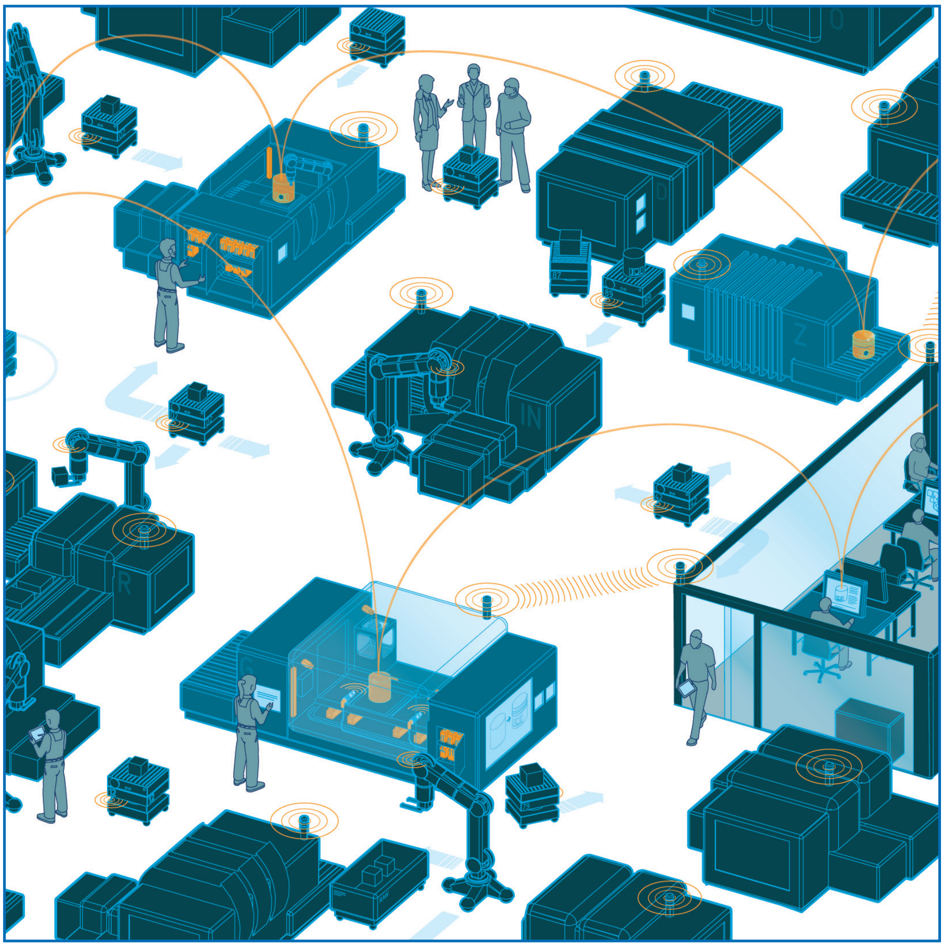
ZVEI: Die Elektroindustrie

White Paper

Examples of the Asset Administration Shell for Industrie 4.0 Components – Basic Part

Continuing Development of the Reference Model for Industrie 4.0 Components



April 2017

German Electrical and Electronic Manufacturers' Association

Examples for the Asset Administration Shell for Industrie 4.0 Components – Basic Part

ZVEI - Zentralverband Elektrotechnik- und Elektronikindustrie e.V.
German Electrical and Electronic Manufacturers' Association
Führungskreis Industrie 4.0, SG Modelle & Standards
Lyoner Strasse 9
60528 Frankfurt am Main, Germany

Contact:
Gunther Koschnick
Phone: + 49 69 6302-318
E-mail: koschnick@zvei.org
www.zvei.org

April 2017

Content

1. Objective and Methodology

This document, "Examples of the asset administration shell for Industrie 4.0 components", was created in March 2016 and developed further by the Spiegelgremium (SG) Modelle & Standards (Models & Standards mirror committee) of the Industrie 4.0 management team (FK I4.0) at ZVEI. Based on a decision in October 2016, the initial status of this document is being published as the basic part.

The aim of the publication is to provide examples relating to the recently agreed "structure of the asset administration shell" and thus to strengthen a common understanding of the content. This applies in particular to the collaboration with VDI/VDE GMA FA 7.21 and the Ontology sub-working group of Plattform Industrie 4.0.

The intention of this document is to strengthen understanding of the asset administration shell contents using illustrative examples. This document does not aim to provide a specification. As the structures of the asset administration shell and the specifications for implementation will be developed on an ongoing basis, for example through the openAAS project, this document will also be modified and supplemented.

The contents from content matter domains presented here are also intended as illustrative examples. They do not in any way reflect the content of a submodel and ignore the current standardisation efforts for the sake of better comprehensibility.

The information in this document is intended for both the factory automation and process automation industries. Terms such as factory, manufacturing and shop floor also include facilities in the process industry.

For better readability, Industrie 4.0 is abbreviated to I4.0 in compound terms. Unlike in previous publications, the term "asset" is used here instead of "thing" to correspond with DIN SPEC 91345.

2. Relevant Existing Content

This section highlights existing content from previous discussions or other working groups, thus emphasising the interconnectedness with other topics.

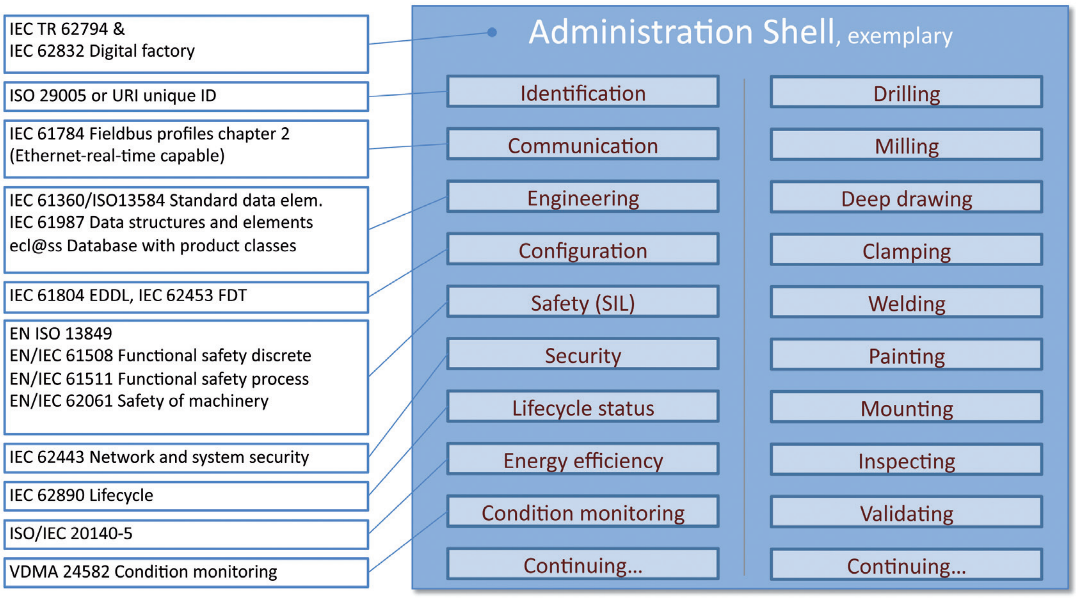
2.1 Idea behind the submodels

The basic idea of I4.0 components is to surround every Industrie 4.0 asset with an asset administration shell that can provide a minimal but sufficient description according to the Industrie 4.0 use cases. At the same time, it is important that we are able to map existing standards in accordance with the definition of the asset administration shell in question.

