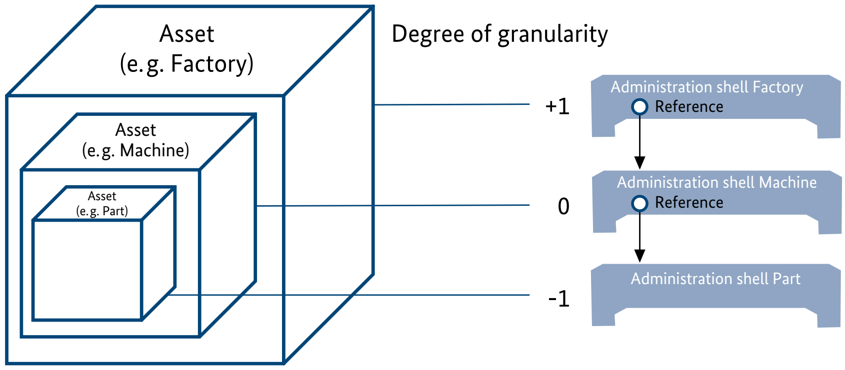
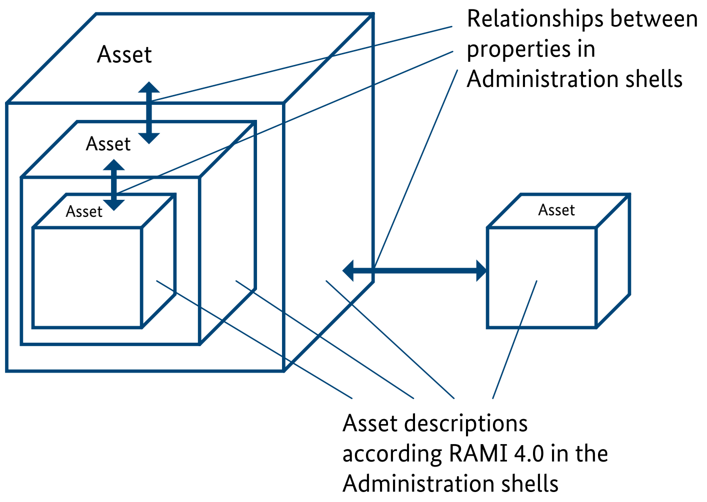


Figure 12: Relationships between assets are described by means of property relationships[1]



1 For the sake of simplicity, the assets structured according to RAMI 4.0 appear as simple cubes in the illustrations.

3.4.2 Connections and relationships

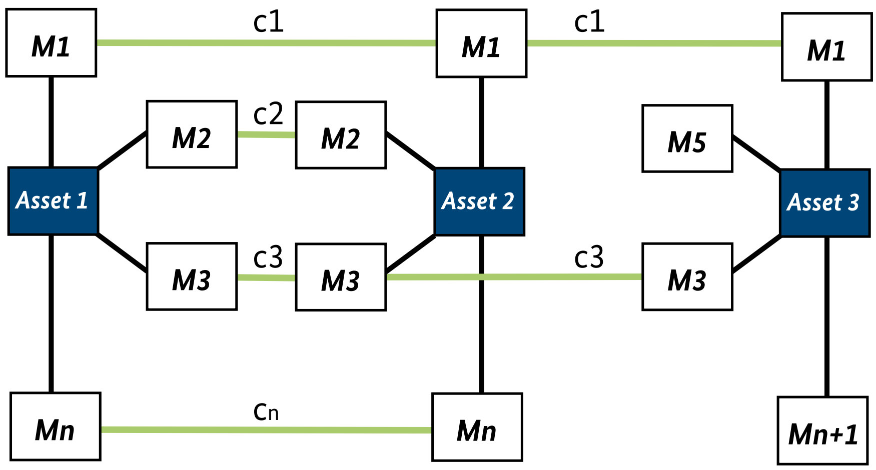
Figure 13 shows three assets that are interconnected in the physical world. Even with the current technology, these are not simply interconnected during the engineering phase. Instead the terms representing these assets are linked together. The properties that uniquely describe each of the interfaces are interrelated with each other (green lines in Figure 13). This situation often is not described explicitly, but is important in the context of Industrie 4.0. In the information world, the connections between the assets are expressed by interrelating the corresponding properties with each other (red lines in Figure 14). Connections between terms representing assets in the physical world become relationships of properties in the administration shell, which are formally mapped so they are machine-processable. It is important to remember that a property also can link to a function (see 3.4 in [2], 3.2.2).

Note: In the information world, it is not enough to simply describe connections between assets, e.g. using OWL. These connections must be formally interrelated by means of properties so they are machine-processable. Since properties represent the characteristics of assets in the information world, the relationships between the properties of the corresponding I4.0 components are displayed as shown in Figure 12. This applies to all types of relationships.

Figure 13 illustrates the situation in the physical world. The line (c1) e.g. for the electrical energy supply is connected to all assets, whereas the function M2 of asset 1 only connects with the matching function M2 of asset 2 (c2). The communication interfaces (c3) of all three assets are also all connected with each other. This is reflected in Figure 14 with the relevant relationships for the information world. In the information world

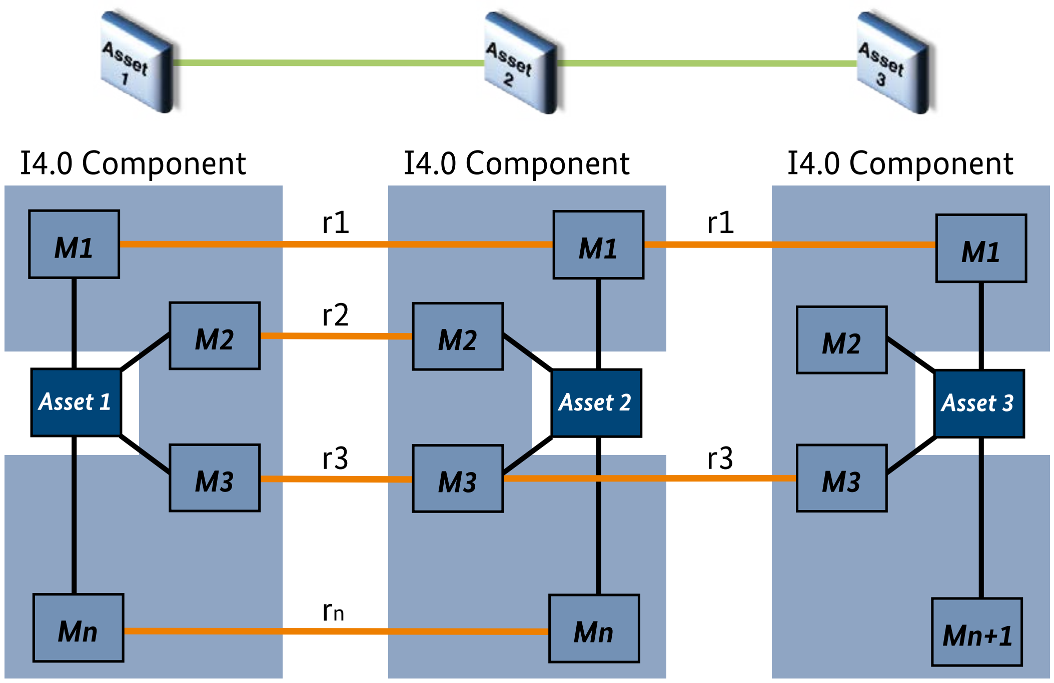
- *static* relationships describe the arrangement as such,
- *dynamic* relationships describe the cooperation between I4.0 components during operation.

Figure 13: In the physical world, assets are connected by means of their properties (P) in the information world (c)



Bn = term n, c = connection

Figure 14: In the information world, the *connections from the physical world* between assets are represented by relationships (r) between the properties (M) stored in the administration shells of the I4.0 components



Mn = property n, r = relationship

3.4.3 Modelling relationships in a relationship table

Relationships that are formalised in this way can already be digitally mapped using simple means. The relationships shown in the above figure can be expressed in a table, for example:

Table 1: Modelling relationships in a relationship table

Relationship	Asset X Property	Asset X Property	Asset X Property
r1	(Asset 1 X M1)	(Asset 2 X M1)	(Asset 3 X M1)
r2	(Asset 1 X M2)	(Asset 2 X M2)	
r3	(Asset 1 X M3)	(Asset 2 X M3)	(Asset 3 X M3)
rn	(Asset 1 X Mn)	(Asset 2 X Mn)	

3.4.4 Relationships are assets

The main purpose of relationships is to enable cooperation between assets. However, these relationships cannot be set up without the appropriate infrastructure. Relationships must be established, maintained, severed and modified. This applies to everything from simple screw fastenings to complex long distance traffic systems via public and private networks. Consequently, this type of infrastructure in the information world also constitutes an I4.0 component having its own administration shell. Figure 14 shows, for example, a relationship created between I4.0 component 1 and I4.0 component 3 using the communication I4.0 component 2. In this case, the energy connection for all three assets is virtually displayed as relationship r1 between the relevant administration shells of the relevant I4.0 components. In the simplest case of a communication relationship, the connection asset is only a line with protocol specifications, for example. The same applies to energy supply e.g. in relation to voltage, maximum current and frequency.

Note: Like all other assets, a communication infrastructure also has a lifetime ("vita"), which is split into a "type" phase and an "instance" phase in accordance with RAMI4.0. For the sake of simplicity, no distinction is made between the two phases when we refer to relationships in this document. However, when setting up an arrangement, this distinction must of course be made, as for every asset.

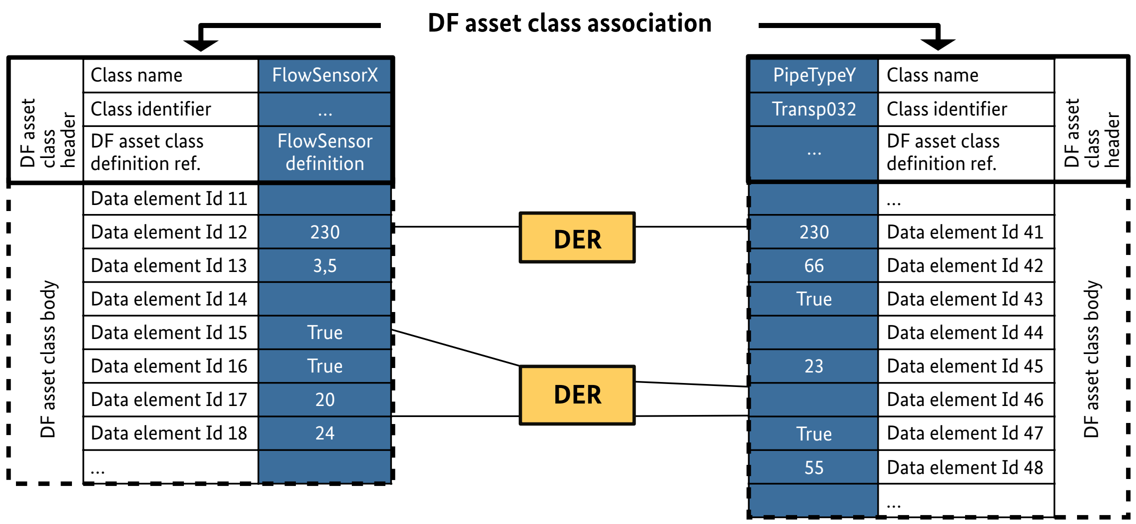
3.4.5 Relationships between DF assets according to IEC 62832

Initial partial results on the Digital Factory are provided in IEC TS 62832-1:2016, which was successfully introduced as a standardisation framework in autumn 2016.

In this framework, property types are called data element types, property instances are called data elements and relationships between properties are called data element relationships (DER). The document refers to assets as PS_Assets, while the terms for representing assets are called DF Assets. These correspond to the I4.0 component in Industrie 4.0. Figure 16 shows that connection types (asset class association) are defined for connections between assets at type level. The relationships between the properties in this case can be defined as data element relationships (DER). A connection between assets is described as an asset link in which the DER can be evaluated. If the property relationships are positively verified in accordance with the rules of a DER, the connection between the assets in the physical world can be established. This also applies in Industrie 4.0, except for the designations that are different for historical reasons. In terms of content, Industrie 4.0 is thus fully compliant with the results of the IEC TC65.

The following applies in Industrie 4.0: If a relationship is created as a type between I4.0 components and is verified according to specially specified rules for compliance in all endpoints, it is considered to be a relationship instance.

Figure 15: Data Element Relationship (DER)s describe relationships between assets (DF asset classes) for creating composite components [3]



3.4.6 Types of relationship and views

Relationships between assets can be considered and described from different views. Possible views for asset relationships are:

- Business relationships
- Constructive relationships
- Functional relationships
- Data relationships
- Communication relationships
- Integration relationships with subrelationships
  - Energy
  - Mechanics
  - Material flows
- Location relationships
- Temporal relationships
- State relationships
- Asset relationships
- Human-asset relationships

This applies within an asset for all subassets in the same way as between assets.

The following applies to all relationship types:

- Relationships describe application-specific connections between assets or their I4.0 components.

- A relationship between I4.0 components is characterised at least by its relationship type with its related properties. It has two or more endpoints.

- An endpoint is characterised in that it is related to an I4.0 component and that the relevant properties including their values are compatible with both I4.0 components. These properties are known as endpoint properties.

- Relationships always have a unique identifier.

