Semantic Interoperability for Asset Communication within Smart Factories

Conference Paper · September 2017

DOI: 10.1109/ETFA.2017.8247689

9 authors, including:

Christian Diedrich
Otto-von-Guericke-Universität Magdeburg

Alexander Belyaev

Florian Pethig
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IO…

Tizian Schröder
Otto-von-Guericke-Universität Magdeburg

Semantic Interoperability for Asset Communication within Smart Factories

Christian Diedrich, ifak - Institut für Automation und Kommunikation e. V., Magdeburg, Alexander Bieliaiev - Otto-von-Guericke-Universität Magdeburg, Alexander Willmann - Fraunhofer FOKUS, Berlin, Heiko Koziolek - ABB, Ladenburg, Florian Pethig - Fraunhofer Anwendungszentrum Industrial Automation (IOSB-INA), Lemgo, Oliver Niggemann - Fraunhofer Anwendungszentrum Industrial Automation (IOSB-INA), Lemgo, Tizian Schröder – Otto-von-Guericke-Universität Magdeburg, Jens Vialkowitsch, Robert Bosch GmbH, Stuttgart, Thomas Usländer – Fraunhofer IOSB, Karlsruhe, Jörg Wende - IBM Deutschland GmbH, Dresden – all Germany

Abstract

Industrie 4.0 (I4.0) aims at a manufacturer-independent, vertical- and horizontal-oriented communication and cooperation within smart factories. This is only manageable using international standards. The so-called reference architecture model for Industrie 4.0 (RAMI4.0) and the requirement specification of an I4.0 component are already available. RAMI4.0 could be the basis for the interoperability of the interactions. The working group "Semantic and interaction model for I4.0 components" (GMA 7.20) made progress in this direction. The language used for the interaction of I4.0 components needs model definitions for the interaction consisting of the structure of the components, syntax for the description of the messages and the means to assign the meaning to the language elements. This paper discusses the results for a broader audience.

Keywords: Smart manufacturing, smart factory, flexible manufacturing, Interoperability, Semantic, Interaction model

1 The basic elements of I4.0 components for smart manufacturing

High-level flexibility of smart manufacturing requires increasing interactions between the components of a system as well a higher ability to react to changing requirements and requests. The concept of the Asset Administration Shell (AAS) [1] is the basic means defined by Industrie 4.0 (I4.0) [2] to fulfill these requirements. An I4.0 system consists of I4.0 components that comprise an asset (possibly composed of other assets) and an AAS. The interactions between I4.0 components are using the interfaces of the AAS to perform I4.0 scenarios. These interactions can be horizontal, i.e. between components of the same plant and factory level, or vertical, i.e. from the product and sensor level up to the business level within or even beyond the boundaries of an enterprise (Figure 1).

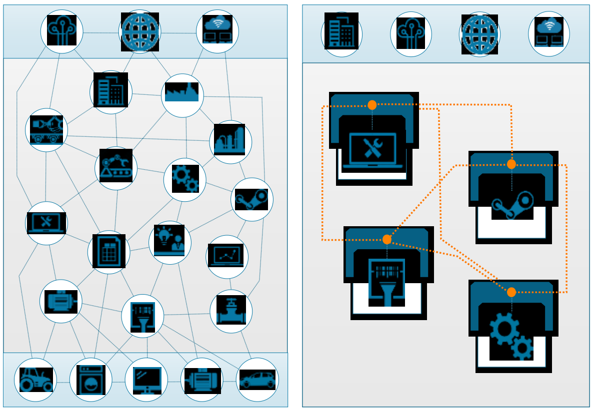

Figure 1 - Asset Administration Shells perform I4.0 interactions

The AAS represents data and functions of an asset in the form of "sub-models" (Figure 2). A sub-model provides information related to the asset as well as access to the capabilities (functions) of the asset. These can be documentation files, order information, certifications, configuration parameters, access to certain control functions, task assignments and others more. One AAS can support multiple sub-models. Properties of a sub-model refer to the concepts and parameters of the IEC Common Data Dictionary (IEC 61360 [10]) which is a common repository of concepts for all electro-technical domains [5].