The asset administration shell is thus made up of a series of submodels. These represent different aspects of the asset concerned; for example, they may contain a description relating to safety or security, but could also outline various process capabilities such as drilling or installation.

The aim is for only one submodel to be standardised for each aspect. For example, it will thus be possible to find a drilling machine by searching for an adminstration shell containing a submodel "Drilling" with appropriate properties. For communication between different I4.0 components, certain properties can then be assumed to exist. In an example like this, a second submodel, "energy efficiency", could then ensure that the drilling machine is able to save electricity when it is not in operation.

Figure 1: Possible submodels of an asset administration shell



Source: ZVEI

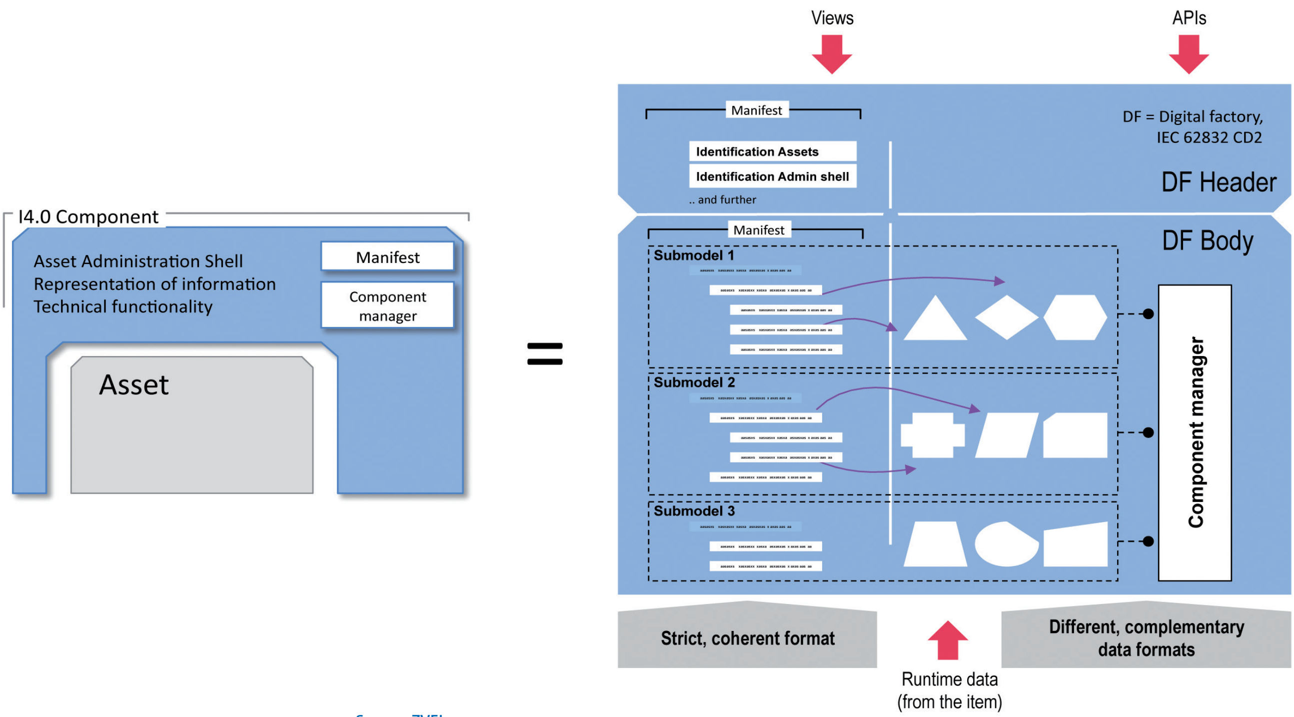
2.2 Basic structure of the asset administration shell

The last document regarding the asset administration shell ("Structure of the Administration Shell") presented a rough, logical view of the asset administration shell's structure. The asset administration shell – shown in blue in the following figure – is composed of a body and a header. The header contains identifying details regarding the asset administration shell and the represented assets. The body contains a certain number of submodels for an asset-specific characterisation of the asset administration shell.

Please note: The integrity of the asset administration shell itself must be protected if necessary. Depending on the requirement, confidentiality may also need to be guaranteed as a option.

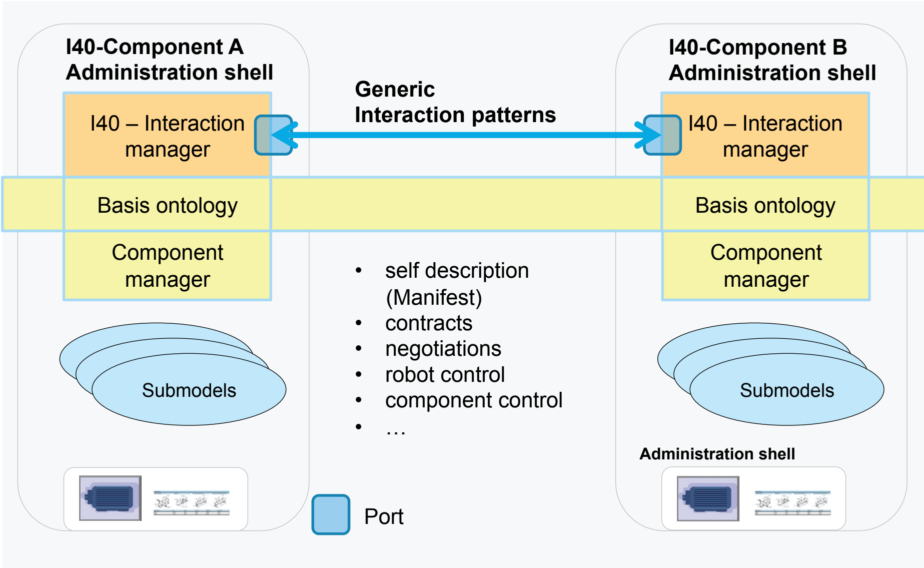
The properties, data and functions will also contain information that not every partner within a value creation network or even within an organisational unit should have

Figure 2: Structure of the asset administration shell



Source: ZVEI

Figure 3: The Plattform Industrie 4.0 Sub-Working Group 3 approach to "the language of Industrie 4.0"



Source: Prof. Diedrich, Plattform Industrie 4.0 Working Group 1, Ontology Sub-Working Group

access to or that requires its integrity and availability to be safeguarded. The structure of the asset administration shell should therefore take into account aspects such as access protection, visibility, identity and authorisation management, confidentiality, and integrity from the very start. A "no security" status can also be implemented if the risk analysis permits this.