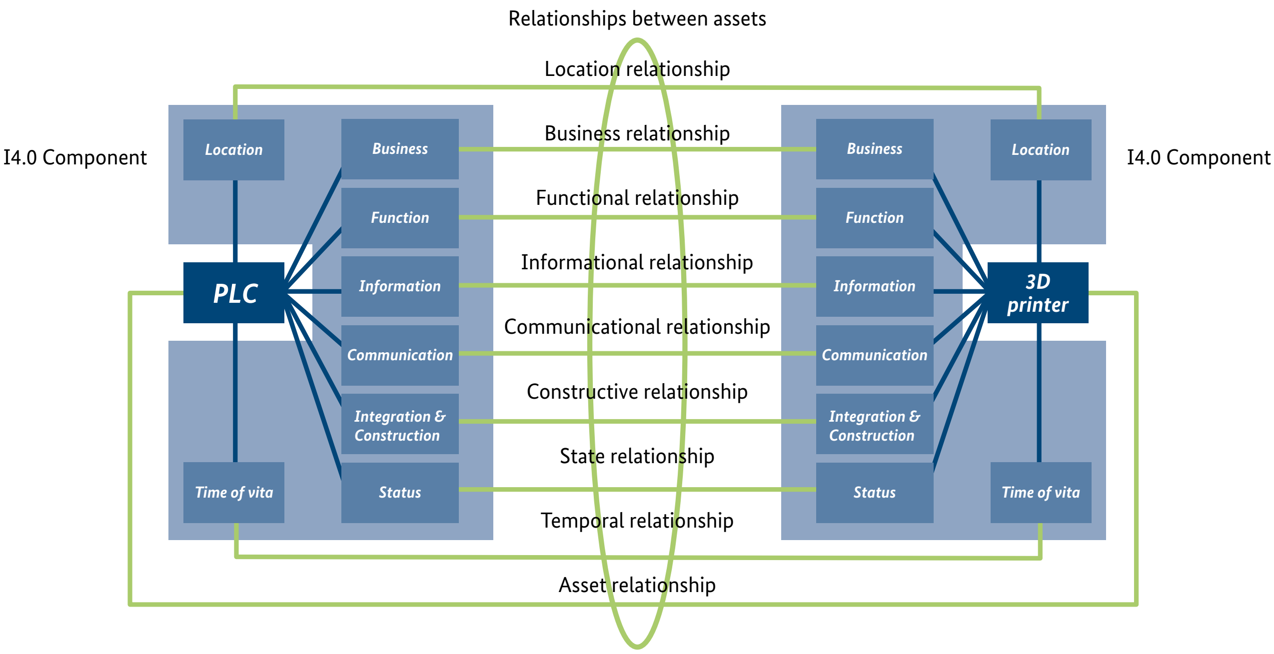
- To enable assets to cooperate with each other based on relationships, a common interaction language, mainly based on properties, is required.

The relationship types explained below Figure 16 are compliant with the basic views introduced in DIN SPEC 91345 [5].

Table 2: Basic views defined by DIN SPEC 91345 [5]

Basic view	Best practice/example
Business	Data and functions are deposited which allow judging on the business suitability and performance of a component in the life cycle phases Procurement, Design, Operation and Realisation. Examples: prices, terms of delivery, order codes
Constructive	Contains properties relevant for the constructive deployment of the component, thus for selection and building structure. Contains a structure classification system pursuant to EN 81346. Contains numerous properties in respect of physical dimensions and regarding start, processing and output values of the component. Contains a modular view of subcomponents or a device structure. Allows an automation view with inputs and outputs of different signal types.
Performance	Describes performance and behavioural characteristics in order to allow a summary assessment and Virtual Commissioning (V-IBN) of an overall system.
Functional	Makes statements on the function pursuant to EN 81346 and on the function of the subcomponents. Here location of the individual functions of the Technical Functionality also takes place, thus for example so-called “skills”, interpretation, commissioning, calculation or diagnosis functions of the component.
Local	Makes statements on positions and local relationships between the component or its parts or inputs and outputs.
Security	Can identify a property as security-relevant. This property should be taken into account for an assessment of security.
Network view	Makes statements in respect of electrical, fluidic, materials flow-related and logical cross-linking of the component.
Life cycle	Contains data on the current situation and historical utilisation in the life cycle of the component. Examples: allocation to production, maintenance protocols and past applications.
Human	In all views properties, data and functions should appear such that humans can understand individual elements, inter-relationships and causal chains.

Figure 16: Sample relationships between assets based on a PLC with a 3D printer



The relationship types are explained using the relationships of a PLC unit to a 3D printer (Figure 16). It should be noted that the image shown in the diagram is simplified, since the relationship assets are simple lines rather than assets.

Business relationship

From a business perspective, this relationship type between the PLC and a 3D printer is characterised in that the 3D printer supplies a service to the PLC and this service requires an order, incurs costs and involves delivery times etc. In this case, checks must also be carried out, for example, on whether the 3D printer delivers the agreed quality at the agreed price. In addition, the legal requirements associated with manufacturing the product up to and including export control must be taken into account. These are all matters that characterise the "business relationship" type of relationship.

Functional relationship

The functional relationship is a logical relationship between cooperating functions of the assets. In the case of the 3D printer, the function "Issue order" in the PLC triggers the function "Execute order" in the 3D printer. The information required at the start of the order is derived from the information relationship.

Informational relationship

The informational relationship covers all data and information that is required, generated or modified at the functional level. It is therefore used to execute functional cooperation between the assets, and also to correlate data from the assets established in a relationship e.g. by means of algorithms.

Communication relationship

Assets cooperating with each other must share information/data. A communication relationship provides everything necessary for exchange or distribution of the data required. A "communication channel" is set up to ensure secure data transfer using I4.0-compliant communication protocols.

Integration relationship

An integration relationship describes the basic connections such as power termination, mechanical construction and relevant material flows from the physical world between the assets, thus reflecting the physical world.

Mechanical construction includes connections such as a simple screw fastening or a pipe coupling, but also an asset placed on a table or, in a building, a column connected to the floor above or below.

Assets of a composite can form energy relationships based on basic energy types, e.g. as a relationship for

- Mechanical energy
- Electrical energy
- Hydraulic energy
- Pneumatic energy

Location relationship

Each asset is assigned to a particular location, which can also change in the case of mobile assets. In other words, this relationship can change over the lifetime of the asset. The location relationship describes the geographical information for two or more assets in a relationship.

Since the location itself is also an asset, any I4.0 component in a particular location has a relationship to a location I4.0 component.

Temporal relationship

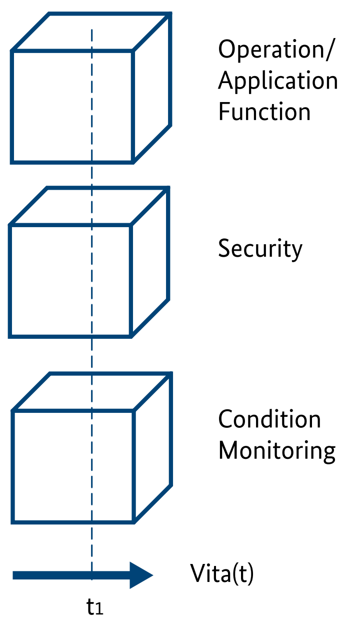
The temporal relationship describes the temporal information for two or more assets in a relationship.

Since time itself is also an asset, any I4.0 component at a particular time has a relationship to a temporal I4.0 component. The temporal asset comprises all relevant information for the particular time. This may be the time from the physical world, but may also be a virtual time, e.g. for simulation purposes. The related I4.0 component provides this information in the administration shell in I4.0-compliant format so that it is computer-processable.

Each description of a static arrangement only applies for the point in time of its description. Since each asset has its own lifetime, it is essential that changes in relationships be recorded in both the static and the functional (operational) part over time, provided these changes are relevant or necessary for executing common functions. The RAMI 4.0 Reference Architecture Model with its time axis is used for this purpose. By way of illustration, let us look at the three processes shown in Figure 17, since Processes also represent assets.

In a schematic representation using the typical RAMI 4.0 cubes, the diagram shows the asset "Operation", and below this, the assets "Security" and "Condition Monitoring". Like all assets, these assets have a lifetime. In other words, their state and location can in theory be determined at any time. This means that all assets can be simultaneously recorded at a particular time t1 together with their states and locations in the physical world. As a result, it is possible to correlate and evaluate the state of the overall system including the local deployment of individual assets in the information world. The snapshot thus created provides a specific view with a consistent time reference and thus establishes a temporal relationship between assets.

Figure 17: Temporal correlation between assets



State relationship

The state relationship describes all state-related information for two or more assets in a relationship.

Since the state itself is also an asset, any I4.0 component at a particular time has a relationship to a state I4.0 component. The state asset comprises all relevant information concerning the state of an asset. This may reflect the state from the physical world, but may also be a virtual state, e.g. for simulation purposes. The I4.0 component provides this information in the administration shell in I4.0-compliant format so that it is machine-processable.

Asset-Asset relationships

An asset-asset relationship represents the connection between two assets in the physical world. It is also an I4.0 component with an administration shell comprising the administration shells of other relationships.

Human-Asset relationship

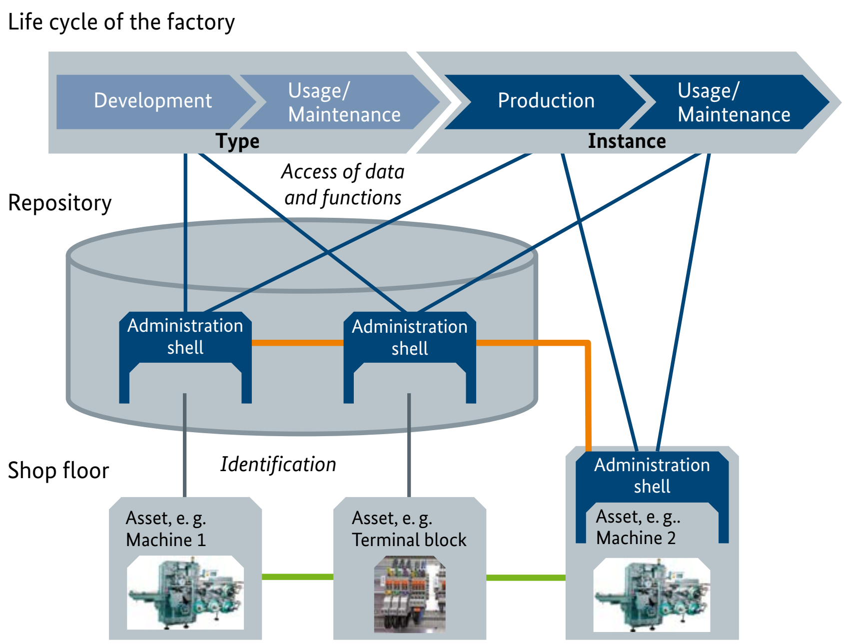
In the context of the model, a human is also an asset, but with options for intervention and control from outside the I4.0 system. The human-asset relationship allows humans to intervene in a I4.0 system from outside. Humans acting as part of an I4.0 system can only enter into the above relationships in the same way as all other assets in the I4.0 system.

3.4.7 Complex of relationships representing an I4.0 Assembly

Each connection between an asset and another asset in the physical world must be described in the information world using the above-described relationships. If corresponding criteria (DER) are provided for the type of relationship, these can be applied to check the connection.

Details of which asset has entered or wants to enter into a connection with which other asset are persisted in the administration shell of the relevant I4.0 component. Since the administration shells of an I4.0 component in principle can be deployed in different locations, and at the same time, always maintaining a unique reference to the asset, the total of all administration shells forms a large repository that may comprise fully distributed databases. Figure 18 shows this complex of relationships schematically.

Figure 18: Added together, all administration shells form an I4.0 Repository (modified from DIN SPEC 91345)



The big advantage of the administration shells that are identically structured according to RAMI 4.0 is that the previously separate consideration of the physical world in terms of assets and of the information world in terms of its digital representation of the I4.0 component is replaced by a homogeneous view. This is because the asset from the physical world must always be considered together with its representation in the information world, its administration shell (s); see also Figure 18, for example. Consequently, connections between the assets from the physical world can be reflected in the information world according to clear rules using relationships between the related administration shells. Thanks to the uniform structuring of the administration shells and their semantic contents, composites can be generated from I4.0 components. These composites are compatible with each other, both as assets in the physical world and as administration shells in the information world. This applies not only when an I4.0 composite component is generated, but also during its entire lifetime. Each change to an asset in the physical world results in corresponding changes in the administration shell and possibly also in the administration shell of the entire composite, which is also an I4.0 component with administration shell.

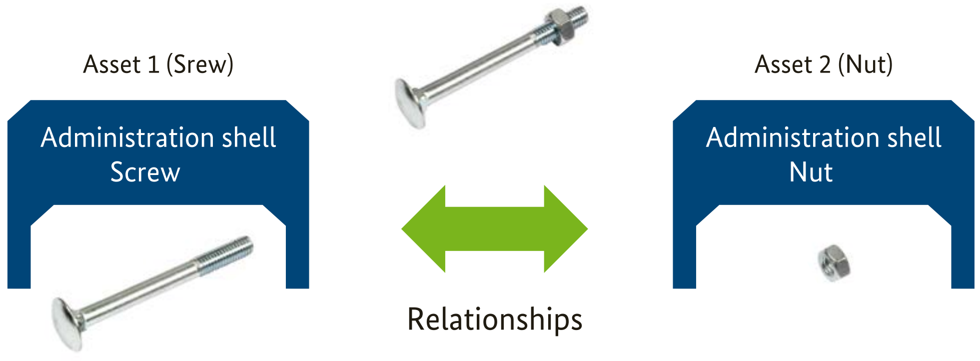
3.4.8 Samples for i4.0 Complex of relationships

Screw fastening

Let us now apply the model description used so far to a simple, concrete assembly for illustrative purposes. Figure 19 shows a screw fastening comprising the assets "Screw" and "Nut". This screw connection from the physical world generates a series of contents between both assets in the information world. First, we have the design aspect, which is described by the mechanical relationship in terms of the hole diameter, screw material, thread etc., on which the nut and its features must fit. Furthermore, the energy-related relationship must be considered: this is characterised by the forces acting on the screw fastening. These are both integration relationships. The functional relationship describes that the screw should be fastened or is fastened with the nut, e.g. with a torque appropriate for the application.

Even if this is not always the case, two parts can sometimes only be screwed together if certain rules or legal requirements are met. These rules can prevent a screw fastening from being completed if certain material incompatibilities or different reliability classes exist. This can then be expressed, for example, in the business relationship or in another relationship.