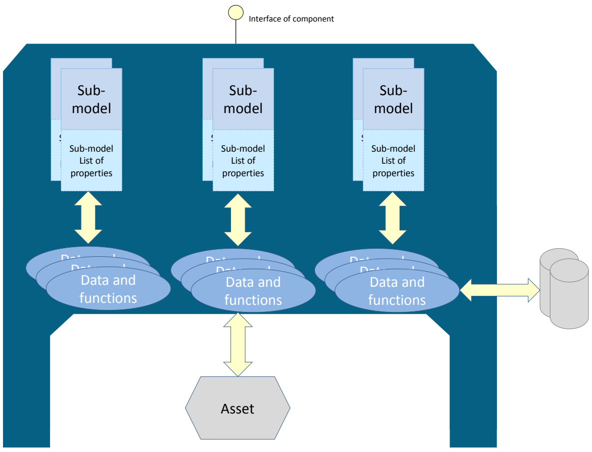

Figure 2. Interaction concept between I4.0 components

Properties and parameters of the sub-models can be accessed via a set of services and interactions (Figure 3). Communication services, such as read and write, build the lowest level of access. Additionally, discovery services enable other I4.0 components to search for and find functions and data offered by the I4.0 component. The concept of smart manufacturing includes new functionality such as negotiations about the capabilities of I4.0 components to fulfill a specific task with defined properties in a defined time period, with a defined quality of service and associated with a defined cost range. These application-related interactions build the top level of the service hierarchy. A proposal how to bring together classical services such as OPC UA (see section 5) and interaction protocols is defined in [8]. Figure 3 illustrates the different kind of services and interactions according to the "layer" dimension of the RAMI4.0. This paper focuses on the concept of the application-related, negotiation-like interactions.

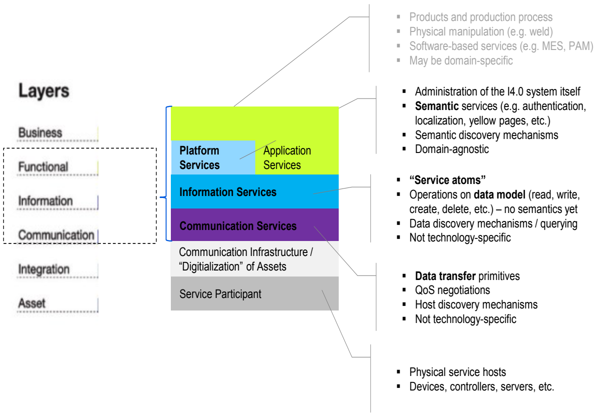

Figure 3. Service and interaction layers of I4.0 components [4]

2 Interaction Requirements in Industrie 4.0 Systems

The digitalization of production components made possible by the emergence of smart devices (equipped with both smart sensors and actuators) enables new types of applications with new architecture patterns. This new holistic approach is commonly referred to as "smart manufacturing". Among others there is the order-driven production scenario. It focuses on the autonomous and automated networking of the production capabilities with the aim of automated planning, distribution and control of production order. All required production systems shall be entitled to interoperate automatically and in real-time according to the current state of the production process. It also includes the monitoring of the running production processes including the operation of machines and equipment. The information about the product to be produced, the processes that produce the product step-by-step, and the technical resources needed to execute the processes are defined during the planning steps of the engineering phase. In the context of the German Industrie 4.0 Initiative not only production planning and control tasks, but also work planning tasks should be automated, in order to be able to react adequately to the increasing dynamics of the production environment.

In order to describe the interactions between I4.0 components the formalized process description is used as a tool for describing the production capability for order-driven production. The formalized process description (FPD) (according to the German Guideline VDI / VDE 3682) is a modelling approach used to develop a consistent usage of the process description and of the technically and logical representation of information for the intended operation and planning of the technical system throughout the life cycle in a clear, structured, comprehensive way.
This is achieved by a formalization of the process description. Within the according guideline formalization means the reduction to a fixed quantity of:

- symbols,
- rules for allowed combination of signs,
- operations with symbols according to their meaning, and
- the information model for all FPD objects.