Each submodel contains a structured quantity of properties that can refer to data and functions. A standardised format based on IEC 61360 is required for the properties. Data and functions may be available in various, complementary formats.

The properties of all the submodels therefore result in a constantly readable directory of the key information or, as it were, the Manifest of the asset administration shell and thus of the I4.0 components. To enable binding semantics, asset administration shells, assets, submodels and properties must all be clearly identified. Permitted global identifiers are ISO 29002-5 (e.g. eCl@ss and IEC Common Data Dictionaries) and URIs (Unique Resource Identifiers, e.g. for ontologies).

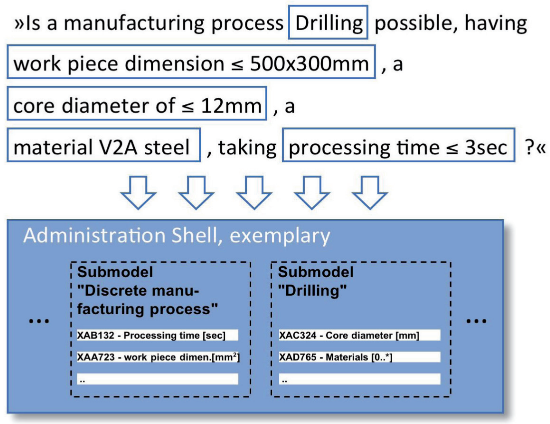
2.3 Interaction between I4.0 components

The "Ontology" sub-working group of Plattform Industrie 4.0 proposes a kind of language that can be used to map interaction patterns between I4.0 components.

For this purpose, an interaction manager for each I4.0 component is responsible for processing the interaction patterns in the I4.0 components network. A domain-independent basic ontology safeguards the connection with the domain-specific submodels in the asset administration shell.

An example of an interaction pattern could be a negotiation regarding capabilities for implementing a manufacturing process. During the negotiation, requirement and confirmation properties could be used that address individual, domain-specific submodels in the asset administration shell.

Figure 4: Example of how an interaction pattern is directed towards the domain-specific submodels in the asset administration shell



Source: ZVEI

2.4 Functions of I4.0 components

Integrating the assets represented by I4.0 components at a functional level requires a standardised description of the available functions (or capabilities) of the assets in question.

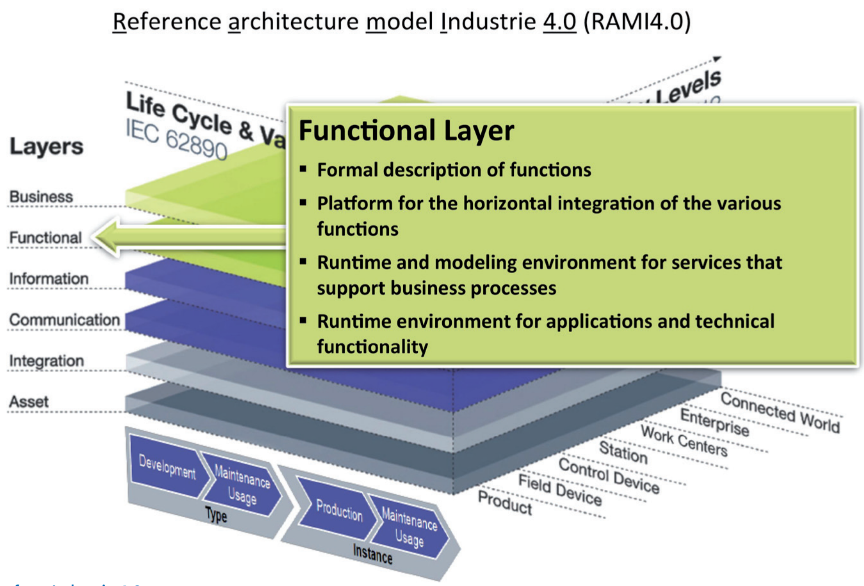
The functions an asset provides are described based on properties. This description is independent of the asset description and could, for example, be divided into the individual parts.

- Properties of the function (e.g. function type, parameters)
- Input variables
- Output variables

Input and output variables of functions could be information regarding materials, energy and information that is described with its relevant properties.

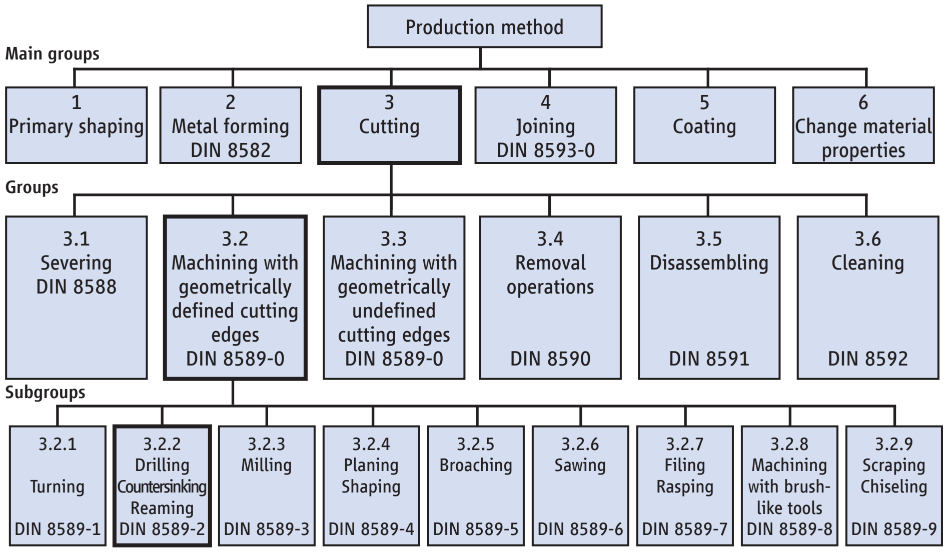
The description of a function can be divided into descriptions of sub-functions. Types of functions, parameters and relevent input/output variables can, for example,

Figure 5: Description of an I4.0 component at the functional level



Source: Plattform Industrie 4.0

Figure 6: Examples of submodels for functions: manufacturing process in accordance with DIN 8580



Source: Dr. Thomas Hardlich and ZVEI

be defined in appropriate submodels (e.g. based on DIN 8580, see figure).