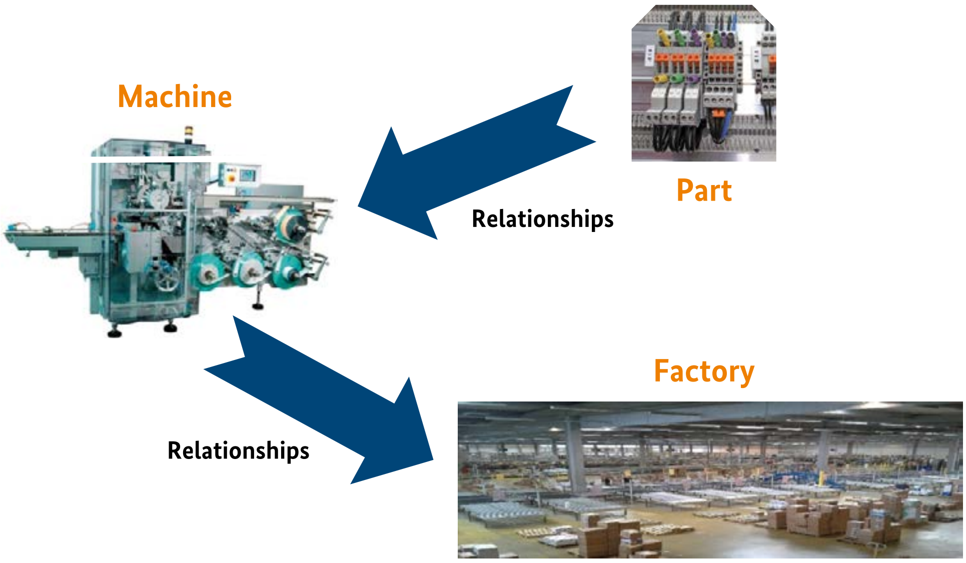
Figure 19: A screw fastening is described in the information world by relationships between the I4.0 component "Screw" and the I4.0 component "Nut"



Machine as part of a factory

Similar to this example, machines and entire systems can be described as shown schematically by Figure 20.

Figure 20: Each Industrie 4.0 arrangement consists of assets whose I4.0 components are in a relationship with each other



Pipe with valve

Any I4.0 assembly therefore always comprises I4.0 components that are interrelated to each other for a specific purpose. From the information perspective, the connections between the assets in the physical world are reflected by the corresponding relationships with information from the administration shells (in the information world).

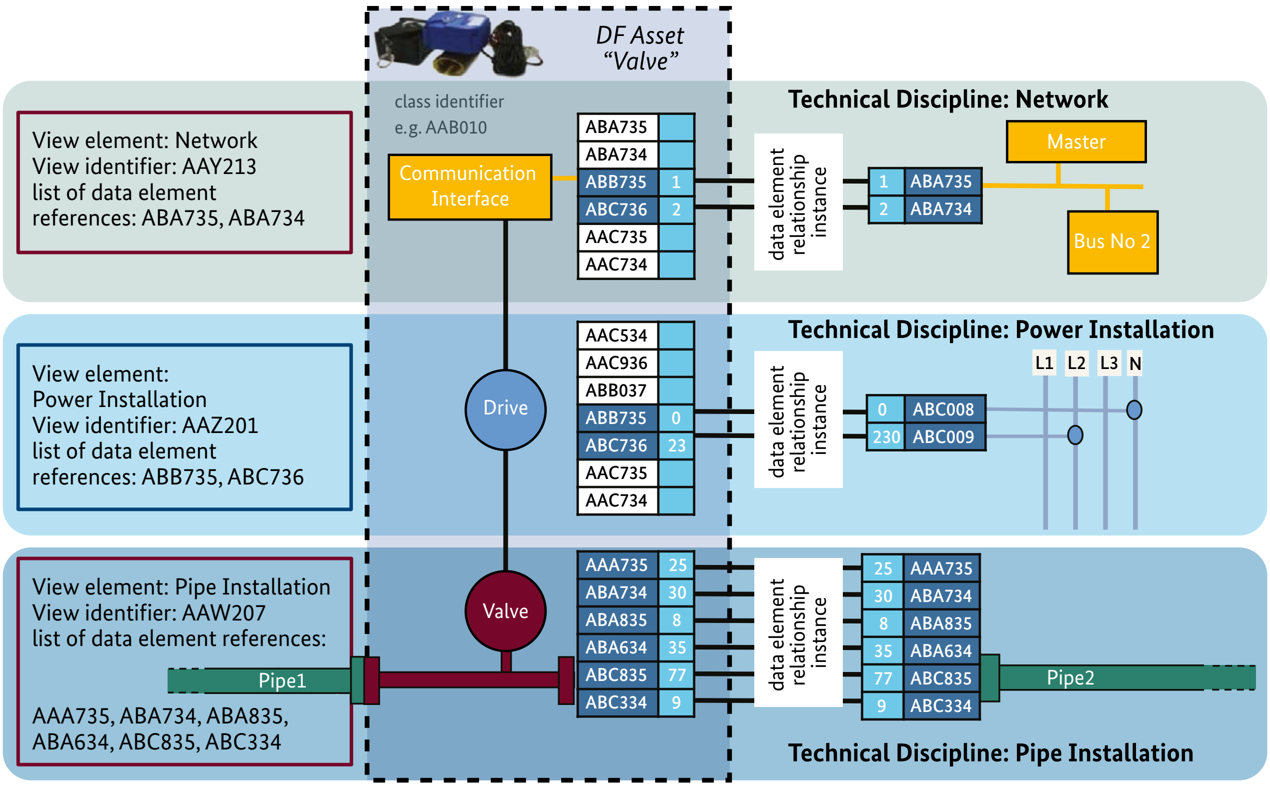
The following section considers a slightly more complex I4.0 arrangement. Figure 21 shows a design, based on different engineering documents, for a valve integrated in a pipe system. The engineering disciplines involved are:

- Mechanical construction
- Energy/Electrical engineering
- Communication engineering

“Pipe 1”, “Pipe 2”, and the valve itself, with its mechanical connectors, are relevant to mechanical construction. The electrical engineering provides the electrical energy supply. For establishing the valve’s cooperative capacity, it must be integrated in the communication network. All three technical disciplines nowadays work independently from each other, use different design tools with various data models and can establish consistency in the overall design only at great effort, often at the construction site.

If all three groups execute their design in accordance with the rules of Industrie 4.0, they use I4.0 components throughout the whole design. In this process, the necessary information is provided in compliance with Industrie 4.0 to the three technical disciplines by reading the contents (properties) from the administration shells. Since the administration shells all have the same structure, both the design within a technical discipline and the overall design are compliant to Industrie 4.0 and consistent. If the different groups require the same information, they access the same properties. Cross-discipline consistency tests thus can be carried out relatively easily. For example consider the test regarding the installation location of different components: This test easily can reveal dual occupancies of installation locations by comparing the installation coordinates of all assets across all disciplines.

Figure 21: Composite asset as a complex of relationships comprising I4.0 components [3]



4 Concept of the composite components

This section uses generally applicable points to describe a methodology for generating I4.0 components from connected I4.0 components. In the documents published to date [2][3], example-based observations have always been used to describe tangible production resources such as machines, servo axes and other products from component manufacturers etc. The relevant asset and related administration shells aim to provide a host of data and complex functions to the other Industrie 4 .0 participants. By their very nature, these I4.0 components are directed at the most diverse use possible.

However, a core purpose of production networks is to set up composites comprising a variety production resources and to direct these to a specific goal, for example, the manufacturing of a product. In this process, flexibility is tailored, requirements are set, parameters are selected and various considerations are combined into an overall design that meets the particular production goal.

The following section describes a concept of I4.0 components that are created by connecting I4.0 components. Composites of I4.0 components are referred to as 'composite components' in the following section.

Other parts of this document address the question of whether strategic planning, requirements specifications and production plans can be described by composite components.

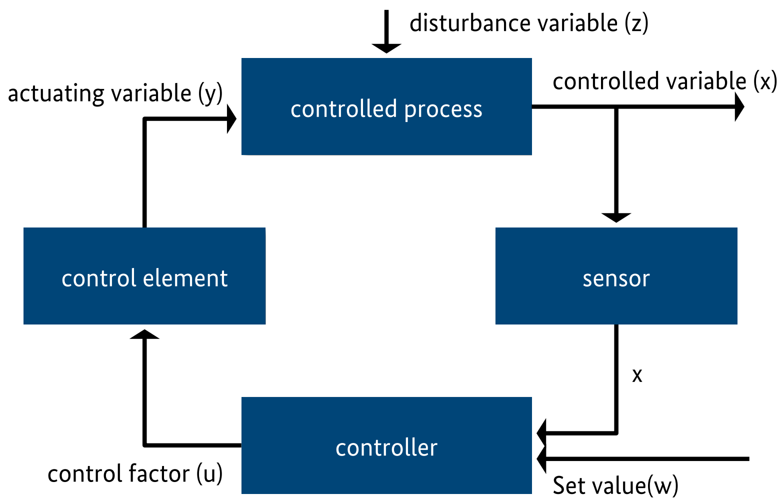
4.1 Composite components assign assets to a goal

Composite components are created when at least two I4.0 components are connected to each other. In this process, at least one property of one of the I4.0 components is set into relation with at least one matching property of the other I4.0 component. This type of composite component has new features in comparison to the individual I4.0 components. At the same time, a composite component is also an I4.0 component, i.e. the composite component also comprises an administration shell.

This can be illustrated by means of a control loop. Each of the four elements of the control loop in Figure 22 is an I4.0 component, comprising an asset and an administration shell. They each perform a specific function. When connected to each other, they establish links to form a closed loop. From an I4.0 perspective, this closed loop is the composite component "control loop". In this example, the link between the individual I.40 components established by connections between the assets only, but connections could also exist between the administration shells.

The I4.0 components displayed here in abstract form in fact are real objects (e.g. the control element is a valve, the controlled process is a pipe, the sensor is a flow meter and the controller is a control algorithm in a PLC). This example also shows the importance of using I4.0 components to rep-

Figure 22: I4.0 composite component "control loop", according to: http://www.chemgapedia.de, Grundlagen Regelung (basics of control technology)



resent elements that are 'passive' in the IT sense, such as the controlled process (a pipe). However, not all assets of the real world are represented by an I4.0 component. For instance, the fastening and connection elements in this example can be modelled as individually known or anonymous assets or even omitted.

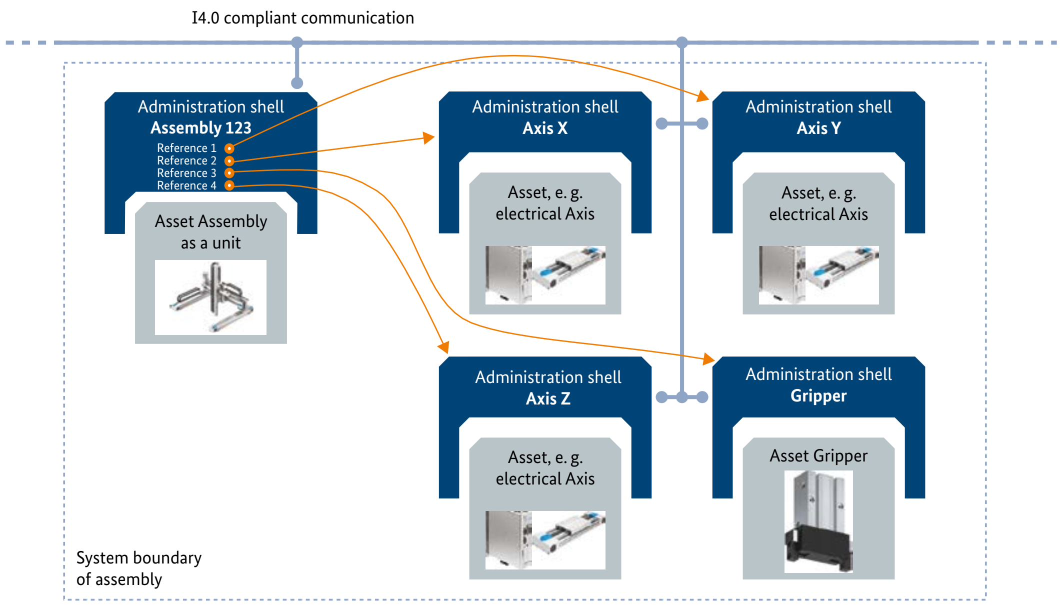
The advantage of representing the closed loop in a composite component is that the overall function of the closed loop can be managed. Its function can be observed and influenced as a whole. For example, new set values or other information can be specified for the whole closed loop (e.g. about the flow medium in question), which then are adapted to all parts of the closed loop (e.g. changed valve opening, adapted linearization for the flow sensor, adapted factors in closed loop), or an emergency signal can be sent to the closed loop to ensure that all parts behave accordingly (e.g. secure position for the valve, execution process for the control algorithm is stopped so that the PLC can use all resources for the emergency control).

This type of composite component for a closed loop can be defined as a one-off a kind (for a specific purpose) or as a template (type definition) for various applications.

In a composite component, the administration shell represents the data, functions and the status of the composite asset. The data and functions of the integrated individual I4.0 components can still be accessed via their corresponding administration shells. These integrated I4.0 components are referenced by the composite component, while the corresponding assets are called "self-managed assets". Individually known or anonymous assets in a composite of assets (i.e. assets without their own administration shell) can be represented by the composite component, in which case they are called "co-managed assets".

An assembly serves as an example in the following sections. To meet a particular production target, I4.0 components from different manufacturers are combined into a goal-directed unit which meets the specific requirements of the production target, for instance which is finely tuned to the particular product, production process and the surrounding machine/system. Very often, this integration of the different components is provided not by a component manufacturer but by a machine or equipment manufacturer. The machine manufacturer uses the flexibility of the components to create an assembly by engineering methods in the disciplines of material flows, mechanics, fluidics, electrical/electronic engineering and software programming.

Figure 23: Example of an assembly that connects numerous components to form a Pick&Place system



This type of assembly usually is created from one or more physical combinations of assets, which together form the system boundary of the assembly. The assembly as a whole thus again becomes an asset to which specific complex data and functions can be assigned and which is represented by the corresponding administration shell.

The complex information that can be assigned to the entire (goal-directed) assembly includes plans from the various engineering disciplines such as material flows, mechanics, fluidics, electrical/electronic engineering and software programming. The functions to be considered arise from the interaction of the different components. In the example presented here, such function could be the execution of a 3-axis pick&place operation for products with a specific geometry.