Standardised submodels for describing functions can be used to define requirements for manufacturing products. For example, a product describes the requirements for necessary processing functions. These requirements can then be compared with the descriptions of a processing function provided by a specific production method.

The figure presents the derivation for the "drilling" submodel as an example. For an example description for this function, see Section 3.8.

Another example of a detailed description of requirements for automation functions is the lists of operating properties for process control devices (e.g. OLOP, operational list of properties, in accordance with IEC 61987-11).

3 Example Content

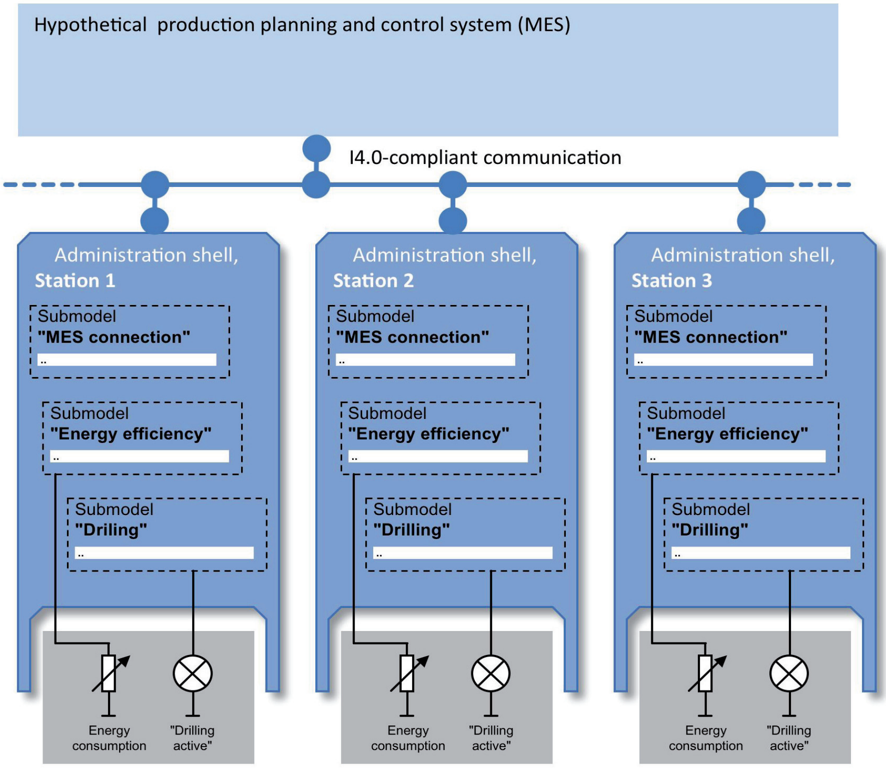
This section provides sample content for submodels. These examples serve solely to establish a synthetic scenario for the sake of the discussion relating to common understanding within Industrie 4.0. They do not claim to be representative, similarly structured or have similar scopes in relation to the actual submodels to come. Any subsequent similarity would be purely coincidental.

3.1 Scenario

Based on the presented content, the aim is to map three similar I4.0 components that are competing for a manufacturing order in a marketplace scenario (see Section 2.3). The execution of the manufacturing process should have different durations depending on the order data. A hypothetical production planning and control system (MES) is to control the marketplace scenario and, subsequently, multiple manufacturing executions.

The scenario presented thus complements the examples from the Ontology sub-working group without being too similar. This way, it is also not necessary to describe a large number of very different hypothetical submodels.

Figure 7: Example submodels for the scenario



Source: ZVEI

With this scenario, multiple units can quickly be bundled to form a clear demonstrator. There is also the option to connect real controls, sensors and actuators as assets. The following hypothetical submodels exist:

MES connection – Connects the station to a higher-level MES system via a small number of properties. Specifies whether the station is producing, ready for production, has a fault or is undergoing maintenance, for example.

Energy efficiency – Specifies some details regarding energy efficiency, for example. These can also be supplied by sensors.

Drilling – Contains a small number of properties and functions to start, terminate or simulate the drilling process. Allows control of an external actuator to display the activation in a demonstrator.

Documentation – Storage of documents as complex data such as PDF (not shown in the figure).

3.2 Definition of properties

This scenario also attempts to show that the actual definition of an property (or data element type) can be made in an external dictionary, such as IEC CDD or eCl@ss, while the instancing submodels in the asset administration shells use these definitions to specifically characterise the property. The same mechanism for property characterisation is also used when exchanging messages in the interaction model for I4.0 components.[1]

This scenario therefore assumes that the properties are defined in an external dictionary. All property definitions in this document are merely examples.

For this document, the hypothetical properties are defined with only a few data fields in accordance with IEC 61360. Further definitions should be made when these properties are required for real submodels. The data fields used here are:

[1] "Interaction model for Industrie 4.0 components", discussion paper by the Plattform Industrie 4.0 Ontology sub-working group; to be published in November 2016.

Table 1: Data fields for data element types (properties)

Field	English name	Explanation	Required	Example
ID (Kennung)	ID	Identifier in accordance with ISO 29002-5, usually hypothetical for this document. In individual cases, real property definitions may be used. Identifiers can also be defined as URIs[2]	Mandatory	BAA120
Versionsnummer	Version number	Number to distinguish the version of a data element type	Mandatory	007
Änderungsnummer	Revision number	Number to distinguish the revision of a data element type	Mandatory	01
(bevorzugter) Name	(preferred) Name	A name consisting of one or more words that is assigned to a data element type	Mandatory	Max. speed
Kurzbezeichnung	Short name	Abbreviated display of the preferred name for the data element type	Mandatory	
Symbol des Formelzeichens	Preferred letter symbol	Formula symbol of the data element type	Optional	n for rotations
Definition	Definition	Information that uniquely describes the meaning of a data element type and allows it to be distinguished from all other data element types	Mandatory	Highest permissible speed at which the motor or supply unit can be operated.
Quellendokument für die Definition des Datentypelements	Source document of definition	Reference to further documents that contain the definition	Optional	http://industrie-i40.org/2016/interaction/negotiation/property_type/task_ref_number
Datentyp	Data type	Data type that an IT implementation uses to represent values of this data type element[3]	Mandatory	INTEGER_MEASURE
Werteformat	Value format	Specifies the type and duration for displaying the values of this data type element[4]	Mandatory	NR1..5 or other
Maßeinheit	Unit of measure	Specifies the unit in which the value of a qualified data element type must be given	Mandatory, "n.a." permitted	1/min
Werteliste	Value list	Specifies the permitted values for a data element type	Mandatory, "n.a." permitted	0..8000

Note: The following text uses the abbreviated forms of IDs; for example, BAA120 with version 7 for the eCl@ss definition of a rotary speed of a synchronous motor that has the complete ID "0173-1#02-BAA120#007".