The purpose of introducing a composite component is to provide the important relationship information between the individual components in the form of I4.0-compliant information elements and thus enable this information to be used by other I4.0 components and higher-level systems. In this way important specifications, for example from the "engineering" phase, can be made available to an extended group of recipients. This is outlined in detail in the following sections.

4.2 Industrie 3.0 and Industrie 4.0

Several sections in this document are based on theoretical concepts that have already been implemented in Industrie 3.0, even if, in many cases, only as 'isolated applications' and in very heterogeneous realizations. However, these concepts are largely accepted in day-to-day production scenarios. In the main, the data and methods based on these concepts are already available and underpin approaches that can also handle non-I4.0 systems.

This is made clear in Section 4.4.2. Another defining characteristic of Industrie 4.0 is that the decisions made in this environment are far more dynamic, situation-dependent and decentralised. Therefore the concepts described here not only deal with data represented in an administration shell, but also draw on the idea that additional 'technical functionality' can be mapped into an administration shell.

4.3 Sub models for different engineering disciplines

This section looks at sub models that map information represented in the planning documents of the various engineering disciplines. It is assumed that the assembly (that is, the asset of the composite component) is adequately described by various engineering disciplines and their relevant documents. Not every engineering discipline takes account of the complete reality; the electrical diagram, for example, does not deal with the screw fastenings of the individual axes. A combination of several sub models therefore is required to produce a sufficiently accurate description of the overall reality of the assembly.

Requirement: The composite component shall support several sub models for various engineering disciplines and the information represented in the relevant documents.

Requirement: Each sub model shall describe the relationships for the relevant subset of the assets that are determined by the system boundary of the composite component.

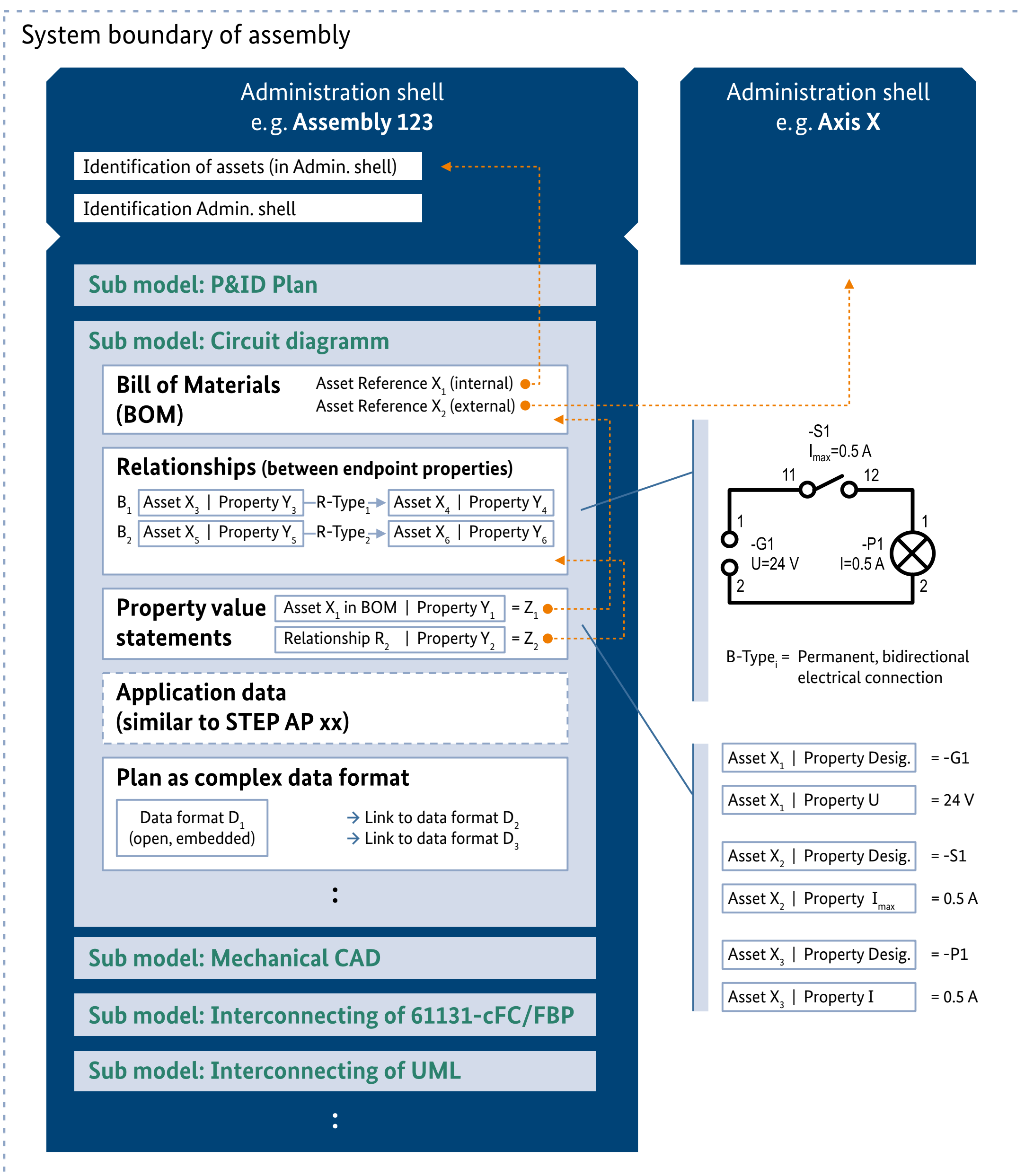
At the same time various engineering disciplines also will use their own organising principles for the assets within in the system boundary. For example the mechanical CAD will focus on the physical connections of assets, while an electrical diagram will be organised according to infrastructural or functional considerations. The principle of nestability, which applies to I4.0 components, can thus be geared towards several types of organising principles.

Requirement: Each sub model of the composite component for the individual engineering discipline shall have its own organising principles for the relevant assets. The hierarchy and organization thus achieved should then serve as an information source for the principle of nestability and allow access by other systems to the I4.0 components that are organised by the composite components.

4.4 Abstract concepts for sub models of composite components

The following section introduces several concepts for sub models of composite components, which are explained by means of examples. These sample concepts can be introduced and applied in an abstract way for several engineering disciplines. A later section then discusses the particularities of individual engineering disciplines.

Figure 24: Overview of abstract concepts for sub models of composite components



The sub models in question define relationships and specifications for individual assets. Several composite components can define different system boundaries and overlapping relationships and specifications. Therefore the following applies:

Requirement: The header of the administration shell of a composite component registers only the assets and their identifiers that are directly and unambiguously assigned to the composite component, not those that have their own administration shell (self-managed assets).

Individual, abstract concepts for sub models of engineering disciplines are now introduced in the following sections:

4.4.1 Bill of material (BOM)

Each of the engineering disciplines defines their own bill of material (BOM). For example, the BOM for an electrical diagram specifies electrical components that are to be connected (such as power supply units, terminals, fuses etc.) and the related infrastructural/connection elements (such as lines and cables). Only elements that are maintained in the BOM are assigned, for example, their own reference designations.

Since the definition of the sub models in question according to Section 4.1 includes other elements of an I4.0 composite, it makes sense to limit the BOMs of the individual sub models to those elements that are assigned as an asset in any administration shell.

Requirement: Each sub model of the composite component for the individual engineering discipline shall maintain its own "Bill of Material (BOM)" list of references to relevant assets. References should refer to co-managed and self-managed assets of the composite component.

Co-managed assets can be used to establish relationships between different sub models of a composite component.

Requirement: Each of the references shall also always allow an assignment to the particular asset, not only to the administration shell. They should therefore also be able to establish a link to the global identifier of the asset, not only that of the administration shell.

Infrastructural elements such as the node of a star connection or the assignment of a shield may be other elements that are not represented as co-managed or self-managed assets in their own right, but are nevertheless relevant for relationships and property value statements. For this reason, a third type of element is allowed in the BOM:

Requirement: Each sub model of the composite component shall also be able to list elements in its "Bill of Material (BOM)" that are only referenced by relationships and property value statements and that do not represent co-managed or self-managed assets.

4.4.2 Relationships between assets

A key element introduced by various engineering disciplines and their planning documents are the relationships between assets (introduced in Section 3). A representation of diverse relational structures between assets that are described by unified semantics and are recognised by many I4.0 systems should support, among others, at least the following use cases:

Table 3: Various use cases in Industrie 4.0 and the meaning of their relationships

Use case	Sample motivation for relationships between assets
Automation of automation	If drive axes are to be able to form axes groups automatically, and if functions are to be automatically distributed across the most appropriate control systems or those provided and control panel displays automatically generated, it is necessary for the engineering information of an assembly or plant to be known to many systems.
Diagnosis	An automatic diagnosis can only be carried out across all assets if the relationships between individual assets and their subunits are known.
Operation procedures	With complex production lines and ever more rapidly changing products, it is crucial for operators to be supported by (multimedia) instructions. These instructions must be displayed automatically and across all assets.
Augmented reality (AR)	For AR, it must be possible to determine the location of each asset from the corresponding productive plant construction.
Commissioning	To support commissioning, it makes sense, for example, if the functionality of a servo amplifier can automatically determine which motor, actual value transmitter and mechanical construction the drive chain is connected to.
Production execution	The product, the production process and the production resources must be interrelated with each other for production to be executed. From an I4.0 view, the priority is to design this relational structure so that the best production approach, appropriate optimisation goal, optimum production process and the most suitable machine and plant are dynamically selected and interconnected with data links.
Maintenance/ fault diagnosis	To support fault diagnosis, it can be very helpful to be able to display the reference designations, the location or even the appropriate circuit diagram on the most suitable device for the maintenance staff.
Optimisation/“big data”	For optimisations and “big data” scenarios, it is important to know which motor is connected to which gripper and by which mechanical axis. Only in this way can complex statistical and semantic analyses be carried out for large construction projects within a reasonable timeframe.

The need to represent a variety of diverse relational structures can also be motivated by the application scenarios of Plattform Industrie 4.0 [6]:

Table 4: Application scenarios in Industrie 4.0 and the meaning of their relationships

Application scenario	Sample motivation for relationships between assets
Order-controlled production (OCP)	Digital description of equipment and process modules that allows these to be combined far more flexibly than today for automated order planning and order placement while enabling use of their specific capabilities.
Adaptable Factories (AF)	Interoperable modularity at all factory levels. Digital description of production means and product variants. Support for the system integrator when setting up and modifying production lines.
Self-organising Adaptive Logistics (SAL)	Description of the location, position and relationships of production lines and intralogistics in relation to each other. Greater integration of logistics into production lines as a result of standardised visualisations and tools.
Value Based Services (VBS)	The product providers supply data as a raw material and procure new services on the basis of this data. The service suppliers must also know about the interrelationship between the production means and sources of this data.
Operator Support in Production (OSP)	Digital models of subassemblies, functions and production means are required to describe faults, visualise interconnections and support multimedia tools and augmented reality. Consistent access to this data must be ensured.
Smart Product Development for Smart Production (SP2)	For product development in value networks, products, equipment and production means must be described by means of established engineering disciplines. A standardised infrastructure must be available for this purpose, indicating how these models are accessed and how the relationships between the models are managed.
Innovative Product Development (IPD)	A digital model should also incorporate market and customer requirements and relate these to various subassemblies and function groups. This supports the system-of-systems approach and the participation of different value adding partners in a development process and subsequent optimisation.
Dynamic Consistent Engineering of Applications (DCA)	For consistent engineering, each partner in a value network must have access to an interoperable model of the plant and the products. Cross-domain relationships and basic model information must be provided for this purpose, in addition to detailed information from the individual engineering disciplines.
Circular Economy (CRE)	To maximise remanufacturing and reuse of production resources, resources must be described at each level in terms of their abilities and their structure. It is important that this information be stored across the entire life cycle and remain accessible even in the event of changes in areas of responsibility.

According to Section 4.3, the relational structure between assets is described very heterogeneously by a variety of engineering disciplines and planning documents. To set up this relational structure so that it has unified semantics and can be processed by numerous different I4.0 systems, relationships should be stored in accordance with 3.4.3. It must be possible to generate this information using the software tools used in the various engineering disciplines. This requirement does not apply to any relationship, but only to those that are provided as endpoints, as described in Section 3.4.6.

Requirement: The relevant sub model of the composite component for the individual engineering discipline shall be able to manage a countable and hierarchical list of property-classified relationships between two and more assets and their specific endpoint properties.

Requirement: To be I4.0-compliant, a software tool used in an engineering discipline to structure/design the asset of the composite component must support the generation of the set of relationships in the corresponding sub model.

However, a sufficiently complete representation of the overall reality is usually only achieved by drawing together several different engineering disciplines:

Requirement: It should be possible to integrate the sets of relationships of the individual sub models of the composite components to form a comprehensive set of relationships.

4.4.3 Property classification of relationships

Classification of the represented relationships must be general enough to ensure comprehensive analyses and processing of the information in many I4.0 systems.

At the same time, the represented relationships must be sufficiently precise and meaningful to be recognised by the representatives of the individual engineering disciplines and taken into account by the methodologies. Possible examples include:

Table 5: Property classifications for different engineering disciplines

Eng. discipline	Examples of property-classified relationships
Mechanical construction	Permanent joint, prismatic joint, hinged joint, cylindrical joint, ball-and-socket joint, each incorporating friction, springs or damping
Electrical engineering	Single-pole electrical connection, multi-pole electrical connection, data bus connection, connection of potentials, connection between terminal modules
Fluidics	Tube or pipe connection, direct screw or plug connection, terminal connector between components, open space connection (sound absorber)
P&ID planning	Pipe connection, sensor and actuator connection
Chain of function blocks in automation technology (61131-CFC/ FBD)	Signal connection, event connection, bus connection
Interconnection of software components (UML Communication Diagram)	Lifetime (vita) in communication connection Note: Individual messages can then be modelled as properties of this relationship.
Factory floor layout	Conveying line (continuous), conveying line (discrete), electrical infrastructure, pneumatic infrastructure, media infrastructure, intralogistics, pedestrian traffic route, vehicle traffic route

For the above-named relationships, it must be assumed that each of the relationship types will need to be extended; an extension to the classification system must be provided. This can be carried out by representing the relationship classes in an IEC 61360 Repository such as eCl@ss.