Note: The term "identification" is to be viewed in different ways. In this document, the term "identifer" is abbreviated to "ID" in tables. As an enhancement to the identifiers, the "Secure Identities" document from Plattform Industrie 4.0 working group 3 consciously refers to a model with several levels (identities, unique identities, secure identities) to provide different selection options depending on the specific use case.

[2] The specified indicator syntax is permitted in deviation to IEC 61360-1
[3] Data type specifications as are usual in IT are permitted in deviation to the definition in accordance with IEC 61360-1
[4] Syntax of the specifications deviates from the definition in accordance with IEC 61360-1

3.3 Characteristic properties – submodels

The hypothetical submodels for this example are shown in the form of simplified tables. To aid with clarity, the fields for the properties (or data element types) used are mirrored in the table; according to Section 3.2, the property are still defined in the respective dictionaries. Each of the submodels thus applies characterisations, usually as assurance or a measured value.

Table 2: Data fields for submodels

Field	English name	Explanation	Required	Example
Hierarchie	Hierarchy	Allows indication of the hierarchical and countable structures of the properties in the submodel by requirement (p)[5]	Mandatory	+-+
ID (Kennung)	ID	(See above)	Mandatory	
(bevorzugter) Name	Preferred name	(See above)	Mandatory	
Definition	Definition	(See above)	Mandatory	
Maßeinheit	Unit of measure	(See above)	Mandatory, "n.a." permitted	
Datentyp	Data type	(See above)	Mandatory	
Werteliste	Value list	(See above)	Mandatory, "n.a." permitted	
Wert	Value	Current value that can be specified through an instanced submodel (for instance in station 2) or through the asset, for example	Optional	2250 1/min
Ausprägungsaussage	Expression semantic	Specifies which role the property plays in an interaction, i.e. which expression the provider of the property intends. Valid values are: • Requirement (for requests that are to be confirmed or rejected) • Confirmations (for responses to requests that describe the capability of an asset) • Measurement (if a measured or actual value is provided)	Mandatory, "n.a." permitted	Requirement, Confirmation, Measurement
Ausprägungslogik	Expression logic	Specifies which function should be used if different expression logics are to be compared with one another	Optional	Equal, greater than or equal, less than or equal, between the value limits
Sicht	View	Indicates which view(s) that the property is to be associated with	Mandatory	Business
R / D / F / A / -	R / D / F / A / -	Indicates whether a reference, complex data content or a functionality is specified in the subsequent columns. "A" stands for Anmerkung (Comment)	Mandatory, "n.a." permitted	F
Inhalt	Contents	Description of the reference (What is referenced?) of the data content (What is referred to and in which format?) or the functionality (Where is this deployed? How is it represented? What does this functionality include?). If indicator = A, simply enter a comment regarding the content of the row.	Mandatory, provided '-' is not entered above	Function module library in accordance with IEC 61131 that is to be deployed in the next 61131-compatible control.

Note: In the examples presented from Section 3.6 onwards, "-" is used for "n. a.". This is solely to ensure better readability in this discussion paper.

[5] See paper "Structure of the Administration Shell – V2"

3.4 Agreement of submodels for various domains

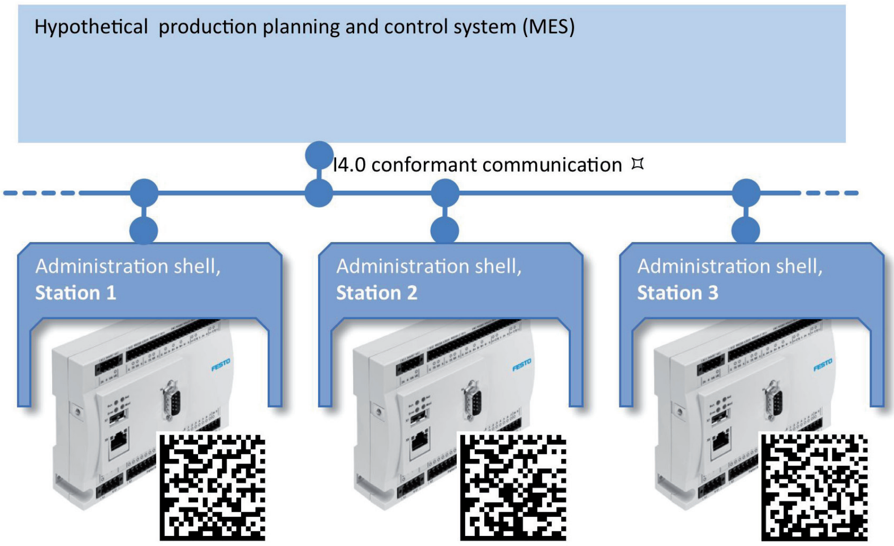
Representation in the form of a simple table also shows how a discussion regarding content and submodels with many different expert groups can also be implemented in a decentralised way.

We thus also recommend using a table like this as a basis for discussions. This tool can also be a component of a consultancy process[6].

3.5 Sample information for the general asset administration shells

The following figure (fig. 8) provides an overview of the identifiers that are created in the three hypothetical asset administration shells for the three stations. The DF header[7] mostly contains the identificators-for the asset administration shell and the assets concerned; in the example, this is one control device each.

Figure 8: Identification in an example scenario



		Asset administration shell		
		Station 1	Station 2	Station 3
DF header	Asset administration shell ID	http://www.zvei.de/demo/11232322	http://www.zvei.de/demo/11232342	http://www.zvei.de/demo/11282322
	ID(s) of asset(s)	[http://pk.festo.com/3S7PLFDRS35]	[http://pk.festo.com/3S7PL9X6K32]	[http://pk.festo.com/3S7PLFNCKDZ]
DF body	ID SM “MES connection”	Type: ADA011 Instance: http://www.zvei.de/demo/2368473473829	ADA011 Instance: http://www.zvei.de/demo/1366423771829	ADA011 Instance: http://www.zvei.de/demo/8364423571326
	ID SM “Energy efficiency”	ADA012 [..]	ADA012 [..]	ADA012 [..]
	ID SM “Drilling”	ADA013 [..]	ADA013 [..]	ADA013 [..]
	ID TM “Documentation”	ADA014 [..]	ADA014 [..]	ADA014 [..]