Requirement: property classification of relationships must be designed to allow the addition of further relationships to the classification system.sschemas um weitere Beziehungen möglich ist.

4.4.4 Property value statements

From the example for the electronics plan, it is also clear that composite components can require or specify properties for (self-managed) assets. Such required/specified property can refer, for example, to a reference designation, an operating mode, a gear factor, a set value or similar. The relevant asset must be able to correspond to this requirement or specification with the properties in one of its sub models.

Requirement: The relevant sub model of the composite component for the individual engineering discipline must be able to manage a countable and hierarchical list of property value statements which are directed at co-managed and self-managed assets on the 'Bill of Material (BOM)' and the properties present in their administration shells.

According to Section 4.1, composite components follow a goal. This also means that the composite components make different property value statements for assets referenced by them. This type of mechanism can also be used to describe alternative planning scenarios by selecting the described assets and their property value statements using different composite components, whereby for example two composite components plan the same assets. Changes in the assignment of assets to composite components can continue to represent changes over the life cycle if, for example, components are reused in different plant parts.

This suggests that many property value statements that could be persisted in an administration shell of a particular asset can be better specified by property value statements through the adequate composite component.

Corresponding determinations may include:

Table 6: Possible property value statements

Reference designations	The plans from various engineering disciplines frequently assign their own reference designations. In the case of an electronics plan, this could be a reference designation according to DIN EN 81346. Different composite components can provide various alternatives for constructing a production station, for example.
Location details	The location details of an asset can be defined on different plans, for example according to a geodetic position, on a location plan, factory layout plan or production line plan. Different composite components could, for example, reflect the history of plant composites or specify a hypothetical plan, for running a discrete event simulation (DES).
Operating mode, set values	A composite component or a plan could provide different operating modes or set values for various products or production scenarios.
ALOP/DLOP/OLOP according to IEC 61987-10	Industrie 4.0 is intended to enable planning processes. Composite components can represent different planning documents and hypotheses, which could, for example, include requirements lists for planned but not yet fully defined assets. A frequently used example could be that of a "role for a pump with a nominal output of 500 l/min".

In Section 4.4.3 a relationship classification is assigned to each of the relationships between assets; no other specifications are made in respect of the relationships. The property value statements option should therefore be extended to allow setting of signal names or similar properties, for example.

Requirement: The relevant sub model of the composite component for the individual engineering discipline must be able to manage a countable and hierarchical list of property value statements which more precisely describe the defined relationships in the sub model.

4.4.5 Application data

In addition to defining individual, independent properties, as described in Section 4.4.4, it can be useful to describe a set of coherent properties and data that adequately supports the well-defined use cases in sufficient detail. This is achieved, for example, with the STEP Application Protocols (AP), such as ISO 10303 AP214 for product data exchange in automotive manufacturing.

Therefore a composite component can also include sub models that describe coordinated property structures for particular use cases. Depending on the use case, these sub models can either be assigned to an engineering discipline (e.g. ISO 10303 AP242 for mechanical 3D construction) or also maintained generally in the administration shell of the composite components. If the property structures are directed at precisely one engineering discipline (such as in the AP242 example), it should also be possible to maintain these structures in the same sub model:

Requirement: The relevant sub model of the composite component for the individual engineering discipline must be able to include a countable and hierarchical list of lower-level sub models that support individual use cases, e.g. according to the template of the STEP application protocols.

Alternatively these coordinated property structures can of course also be maintained as independent sub models.

4.4.6 Storing detailed plans using complex data formats

The descriptive means described in the above sections (relationships, property value statements and application data) cannot provide a holistic view of all information contained in conventional plans from the various engineering disciplines. This is also not necessary with respect to the use cases described in Section 4.4.2.

However, this information should be made available to various users, such as engineering systems or maintenance staff, by means of an I4.0 infrastructure. It should therefore be possible to store, use and maintain complex data formats such as 3D data formats or circuit diagrams in the composite component for use by the individual engineering disciplines.

Ideally, in an Industrie 4.0 context it should be possible to specify a standardised open data format for each planning document needed in each engineering discipline. This data format should adequately meet the extensive requirements in terms of description depth and integration with engineering systems and also ensure the necessary efficiency for the engineering processes that are managed, for example, by company-specific Product Lifecycle Systems (PLM).

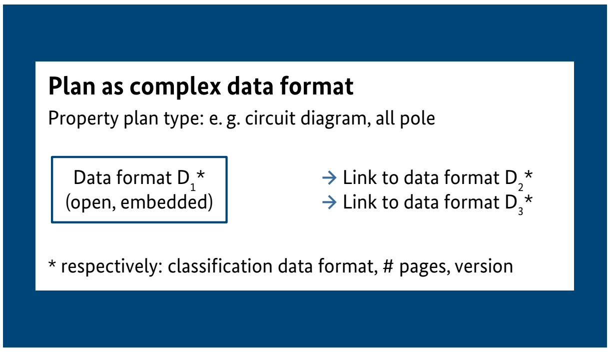
However, according to the current feedback from companies[2], it appears unlikely that even by combining forces, Industrie 4.0 would be able to name open data formats that could completely replace the existing data formats and eliminate the ‘double bookkeeping’ associated with PLM.

Since different data formats are currently used in the various engineering disciplines, there are no signs of a standardised data format being used in the foreseeable future. However, the shared use of properties in this document will lead to an interlinking of data models, which will ultimately ensure the cooperative capacity of the I4.0 components across engineering disciplines. The approach in this document foresees, that a detailed, sufficiently open data format can be stored in such a sub model. Additional,

2 Internal surveys, but also online surveys: http://www.marktstudien.org/marktstudien/marktstudie-engineering-prozess/

detailed data formats or the relevant working revisions in company-specific PLMs should be referenced by internet-compliant links.

Figure 25: Details from above diagram



In sum, Industrie 4.0 will continue to support the establishment of open, interoperable data formats. In the interests of interoperability and the resulting data volume, non-open, non-interoperable data formats cannot be directly embedded in the administration shell. The criteria used to select such a data format are as follows:

1. Open data format; can be implemented in many software systems.
2. Future widespread use of this format does not endow any one manufacturer with a particular advantage.
3. Reading/displaying the represented data is simply organised through software libraries.
4. Existing engineering systems already allow the data to be displayed, imported, or exported to an adequate extent.

We can thus state the following:

Requirement: The relevant sub model of the composite component for the individual engineering discipline should allow complex data formats to be embedded for individual planning documents or to be referenced.

Requirement: In general, only open data formats that are recognised by the Industrie 4.0 reference architecture can be considered as data formats to be embedded for an individual sub model of the composite component. For company-internal use, a non-open data format can also be embedded, provided that an open data format is also embedded. Other data formats can be referenced by a URI[3].

Requirement: For each embedded or referenced complex data format in a sub model of the composite component, the following should be provided: a classifying property for the planning document of the relevant engineering discipline, a classifying property for the relevant data format, a property for the represented pages and a property for a version status.

Requirement: It must be possible to maintain additional, non-standardised properties for each referenced complex data format in a sub model of the composite component. A link to company-specific Product Lifecycle Management (PLM) Systems is permitted.

4.5 Automation ML

"AutomationML (Automation Markup Language) is a neutral data format based on XML for the storage and exchange of plant engineering information, which is provided as open standard. The goal of AutomationML is to interconnect the heterogeneous tool landscape of modern engineering tools in their different disciplines, e.g. mechanical plant engineering, electrical design, HMI development, PLC, robot control."[4]

3 Uniform Resource Identifier; see https://de.wikipedia.org/wiki/Uniform_Resource_Identifier

4 Citation; see Wikipedia, https://en.wikipedia.org/wiki/AutomationML

AutomationML has the potential to become a key representation data format for engineering and structural knowledge in the future. However, at the time of writing this paper, AutomationML does not yet have its own complete semantics. It can be assumed that properties from eCl@ss will be used for the semantics.

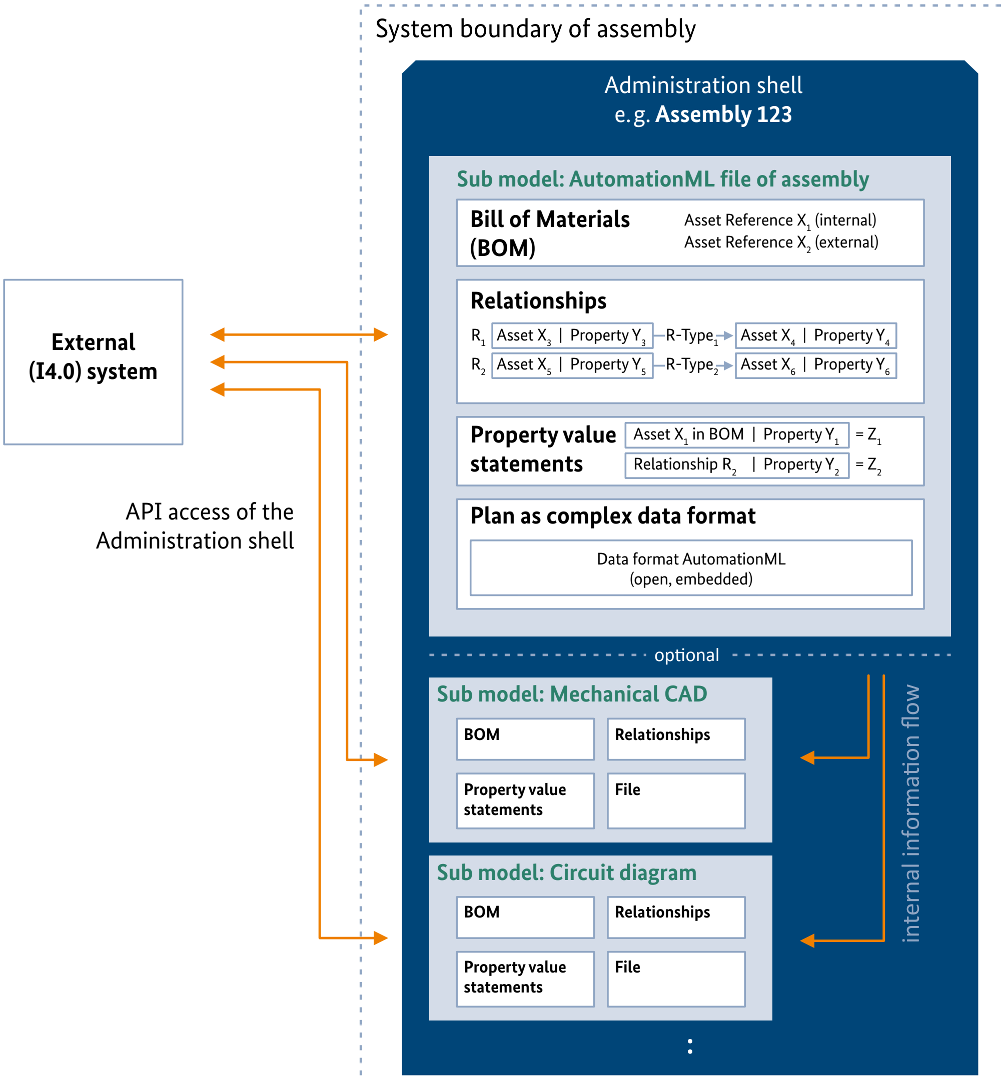
AutomationML is therefore considered to be another document that

(a) enables direct embedding of AutomationML files and

(b) foresees further development of AutomationML beyond the specifications for composite components.

The aims of embedding are to provide information on: which assets are captured by an AutomationML representation, extract a useful minimum of relationships and property value statements to the semantics of Industrie 4.0, ability to store and to maintain the file itself in the administration shell and in I4.0 systems.

Figure 26: Overview of the embedding of AutomationML as a document in composite components



We can thus state the following:

Requirement: A composite component should allow AutomationML data as a sub model for a planning document that defines the assets and the relationships between these in a composite. AutomationML is an open and complex data format and thus suitable for embedding (→ 4.4.6).

To maintain logically consistent administration, the scope of the AutomationML file must be aligned with the scope of the composite component asset, or a composite component must be defined in accordance with the scope of the AutomationML file:

Requirement: If AutomationML is used as a document for a composite component, the scope of the representation in AutomationML must correspond to the scope of the represented system boundary of the composite component.

AutomationML could potentially define very extensive semantic specifications about the assets in question. At least the information explained as relevant for Industrie 4.0 in this paper (→ 4.4.1, 4.4.2, 4.4.3, 4.4.4) should be provided as a source of information in the composite:

Requirement: If AutomationML is used as a document for a composite component, assets that are represented by AutomationML and that correspond to other elements of an I4.0 composite should be referenced as co-managed and self-managed assets of the administration shell. Property-classified relationships and property value statements should be extracted in line with other sub models for planning documents in compliance with I4.0.

Established use of AutomationML makes it likely that information about mechanical, electrical and fluidic construction will be represented using complex data formats as described in Section 4.3. This is currently enabled, for example, by embedding COLLADA or JT data in AutomationML. The following should apply to avoid an accessing I4.0 system having to implement several access paths to this data:

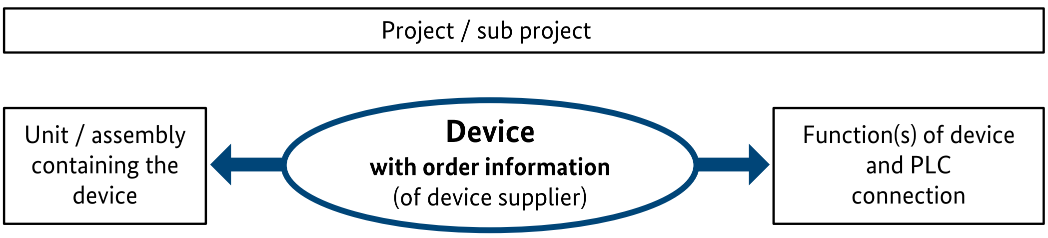
Requirement: If AutomationML is used as a document for a composite component and contains suitable complex data contents that can be used as a sub model of the composite of the assets for an engineering discipline, it must be possible for these data contents to be extracted as a sub model for the corresponding engineering discipline.