Source: ZVEI

[6] See paper “Structure of the Administration Shell – V2”, Section 4.1
[7] DF due to Digital Factory, IEC 62832 CD2

3.6 "MES connection" submodel

This hypothetical submodel aims to show a connection to an MES system in the simplest possible way. The example given here simply reflects the current status of a machine; any influence, for example, would be performed by another submodel.

Table 3: "MES connection" submodel

Hier-archy	ID	(preferred) Name	Definition	Unit of measure	Data type	Value list	Value	Expression semantic	Expression logic	Views	R/D/ F/A/-	Contents
						Property definition	Property characterisation					
I	AAA020	Asset production status	This property determines, if the associated asset is able to execute a production task at the time being.	–	ENUM	{Idle, Running, Failure, Restrained, Scheduled down, Unscheduled down}	Running	Measurement	Equal	Business	D	-
I	AAA021	Operating hours	This property determines, how long cumulatively the associated asset was switched on ‚mains'.	s	INT64	0..*	153453 s	Measurement	Equal	Performance	D	-

Note: The formulation of a real submodel would be based on the definition of the MES working group at ZVEI, for example.

3.7 "Energy efficiency" submodel

This hypothetical submodel aims to provide an example of how assets and their asset administration shells can provide current consumption values.

Table 4: "Energy efficiency" submodel

			Property definition				Property characterisation					
Hierarchy	ID	(preferred) Name	Definition	Unit of measure	Data type	Value list	Value	Expression semantic	Expression logic	Views	R/D/F/A/-	Contents
+--	AAB010	Electrical energy	This is a group of properties concerning about electrical energy consumption.	-	-	-	-	-	-	Performance	-	
--|	AAB011	Electrical consumption actual	Current, actual electrical consumption.	W	REAL	0..*	93.6 [W]	Measurement	Equal	Performance	-	
--|	AAB012	Electrical consumption cumulative energy	Integrated electrical consumption over time.	Wh	REAL	0..*	118.86 [Wh]	Measurement	Equal	Performance	-	
--|	AAB013	Electrical consumption cumulative start date	Date and time the integration of electrical consumption was started.	-	UTC Date & Time	n/a	2002-05-30T09:30:10Z	Measurement	Equal	Performance	A	For XML UTC time format see: http://www.w3schools.com/xml/schema_dtypes_date.asp
+--	AAB020	Pneumatic energy	This is a group of properties concerning about pneumatic energy consumption.	-	-	-	-	-	-	Performance	-	
--|	AAB021	Actual supply pressure	Supply pressure of the asset sensed at the inlet of the asset.	bar	REAL	0..*	8 [bar]	Measurement	Equal	Performance	-	
--|	AAB022	Pneumatic consumption actual	Current, actual pneumatic consumption.	l/h	REAL	0..*	212 [l/h]	Measurement	Equal	Performance	-	
--|	AAB023	Pneumatic consumption cumulative energy	Integrated pneumatic consumption over time.	l	REAL	0..*	3424 [l]	Measurement	Equal	Performance	-	
--|	AAB024	Pneumatic consumption cumulative start date	Date and time the integration of electrical consumption was started.	-	UTC Date & Time	n/a	2002-05-30T09:30:10Z	Measurement	Equal	Performance	-	

3.8 "Drilling" submodel

This hypothetical submodel aims to provide an example of how confirmation properties can be set and functionalities for simulation and program execution can be accessed.

Table 5: "Drilling" submodel

				Property definition			Property characterisation					
Hierarchy	ID	(preferred) Name	Definition	Unit of measure	Data type	Value list	Value	Expression semantic	Expression logic	Views	R/D/ F/A/-	Contents
I	AAC001	Drill tool diameter max.	Maximum diameter of drill tool which can be tooled in	mm	REAL	0..*	12 [mm]	Confirmation	Less Than	Performance	-	-
I	AAC002	Drill revolutions per minute max.	Maximum revolutions per minute for drill while drilling	1/min	REAL	0..*	2000 [1/min]	Confirmation	Less Than	Performance	-	-
F-=	AAC003	Simulate drill time	Determined by simulation or estimated the process time for whole drilling process	sec	REAL	0..*	0.21 [sec]	Confirmation	Less Than	Performance	F	Synchronous function call, taking the input parameters (AAC004.. AAC007) and returning one REAL
--I	AAC004	Drill tool diameter	Tool diameter to use	mm	REAL	0..*	5 [mm]	Requirement	Equal	Performance	-	
--I	AAC005	Drill feed rate	Feed rate to be used	mm/sec	REAL	0..*	3 [mm/sec]	Requirement	Equal	Performance	-	
--I	AAC006	Drill depth	Depth to drill to	mm.	REAL	0..*	8.2 [mm]	Requirement	Equal	Performance	-	
--I	AAC007	Work piece material	Material class to drill in	-	–> CAA001	-	CAA005	-	-	Performance	-	
F-=	AAC008	Start drill program	Starting preconfigured drill program	-	–> CAB001	-	-	-	-	Performance	F	Asynchronously starts the drill program and returns immediately with success/error
--I	AAC004	Drill tool diameter	Tool diameter to use	mm	REAL	0..*	5 [mm]	Requirement	Equal	Performance	-	
--I	AAC005	Drill feed rate	Feed rate to be used	mm/sec	REAL	0..*	3 [mm/sec]	Requirement	Equal	Performance	-	
--I	AAC006	Drill depth	Depth to drill to	mm	REAL	0..*	8.2 [mm]	Requirement	Equal	Performance	-	
--I	AAC007	Work piece material	Material class to drill in	-	–> CAA001	-	CAA007	Requirement	Equal	Performance	-	
--I	AAC009	Drill position X	X coordinate to drill	mm	REAL	0..*	12 [mm]	Requirement	Equal	Performance	-	
--I	AAC010	Drill position Y	Y coordinate to drill	mm	REAL	0..*	42 [mm]	Requirement	Equal	Performance	-	
F-=	AAC011	Abort drill programm	Abort current drill program	-	–> CAB001	-	-	-	-	Performance	F	Asynchronously aborts the drill program and returns immediately success/error

The following classification of materials, also based on properties, is hypothetically assumed:

Table 6: Example classification of materials

Hierarchy	ID	Name	Definition
+	CAA001	Material	
+-+	CAA002	Metal	
+-+-+	CAA003	Non-alloy	
+-+-+-+	CAA004	Steel	
+-+-+-+-l	CAA005	S275JR	
+-+-+-+	CAA006	Aluminum	
+-+-+-+-l	CAA007	AW-6060	
+-+-+-+-l	CAA008	AW-7020	
+-+-+	CAA009	Alloy	
+-+-+-+	CAA010	Copper	
+-+-+-+-l	CAA011	CR004A	

The following classification structures general success/failure values for program calls. Hierarchies can be set up for more specific error messages. This will make it possible to easily check for OK/NOK classes at the same time.

Table 7: Example classification of success/failure values

Hierarchy	ID	Name	Definition
l	CAB001	OP OK	Operation successful
+	CAB002	OP NOK	Operation unsuccessful
+-l	CAB003	OP INV	Operation unsuccessful, because preconditions were invalid/ not met

3.9 "Documentation" submodel

This hypothetical submodel aims to show how complexdata contents can be referred to by the submodels.

Table 8 "Documentation" submodel

			Property definition				Property characterisation					
Hierarchy	ID	(preferred) Name	Definition	Unit of measure	Data type	Value list	Value	Expression semantic	Expression logic	Views	R/D/F/A/-	Contents
+--	AAD001	Documentation item	Groups multiple properties towards an item.	-	Set of properties	-	-	-	-	Design	A	Multiple items with the same ID "AAD001" shall be possible.
--|	AAD002	Asset ID	Respective asset ID of documentation item	-	STRING	-	http://pk.festo.com 3S7PLFDRS35	Confirmation	Equal	Design	A	"" for default, if only one asset in administration shell.
--|	AAD003	Doc. item type	Type of documentation	-	–> CAC001	-		Confirmation	Equal	Design	-	
--|	AAD004	Doc. item title	Title of documentation	-	STRING	-	"Analogue modules for .."	Confirmation	Equal	Design	-	
--|	AAD005	Doc. item file name	File name of the associated data file, as provided by the supplier	-	STRING	-	"CPX_AM01.PDF"	Confirmation	Equal	Design	-	
--|	AAD006	Doc. item version	Version of the documentation	-	STRING	"1.1"	"2.0.0"	Confirmation	Equal	Design	-	
--|	AAD007	Doc. item data format	Date format of the complex data object		–> CAD001	CAD001	CAD001 == PDF	Confirmation	Equal	Design	-	
--|	AAD008	Doc. item BLOB	Complex data object of the documentation item	-	BLOB	-	-	Confirmation	Equal	Design	-	

Note: For an actual definition of the submodel, the specifications of VDI 2770, of the Product Data working group or "Dublin Core" could be taken into account.

The following classification of various types of documentation is hypothetically assumed (in accordance with VDI 2770):

Table 9: Example classification of types of document field

Hierarchy	ID	Name	Definition
+	CAC001	Documentation	Every kind of documentation
+-+	CAC002	Technical documents	
+-+-l	CAC003	Technical Specification	Data record sheet, stress analysis, specification sheet, ….
+-+-l	CAC004	Drawings / Schematics	Exploded drawing, 3D model,
+-+-l	CAC005	Bill of materials	Bill of material
+-+-l	CAC006	Certifications	Atex certificate, declaration of conformity,..
+-+	CAC007	Activity related documents	
+-+-l	CAC008	Assembly / Implementing / Dismantling	Assembly instruction, floor plan, …
+-+-l	CAC009	Operation	Instruction for use, IBN instruction
+-+-l	CAC010	Safety	Safety instructions
+-+-l	CAC011	Inspection / Maintenance/ Assessment	Maintenance timetable, calibration instruction, ..
+-+-l	CAC012	Repair / Service	Repair instruction, spare part list, …
+-+	CAC013	Contract documents	
+-+-l	CAC014	Contract documents	Bill of delivery, invoice, …

The following classification of permitted file formats is hypothetically assumed:

Table 10: Example classification of file types for document fields

Hierarchy	ID	Name	Definition
+	CAD001	Documentation data formats	Allowed data formats for I4.0 Documentations
--l	CAD002	PDF	PDF file, cold standard
--l	CAD003	HTML	Single file HTML file

3.10 Discussion of individual properties

This section looks in more detail at individual aspects of the properties used; for example, in relation to a possible technical implementation, a demonstration scenario, an explanation, etc. The properties, including the individual entities of the submodels, are referenced through the property ID.

Table 11: Discussion of individual properties

ID	Discussion
AAA001	Use of an ENUM. How can this be "meaningfully" accessed via the API of the asset administration shell? Inspired by SEMI/OEE. See http://www.oeestandard.com/eng/eng_4_definition.html. It is necessary to check whether a return value for a classification should be returned instead of an ENUM (→AAC008); this would enable a highly granular classification; for example that of a standstill.
AAA002	Should count up monotonously at one-second intervals. Can be mapped in an asset (control) with a remanent variable, for example.
AAB010	This property is used to organise a group of properties relating to "electrical energy" hierarchically in the submodel. This property therefore does not have direct property values and could also be created as purely organisational in another "Dictionary" or as a URL, such as "www.zvei.de/demo/9892843".
AAB013	Here, it is necessary to clarify whether IEC61360 is aware of a unique representation for the date and time. Otherwise the XML specification could be used. Saving in UTC (without a time zone) shall be provided for.
AAC001 .. AAC002	These properties show (in an entirely inadequate way!) the outline specification of the "Drilling" process capability (confirmation properties?). A simple check at the individual property level would probably not be sufficient to identify suitable assets for a workpiece (→negotiation models in the Ontology sub-working group).
AAC003	This property refers to a "Simulate drill time" function that maps multiple input parameters to an output parameter. The "result property" is also the return value for this function.
AAC004 .. AAC007	These properties are the input parameters for the function. It is necessary to check whether they should be mapped purely as an property of the submodel (case a) or whether they should actually only represent a semantic annotation of a function or procedure call in this table. Case a would be elegant from an informative standpoint and would also allow several function calls to be started in parallel at the same time. This would certainly be advantageous for planning. Case b, in which the parameters would effectively be transferred purely through properties is easier to design and would allow precisely the same properties to be used for several function or procedure calls (e.g. "Simulate drilling", "Emulate drilling" and "Execute drilling").
AAC007	The "Work piece material" property refers to sub-elements of a classification starting from "CAA001". This allows a semantically unique and consensually agreed classification of materials to be processed.
CAA001 .. CAA011	Individual materials, classified in a structured manner, that could be used for a manufacturing process. Note: This classification should not be underestimated. For steel, for example, there are hundreds of steel types according to various standards; the properties of these steels vary considerably.
AAC008	This property refers to the programm call "Start drill program". Unlike AAC003, this call starts an asset functionality with a longer duration that most likely blocks resources, namely execution of the drilling program. Property "AAA001" (MES connection) should be set to "Running" accordingly. The value of the call returned asynchronously straight away, i.e. the actual property, refers to a classification that structures the general failure values for program calls. In this way, either a simple Yes/No or a more detailed error message can be returned.
CAB001 .. CAB003	Individual error messages of a programm call, classified in a structured manner.