4.6 VDMA 66415

VDMA 66415, a specification drafted by the German Engineering Federation, describes "a universal data base to provide a shared information overview and data management for all jobs involved in order-related engineering processes (trades, companies, individuals) in mechanical and plant engineering"[5]. Data content and organisation correspond to the user view in engineering (construction, design) and are based on the physical or functional structure of a machine or plant (project/order).

It should include user-defined data ('project-defining data') that describe a composite of automation devices and the structural and functional units containing these:

Figure 27: The device as reference point for VDMA 66415



5 VDMA 66415 draft, not yet published

According to VDMA 66415, 'project-defining data'

- is data defined by the user during the construction and design process,
- are of interest for at least two of the jobs involved in engineering (typical example: reference designations and functional text of a device);
- relate to electromechanical, fluid power, electronic or programmable electronic devices, as contained for example in common parts lists (sensor/actuator lists, motor and component lists) and/or I/O lists."

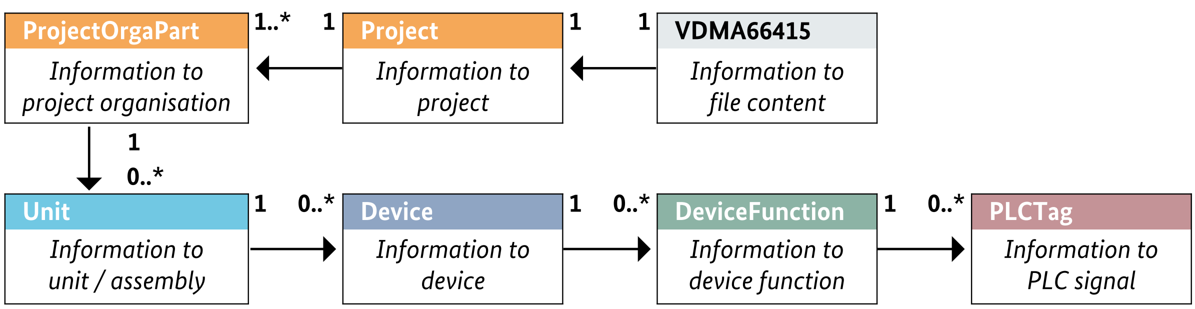
"With the same data content and data organisation, the universal data base can be represented in different data formats or file formats and aimed at the different user groups:"

Table 7: VDMA 66415: Table overview of VDMA 66415 file formats

File format	File structure	Target group, us
Table format	Table	Users in mechanical and plant engineering, standardised devices/components/IO lists in engineering, construction, design, manual writing and reading of project data
XML format	Project tree	Software development manufacturers or users, simple automated read/write support for project files
XML format	AutomationML	Manufacturers of software-based engineering tools, automated read/write support for project files

VDMA 66415 includes, inter alia, the names and semantically unambiguous identification of relevant data and properties and the structures resulting from these:

Figure 28: VDMA 66415: Overview of organisation of data structure in project tree



For these reasons (suitable subject matter of the composites, focus on semantically unambiguous data, format independence), the VDMA 66415 data base is highly suited to being mapped as a sub model for composite components.

The following thus applies, in the same way as for AutomationML:

Requirement: A composite component should allow VDMA 66415 data as a sub model for a planning document that defines the assets and the relationships between these in a composite. VDMA 66415 is an open and complex data format and thus suitable for embedding (→ 4.4.6).

In this case, it is vital that the described scopes match the system boundary of the composite component:

Requirement: If VDMA 66415 is used as a document for a composite component, the scope of the representation in VDMA 66415 must correspond to the scope of the represented system boundary of the composite component.

The following also applies for the minimum usable information for Industrie 4.0:

Requirement: If VDMA 66415 is used as a document for a composite component, assets that are represented within VDMA 66415 and that correspond to other elements of an I4.0 composite should be referenced as co-managed and self-managed assets of the administration shell. Property-classified relationships and property value statements should be extracted in line with other sub models for planning documents in compliance with I4.0.

In addition to providing a representation as an embedded complex data format (as a table, XML, AutomationML), you can also treat the entire data format here as a hierarchical property structure with varying countability:

Requirement: For the purposes of mapping VDMA 66415 in administration shells of composite components, a sub model should be defined and standardised. This sub model should clearly depict the information content from VDMA 66415 as a property structure, as already described[6].

4.7 Specifications for individual engineering disciplines

The following section outlines some possible best practices for individual engineering disciplines. A table is provided below for this purpose, along with a list of individual discussion points. The information used in this section describes a current situation and should not be considered as normative references.

6 Plattform Industrie 4.0, 2016, Output Paper 'Structure of the Administration Shell', Chapter 3.2ff

Table 8: Possible key points for the specifications of individual sub models for engineering disciplines

Eng. discipline	Possible planning documents	Possible elements of the BOM	Sample property value statements	Open, embedded data format	Possible referenced data formats	VDMA 66415
Mechanical construction	3D model, 2D drawings, sectional views	Geometries of assets, other geometry objects, geometries and positions of mounting points and connection elements, lines, cables, tubes/ pipes	Location details	STEP[1], IGES, COLLADA, JT	Dassault Solid-Works/CATIA, Autodesk Auto-CAD/ Inventor, PTC ProEn-gineer/ Creo, Siemens NX	✔
Electrical engineering	Circuit diagram, connecting diagram, cable plan, arrangement diagram	Electrical equipment and elements, cables	Reference desig-nations, Location details, operating mode, set values, ALOP etc.	DXF[2], PDF[3], SVG[4]	EPLAN P8, ZUKEN E3.series	✔
Fluidics	Circuit diagram, arrangement diagram	Fluidic equipment and elements, hoses/ pipes	Reference desig-nations, Location details, operating mode, set values, ALOP etc.	DXF, PDF, SVG	Vendor-specific data formats	✔
P&ID planning	Flow chart		Reference desig-nations, Location details, operating mode, set values, ALOP etc.	DXF, PDF, SVG	Vendor-specific data formats	✔
Chaining of function blocks in automation technology	61131-CFC, 61131-FBD	Software function blocks according to IEC 61131/61499, inputs/outputs, events	–	PLCopen XML[5]	Vendor-specific data formats (Siemens, Beck-hoff, 3S, etc.)	✔[6]
Interconnec-tion of soft-ware compo-nents	UML Com-munication Diagram	Software function blocks in any high-level language	Communication attributes	XMI[7]	–	
Kinematic validation of construction	3D model with kinematic components	Assets that show supporting elements and kinematic ele-ments. Assets that show the geometries and dynamics of products.	Location details, torques, travel distances, weights	COLLADA, JT	Import data for virtual commis-sioning	

→

Eng. discipline	Possible planning documents	Possible elements of the BOM	Sample property value statements	Open, embedded data format	Possible referenced data formats	VDMA 66415
Virtual commissioning	3D model, software projects	Geometries, dynamics, kinematics. Representations of automation technology devices. Projects that connect these to applications.	Location details, torques, travel distances, weights, device responses to process and non-cyclical data.	AVANTI[8]	WinMOD, SIMIT, etc.	
Factory planning	Location plan, factory floor layout	Equipment, machines, plants, infrastructural elements, conveying lines, traffic routes	Reference designations, location details, capacities, dimensions/ cross-sections	DXF, PDF, SVG	–	
Plant planning				No need for a complex data format	–	
Production planning	Work plan	Products, semi-finished goods, materials, tools, equipment, machines, plants	ALOP, quantities, quality levels, times, availability	CSV[8], No need for a complex data format	–	

The individual discussion points are as follows

1 STEP, since it may be possible to adopt additional STEP definitions. Alternatively, the technically more advanced IGES format could be used.
A check should be carried out to verify whether COLLADA or JT should be set as a data format in recognition of AutomationML.

2 DXF is an open format which should not endow any particular manufacturer, including Autodesk, with any decisive advantage.

3 PDF is a highly complex graphics format. The display also requires a complex software component; assigning information to displayed image sections can become complicated (use case 'Maintenance/fault diagnosis', → 4.4.2)

4 SVG has the distinct disadvantage that standardised support is not available for several pages.

5 PLCopen XML can be used as an interchange format for FBD. CFC is not a 61131 standard language and is not represented in PLCopen XML.

6 VDMA66415 considers the assignment of PLC variables, no concatenation of functions

7 XMI is an open, vendor-neutral format standardised by the Object Management Group (OMG) which allows the exchange of all UML models.

8 See for example: http://avanti-project.de/index.html

5 Smart products and order-controlled production

According to Section 4.4.2, executing production is an important motivation for connecting assets using relationships. The following sections expand on the concepts relating to this.

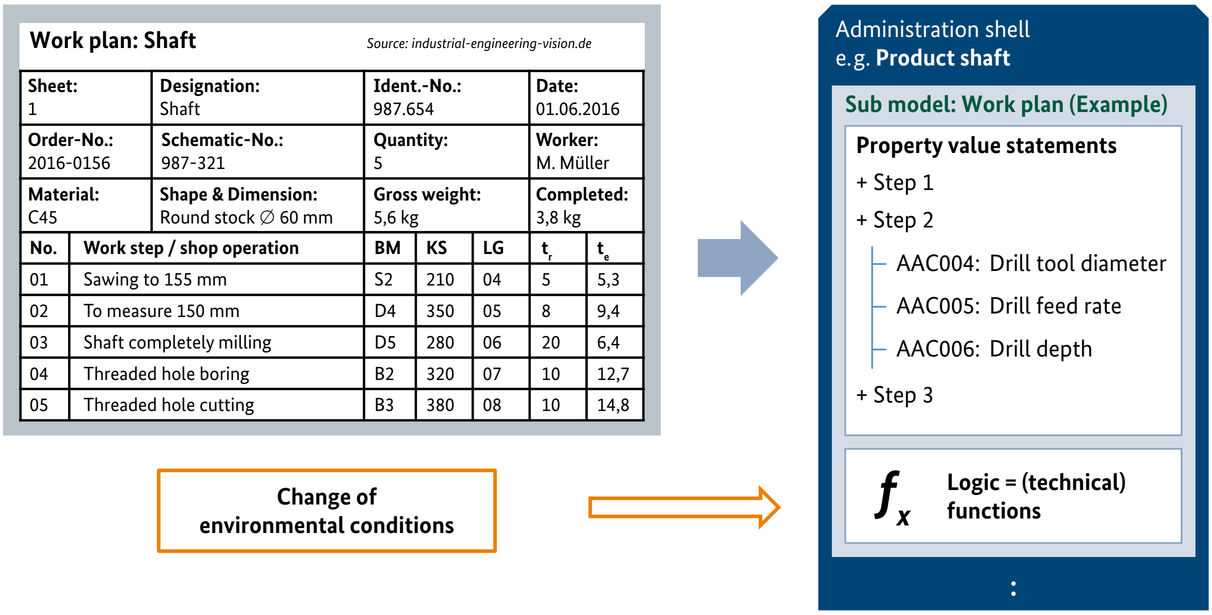
For production execution, another group of assets enters the stage: those of smart products. These are converted by administration shells into I4.0 components. For production execution, novel and dynamic shaping of the relational structure between the product assets and the assets of the machines and plants is required, as described in 'Structure of the Administration Shell' [2] and Chapter 4. For example, the best production approach, appropriate optimisation goal, optimum production process and most suitable machine and plant should be selected in each case and interconnected with data links. This results in order-controlled production that can dynamically respond to changes in environmental conditions and can provide an order-related interface for horizontal integration in value networks.

Similar to the evolution from Industrie 3.0 to Industrie 4.0 and Chapter 4, this process involves building on an established concept, carrying out standardisation in terms of descriptions and providing dynamic scope for design. In the case of order-controlled production, the concept is that of work plans.[7]

5.1 Scenarios for mapping work plans on machines and plants

In the following discussion, it is assumed that the data from a conventional work plan are translated (transformed) into property value statements in a sub model (→ 4.4.4). It is therefore assumed that the work to be performed for a production order is converted into a sequence of work steps with property value statements for the production processes to be carried out:

Figure 29: Transforming a work plan from Industrie 3.0 into property value statements for Industrie 4.0; partial source (German only): https://industrial-engineering-vision.de/arbeitsvorbereitung/arbeitsplan/



Work plan: Shaft *Source: industrial-engineering-vision.de*

Sheet: 1	**Designation:** Shaft	**Ident.-No.:** 987.654	**Date:** 01.06.2016
Order-No.: 2016-0156	**Schematic-No.:** 987-321	**Quantity:** 5	**Worker:** M. Müller
Material: C45	**Shape & Dimension:** Round stock ∅ 60 mm	**Gross weight:** 5,6 kg	**Completed:** 3,8 kg

No.	Work step / shop operation	BM	KS	LG	t_r	t_e
01	Sawing to 155 mm	S2	210	04	5	5,3
02	To measure 150 mm	D4	350	05	8	9,4
03	Shaft completely milling	D5	280	06	20	6,4
04	Threaded hole boring	B2	320	07	10	12,7
05	Threaded hole cutting	B3	380	08	10	14,8