AAC009 .. AAC010	Drilling position to be used for the "Drilling" program call. It is necessary to check how multiple drilling positions and parameter sets could be transferred as a bundle.
AAC011	Terminates the drilling program that is currently running. Refers to the corresponding asynchronous functionality. Property "AAA001" (MES connection) should be set to "Idle" accordingly.
AAD001	It should be possible to store several different documentations in the asset administration shell for each asset. Multiple entries are required for each documentation.
AAD002	It should be possible to save documentation for a specific asset of the asset administration shell if the asset administration shell refers to multiple assets.
AAD003	This property in turn classifies various types of documentation with a reference to classification hierarchy "CAC001".
AAD005 .. AAD006	The file name and version should be retained to allow coordination with the manufacturer's server content, too.
AAD006	It is assumed that only one version of documentation will usually be made available in the asset administration shell. This version should correspond to the hardware and software configuration of the asset.
AAD007	This property in turn classifies various file formats with a reference to classification hierarchy "CAD001". Note: From the perspective of maximum stringency, the aim should be for the submodel to specify this file format; not the classification dictionary "used", such as eCla@ss.
AAD008	This property refers to a complex data object in the asset administration shell ("BLOB").
CAD002	It is necessary to check whether it is possible to add images to a simple HTML file in accordance with the standard.

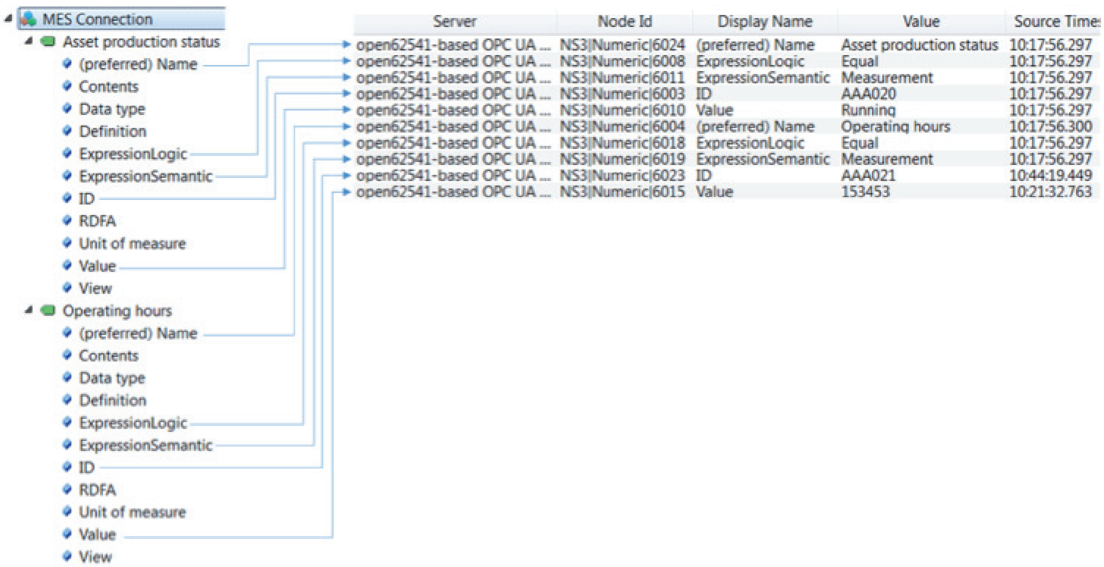
4 Presentation of the Submodels in Example Implementations

4.1 OPC-UA view of the "MES connection" submodel

The following section shows how the information from a submodel from Section 3 would be displayed for an example implementation on accessing systems and users.

The following figure provides an example of how the submodel from Section 3.6 "MES connection" is displayed in the model interaction with an OPC-UA client. Only selected data elements are listed. These are sorted alphanumerically, not in the order of the table.

Figure 9: Example view of a submodel in an OPC-UA client



Source: Florian Palm, RWTH Aachen University, project "Open AAS"

All the submodel information for an OPC-UA client is thus available for reading and writing[8] and can be browsed hierarchically. Of course, this does not affect access via message-oriented, I4.0-compliant communication.

[8] Subject to corresponding access rights.

List of authors

Dr. Heinz Bedenbender, VDI - Verein Deutscher Ingenieure e. V.
Meik Billmann, ZVEI - Zentralverband Elektrotechnik- und Elektronikindustrie e. V.
Prof. Dr. Ulrich Epple, RWTH Aachen University
Dr.-Ing. Thomas Hadlich, Otto-von-Guericke University Magdeburg
Martin Hankel, Bosch Rexroth AG
Dipl.-Ing. Roland Heidel, Roland Heidel Kommunikationslösungen e.K.
Oliver Hillermeier, SAP SE
Dr.-Ing. Michael Hoffmeister, Festo AG & Co. KG
Dipl.-Ing. Haimo Huhle, ZVEI - Zentralverband Elektrotechnik- und Elektronikindustrie e. V.
Michael Jochem, Robert Bosch GmbH
Markus Kiele-Dunsche, Lenze Automation GmbH
Gunther Koschnick, ZVEI - Zentralverband Elektrotechnik- und Elektronikindustrie e. V.
Dr. Heiko Koziolek, ABB AG
Lukas Linke, ZVEI - Zentralverband Elektrotechnik- und Elektronikindustrie e. V.
Dr. Steffen Lohmann, Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS)
Florian Palm, RWTH Aachen University – Chair of Process Control Engineering
Reinhold Pichler, DKE German Commission for Electrical, Electronic and Information Technologies in DIN and VDE
Dipl.-Ing. Stefan Pollmeier, ESR Pollmeier GmbH
Dipl.-Ing. Benedikt Rauscher, Pepperl + Fuchs GmbH
Frank Schewe, Phoenix Contact Electronics GmbH
Karsten Schneider, Siemens AG
Bernd Waser, Murrelektronik GmbH
Ingo Weber, Siemens AG
Prof. Dr.-Ing. Martin Wollschlaeger, TU Dresden (Computer Science)
Dr. Marcus Zinn, Schneider Electric Automation GmbH

List of figures and tables

ZVEI - German Electrical and Electronic
Manufacturers' Association
Lyoner Strasse 9
60528 Frankfurt am Main, Germany
Phone: +49 69 6302-0
Fax: +49 69 6302-317
E-mail: zvei@zvei.org
www.zvei.org

Title page image credit: ZVEI