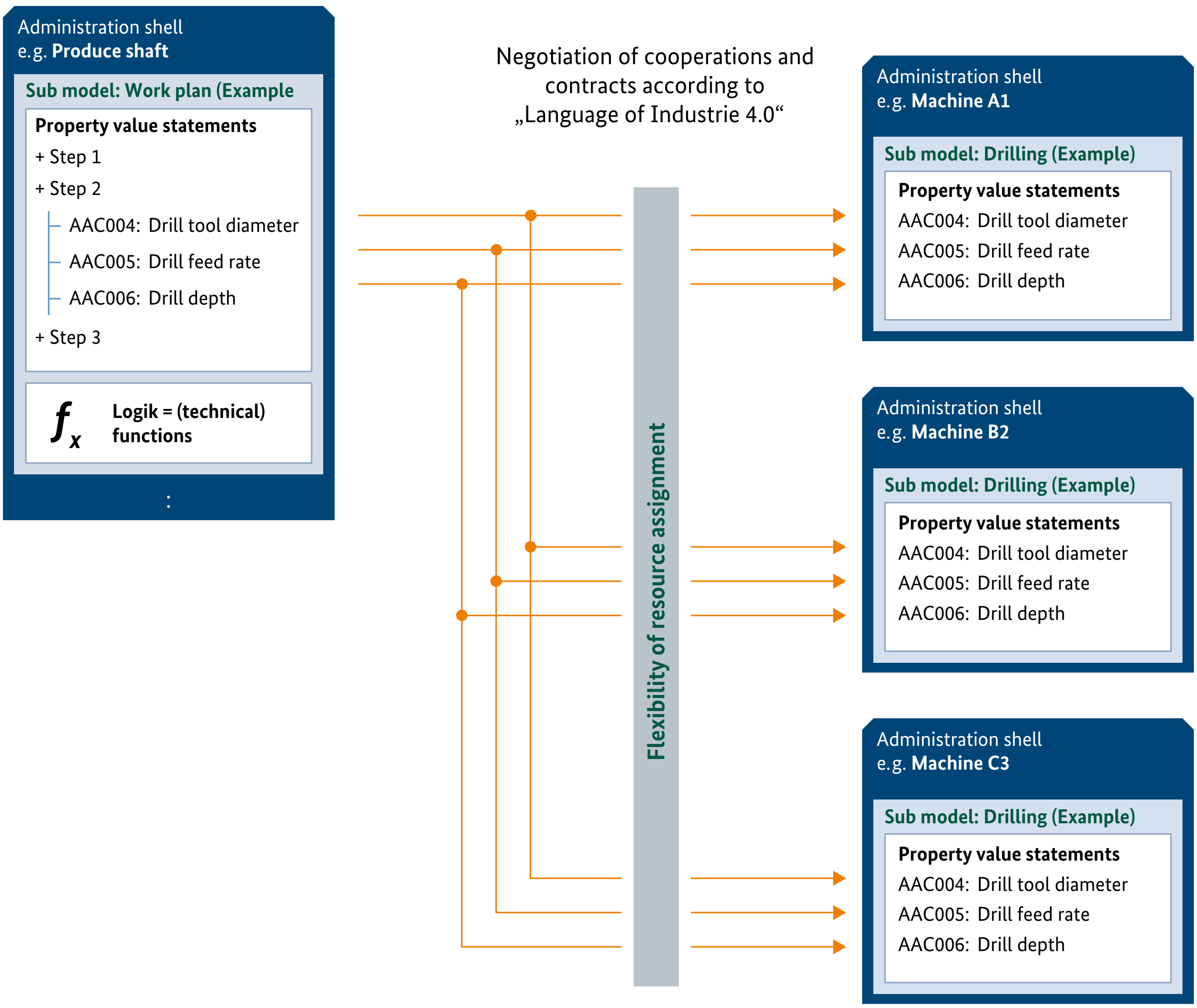
7 See for example (German only) https://de.wikipedia.org/wiki/Arbeitsplan or also https://industrial-engineering-vision.de/arbeitsvorbereitung/arbeitsplan/

Even at this early stage, the technical functionality of a sub model and the administration shell can be used to respond intelligently to external environmental conditions and, for example, to adapt the sequence of production steps or production parameters in line with orders.

5.2 Flexibility in assigning resources as a starting point

The mapping of sharedly used properties to each other in this area is essential. It results in flexible mappings, hat generate immediate benefits such as flexibility in assigning resources. Production resources can include machines, plants, infrastructure and logistics elements. This type of flexibility can be used, for example, to distribute production orders flexibly between a group of similar machines (product actively searches for a machine that executes the next processing step) or, in the event of a machine outage, reassigns production orders to similar machines.

Figure 30: Resource flexibility through use of common properties



In the case of the example displayed in the diagram, production step 2 can be executed on various machines (A1, B2 or C3). The ability to drill is represented in the same way on all three machines. Basic compatibility is established if each of the abilities correspond in terms of their essential properties. Based on the negotiation of cooperation types and thus on the "Language of Industrie 4.0" (→ Section 5.7), a concrete resource assignment can be established. This assignment can be flexible and dynamic.

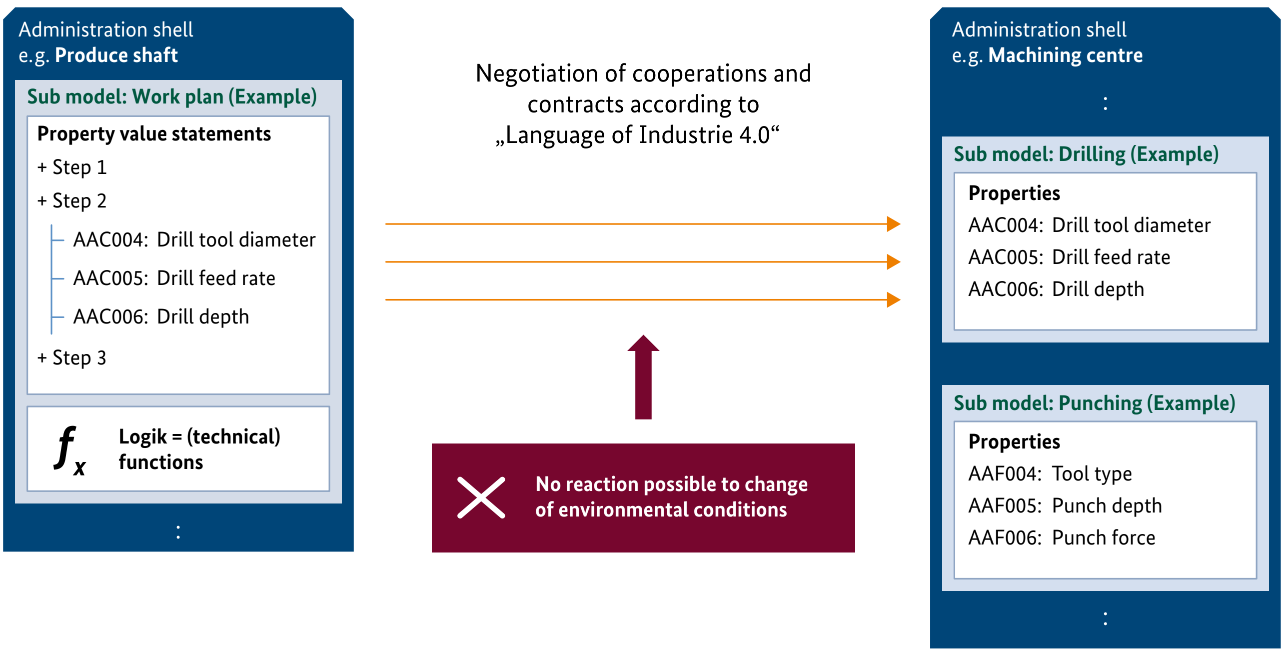
This also means that interaction patterns can use the same property as a requirement (on the left-hand side) or also as an assurance (on the right-hand side)[8]. In this process, negotiability of the interaction patterns ensures that the best decision is made at any time.

However, if the abilities of the various machines do not correspond, more extensive and novel scenarios must be considered. Therefore several scenarios should be used to interrelate a work plan to be implemented for production execution with the assets of machines and plants (production resources). Each of these scenarios is described in the following sections, along with advantages and disadvantages.

5.3 Direct mapping on a production process

If the property value statements of a step in the work plan can be directly mapped onto the property structures of a production process, direct mapping can be carried out as follows:

Figure 31: Direct mapping of work plan on production process



8 See also Chapter 5.3.2 in: Discussion paper "Interaction Model for Industrie 4.0 Components", Plattform Industrie 4.0, November 2016

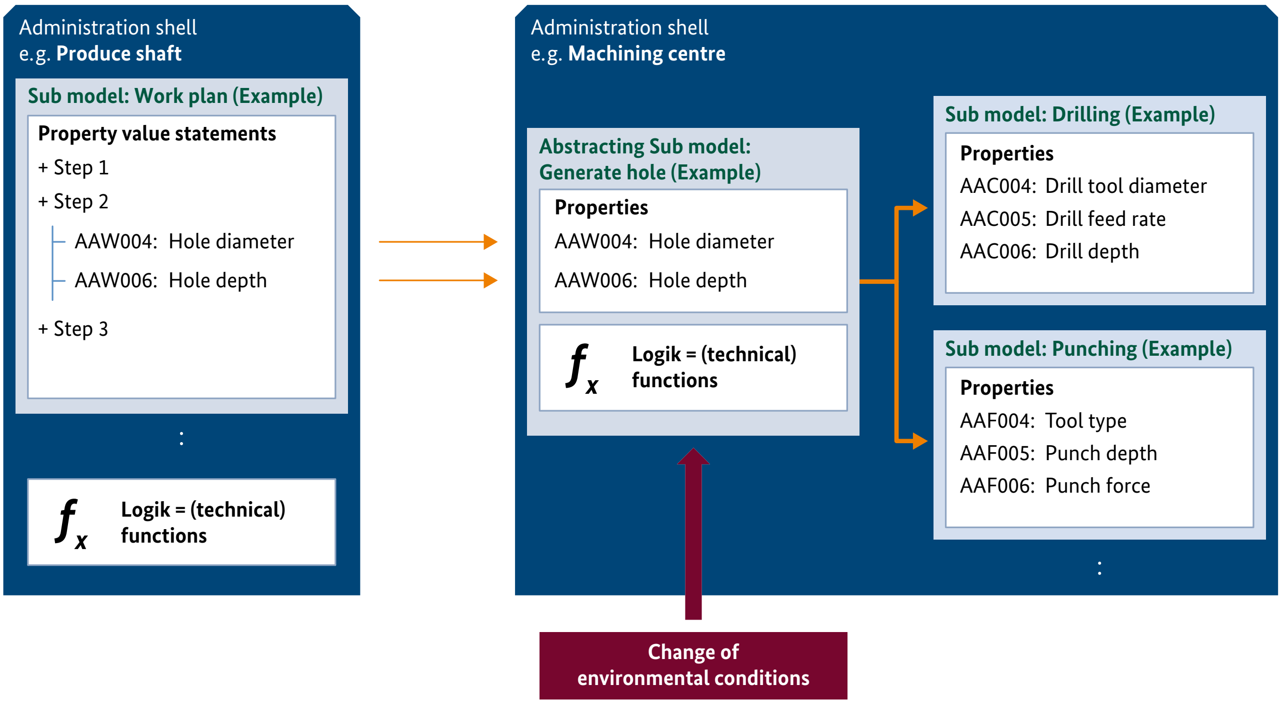
Advantages	• Simple conversion of existing specifications and recipes from Industrie 3.0 ('brown field', 'legacy') is possible.
Disadvantages	• The property value statements in the work plan must correspond to those of the production process in the machine/plant. There is no flexibility with regard to selecting the production process. • Environmental conditions cannot be taken into account in mapping on the production process. No changes can be made, for example, to the setting of optimisation targets. • Adjustments to the particular features of an individual machine/plant ('recipe optimisation') can only be carried out with explicit property value statements. • Much of the knowledge about the individual production processes is coded in work plans; switching production processes is laborious and expensive.

5.4 Translation by administration shell of the resource

The administration shell of a production resource, for example of a machine or plant, is not restricted to offering only one sub model of a production process. A processing centre could thus enable both the 'Drilling' and 'Punching' (as in nibbling) processes. This in turn means that **additional sub model**s can be set up which **generalise the specific production processes**. In this way, you can create an entire hierarchy of sub models that offer different and alternative abilities.

Translation by means of abstracting sub models is carried out here in the context of the administration shell of the production resource. The starting point is what the production resource provider knows about possible production processes that can be supplied by the individual asset.

Figure 32: Translation carried out by an abstracting sub model of the production resource



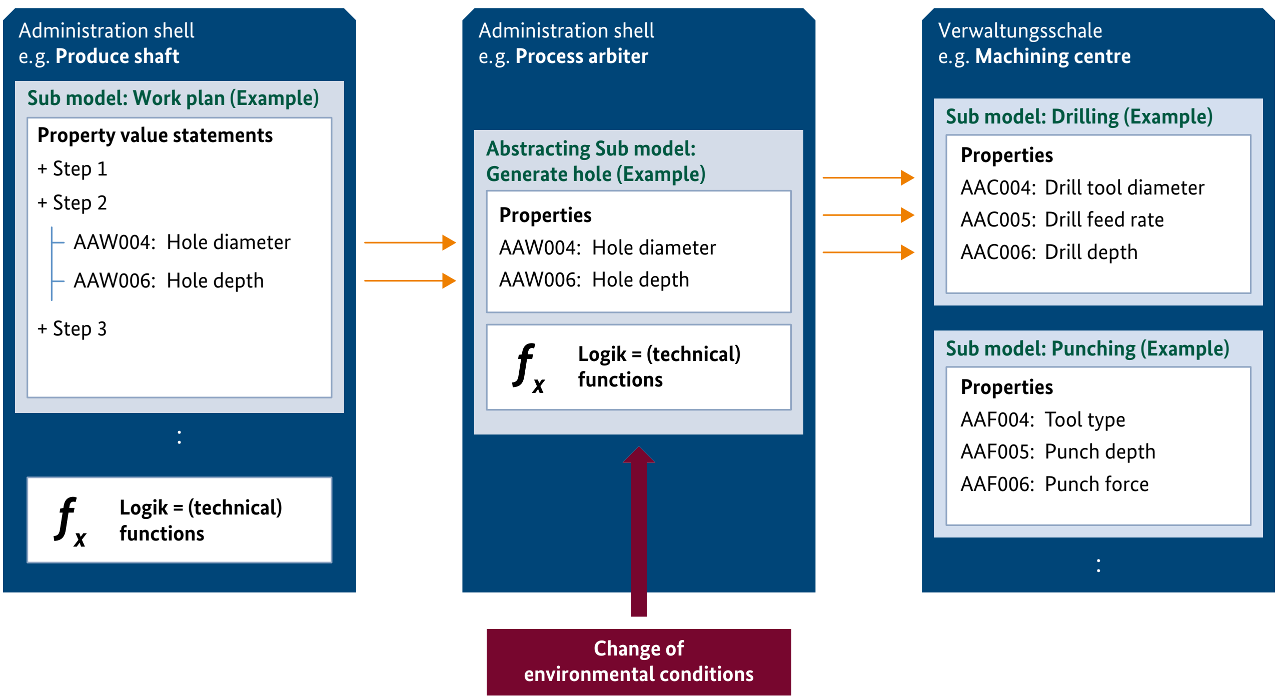
The following evaluation can be made:

Advantages	• Retention of simple scenario involving only two assets. • The translation can respond to changes in environmental conditions. • The supplier of the administration shell has extensive domain knowledge about the machine/plant and is thus in a position to adapt the translation logic optimally to these. • The supplier of the production resource can offer and receive payment for added value. • Any conflicts between resources within the machine/plant can be resolved optimally.
Disadvantages	• The changes in environmental conditions must be communicated with a consistent description to all production resources. • It is difficult to optimise the translation to meet product requirements.

5.5 Translation by arbiter

The translation process can also be carried out by a special arbiter. This arbiter can combine knowledge of various production processes and a specific array of various products. In this way, the arbiter can provide abstracted production ability in the I4.0 network and thus coordinate production orders. This mechanism mainly refers to the mapping of property value statements onto a concrete sub model for a production process. Subsequent negotiating of collaborations to be entered and the activation of production contracts (orders) can take place separately, directly between the product asset and the asset of the production resource. Since properties are described in an information format that can be read both by humans and machines, humans can also intervene to make corrections. Human interaction can thus be incorporated in manual, semi-automatic or automatic processes.

Figure 33: Translation process carried out by a special arbiter



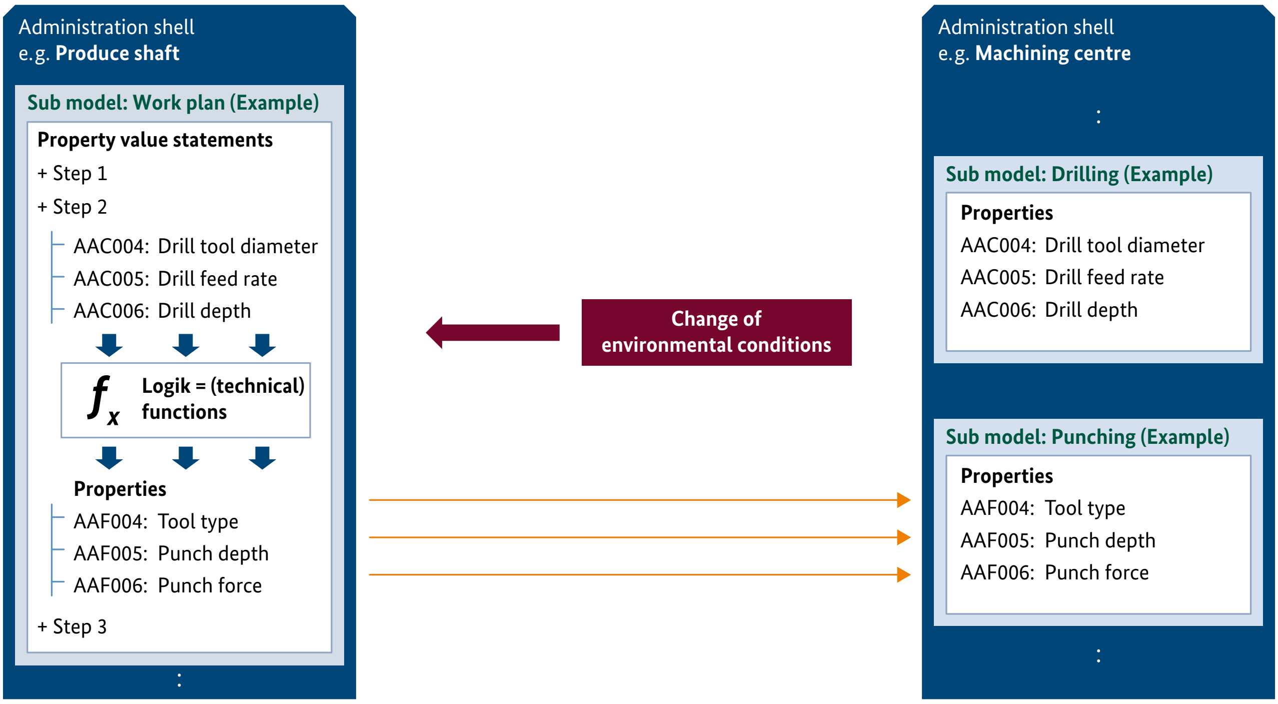
The following evaluation can be made:

Advantages	• Thanks to specialisation, the arbiter can implement optimisation with respect to products and production processes. • This specialisation can allow, for example, highly qualified arbiters to be implemented for certain industrial sectors. • The translation can respond to changes in environmental conditions. Depending on the number of arbiters, the changes in environmental conditions only have to be transmitted to a few recipients. • A new partner can position himself in the value network as a supplier of the arbiter assets and obtain market share for added value.
Disadvantages	• The arbiter represents a single point of failure and thus poses a threat to overall system availability. • The structure of three participating assets is highly complex and requires the 'Language of Industrie 4.0' as a communication mechanism. • It is difficult to engage with the specifics of the production resources,e. g. machines and plants.

5.6 Product executes translation for multiple production processes

The smart product can also independently carry out a translation with regard to several possible alternative production processes and the corresponding property value statements. The translation is then carried out, depending on the particular situation, by the technical functionality within the administration shell of the product. However, this means that the smart product must possess knowledge about itself as well as about all possible production processes. Regarding the required quantities of information to be respected, this translation needs to be implemented for every product instance. If facilitated by a product type, the arbiter scenario would apply, as a different asset administration shell would exist (→ 5.5).

Figure 34: Translation carried out by the product itself



The following evaluation can be made:

Advantages	• Simple conversion of existing specifications and recipes from Industrie 3.0 ('brown field', 'legacy') is possible.
Disadvantages	• A very large amount of knowledge and functionality must be mapped to a very high number of product instances. • The changes in environmental conditions must be communicated with a consistent description to all product instances. • A translation with regard to a larger number of eligible production processes does not scale well. • It is difficult to optimise the translation in relation to production processes and individual machines/plants. • Much of the knowledge about the individual production processes is coded in work plans; switching production processes is laborious and expensive.

5.7 Importance of 'language of Industrie 4.0'

The 'Language of Industrie 4.0' plays an especially important role in these scenarios. The alternatives described in the previous sections differ mainly in terms of the location where knowledge and technical functionality are stored. The interaction patterns of 'Language of Industrie 4.0' [7] [8] are based on this domain knowledge and enable more extensive optimisation and adaptation processes. In this regard, abstracting sub models provide a better starting point for the interaction patterns as well as allowing for more technical considerations.

After or during implementation of the interaction patterns, the decentralised technical functionality can provide other services.

5.8 Requirements associated with the different scenarios

The aspects discussed in Sections 5.3 to 5.6 highlight that each outlined scenario has advantages and disadvantages. Different scenarios may be utilized in a factory, depending on if the complexities are more on products or on production resources. Different partners in the value network want to offer added value. Therefore, from a pre-competitive view, no single preferred scenario can be specified as an optimum. The following applies to all scenarios:

<u>From Section 5.1:</u>

Requirement: The property value statements in sub models of the composite component shall allow a representation of manufacturing processes and their possible sequences, references to possible production resources, subproducts, materials and tools as well as any value statements and limiting requirements that are directed at the sub models of other assets.

To support dynamic scheduling of suitable production resources for a manufacturing process, it must be possible to set up not just 1:1 specifications of assets, but also to specify different equivalence classes of assets for selection; that is, specify a set of assets from which an asset can be dynamically selected. This type of specification can help the 'Language of Industrie 4.0' to determine the most suitable asset.

Requirement: The references to possible production resources in sub models of the composite component shall provide an expandable classification of different equivalence groups and their supporting parameters.

Traceability and quality documentation are vital in Industrie 4.0.

Requirement: For traceability of production processes ('Tracking & Tracing') and quality documentation, it must be possible to document the following in a suitable sub model of the composite component: sequence of production steps, selection of references to assets and other resources, specifications and requirements that are directed at sub models of other assets.

<u>From Section 5.3:</u>

A sub model of an asset can thus refer to many other different assets in its property value statements and relationship definitions. For efficient operation, the following should therefore apply:

Requirement: The service-oriented retrieval interfaces of the administration shell[9] shall allow to efficiently identify, which assets and properties referenced by the sub models of the administration shell are assigned.

<u>From Section 5.4:</u>

The ability to build on existing sub models and either generalise or abstract their functions and the information they contain is an important concept for providing domain knowledge such as production abilities and the definition of added value offerings.

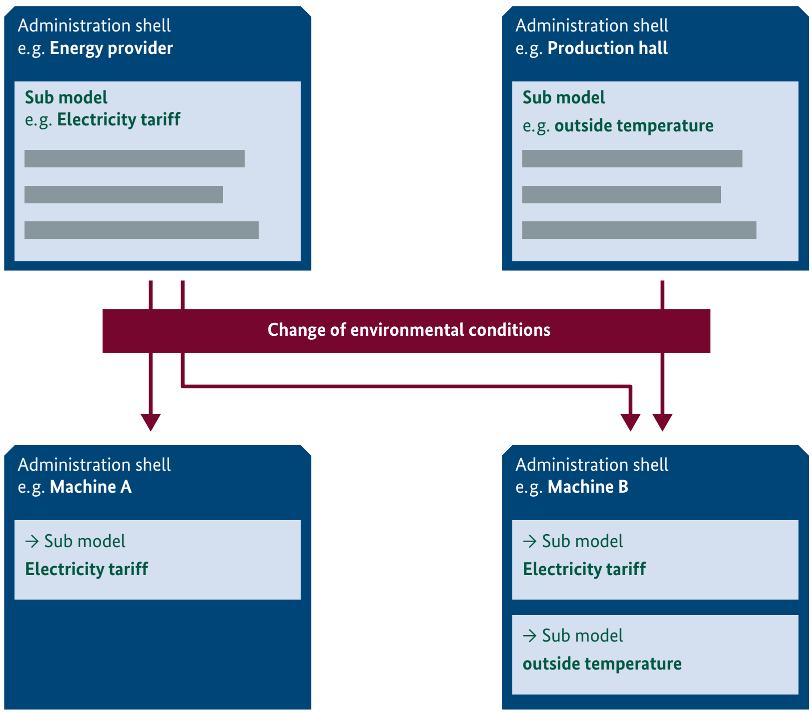
Requirement: The structure of the administration shell (of all assets) shall allow the creation of sub models that generalise about other sub models of the particular administration shell. This ability should be accessible to the technical functionality of the abstracting sub model.

---

9 API; see 'Structure of the Administration Shell' document

For this approach to work, it must be possible to communicate a change in different environmental conditions to the administration shell. In the following statements, it is assumed that these environmental conditions are managed as a sub model of the 'interested' administration shell in each case. In this way, it is possible for an asset or the corresponding administration shell to express which environmental conditions are relevant for the change to the internal behaviour. Various groupings (that is, sub models) of environmental conditions are thus enabled.

Bild 35: A change in environmental conditions is communicated to different administration shells



Environmental conditions must be communicated transparently.

Requirements: The ‘Language of Industrie 4.0’ must allow sub models of an administration shell to be updated in line with changes to another administration shell with an identical sub model.

From Section 5.5:

The scenario described in this section places additional demands on the ‘Language of Industrie 4.0’:

Requirement: The ‘Language of Industrie 4.0’ must be able to handle the necessary negotiating of collaborations and contracts within the changing framework of three (product, arbiter, resource) or two partners (product, resource).

The described scenario relating to possible single point of failures highlights a need:

Requirement: The ‘Language of Industrie 4.0’ should allow redundancy of an I4.0 component to be defined in order to ensure system stability and/or load distribution. For this purpose, restrictions such as statelessness can be taken into account.

From Section 5.6:

No further requirements can be identified from this section.

Future expandability of sub models:

The requirements outlined in the previous section for sub models for smart products and order-controlled production are far from adequate for all future scenarios. Expandable sub models are intended to prevent sub models from competing with or excluding each other.

Requirement: The structure of the administration shell shall be designed such that sub models can be expanded by inheritance relationships in accordance with the principles of object-oriented programming.

Bibliography

[1] acatech, "Forschungsfragen in „Produktionsautomatisierung der Zukunft", acatech 04/2012"

[2] Plattform Industrie 4.0, 2016, „Struktur der Verwaltungsschale – Fortentwicklung des Referenzmodells für die Industrie 4.0-Komponente" (EN: "Structure of the Administration Shell – Continuation of the Development of the Reference Model for the Industrie 4.0 Component")

[3] ZVEI, 2016, „Beispiele zur Verwaltungsschale der Industrie 4.0-Komponente – Basisteil" (EN: "Examples of the Asset Administration Shell for Industrie 4.0 Components – Basic Part")

[3] IEC FDIS 62832

[4] Plattform Industrie 4.0, 2014, 'Umsetzungsstrategie Industrie 4.0' der Plattform (EN: 'Implementation Strategy Industrie 4.0' Report on the results of the Industrie 4.0 Platform)

[5] Beuth, 2016, DIN SPEC 91345:2016-04, Reference Architecture Model Industrie 4.0 (RAMI4.0)

[6] Plattform Industrie 4.0, 2016, „Fortschreibung der Anwendungsszenarien" (EN: "Aspects of the Research Roadmap in Application Scenarios")

[7] Plattform Industrie 4.0, 2016, "Interaction Model for Industrie 4.0 Components"

[8] Plattform Industrie 4.0, 2016, „Weiterentwicklung des Interaktionsmodells für Industrie 4.0-Komponenten"

AUTHORS:
Dr Heinz Bedenbender | Alexander Bentkus | Prof. Dr Ulrich Epple | Dr Thomas Hadlich | Martin Hankel | Roland Heidel | Oliver Hillermeier | Dr Michael Hoffmeister | Haimo Huhle | Markus Kiele-Dunsche | Dr Heiko Koziolek | Dr Steffen Lohmann | Marco Mendes | Dr Jörg Neidig | Florian Palm | Stefan Pollmeier | Benedikt Rauscher | Frank Schewe | Bernd Waser | Ingo Weber | Prof. Dr Martin Wollschlaeger

This working paper has been elaborated in the working group "Models and Standards" of the ZVEI in cooperation with the Working Group "Reference Architectures, Standards and Norms" (Plattform Industrie 4.0).