This graphical description of a production process is constructed as follows. A production process is described by a graph in which physical, chemical, biological as well as process automation and factory automation-based states are connected via process operators. Through this connection a state *ante* (input of the process operator) is transferred to a state *post* (output of the process operator). Here, a state comprises multiple views, e.g. information about the product (material, substance etc.), energy needs and usage as well as and (meta-)data sets. The process operator transforms given inputs into new states of products, energies and information. To carry out this transformation the process operator uses technical resources. More than one product, energy and information can serve as input or output of the process operator. A delimitation of the system from the environment is ensured through the balancing space, which encloses the graph at the defined inputs and outputs. An example for individual symbols of the above stated description of the technical process is shown in Figure 4. The process within the balancing space is structured via decomposition and composition, which takes place exclusively in process operators. Furthermore, each object of the graph (product, process and technical resource) is described by a set of properties, which are included in a defined information model. The coupling of input product, transforming process, output product and technical resource will be further referred to as PPR binding (product, process, resource binding). For example, if a hole should be drilled in a work piece, the geometrical properties of the input product will be changed. However, other characteristics of the product may also be important, for example, the toughness of the material. The product transformation is described by the process operator. In this example this operator is "drilling". A drilling machine, a laser cutting machine or a milling machine may be used as a technical resource that performs the product transformation. The properties of the process include, for example, quality characteristics such as drilling diameter and tolerances, surface roughness and the duration of the transformation (drilling process). The technical resource has characteristics that describe the capabilities of the machine, e.g. the possible drill diameters, the rotational speeds, the range of the engagement angles and the above mentioned quality characteristics.

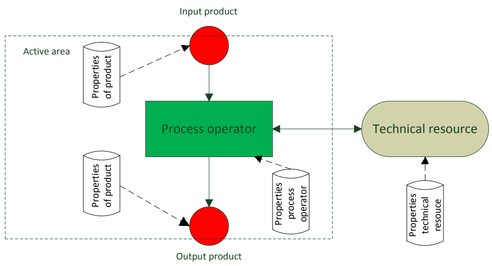

Figure 4 Graphical representation of the formalized process description (without energy and information)

In order to carry out a production step, the transformation from the input to the output product must be defined and the production system (or its components) (technical resource) required for this purpose must be selected. In I4.0 systems the allocation of the process steps to the technical resources and the process step for the product transformation should not be performed manually, but automatically during the operation. This allocation is shown in Figure 5.

In I4.0 systems, assets are represented by AAS. The selection of a technical resource to a required process operator should be reached through negotiation between the I4.0 components (Figure 5). This negotiation requires interactions.
Apart from that, further interactions are needed during the operation for the detailed control and monitoring of the production processes, as well as for maintenance and diagnosis of the technical resources.

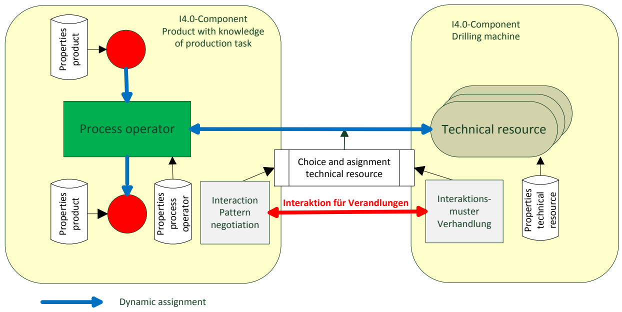

Figure 5 I4.0-components allocate by the process operator and technical resource

The negotiation interactions and the interactions in the operation and maintenance processes are described below. The functions of the I4.0 component are accessible via the interaction model (Figure 6). The assets are controlled by the corresponding functions in the AAS. The interaction model comprises messages that are exchanged between I4.0 components i.e. their sub-models. Further messages can cause state transitions in them. The behavior model is based on coupled automata. This model is described in [3].

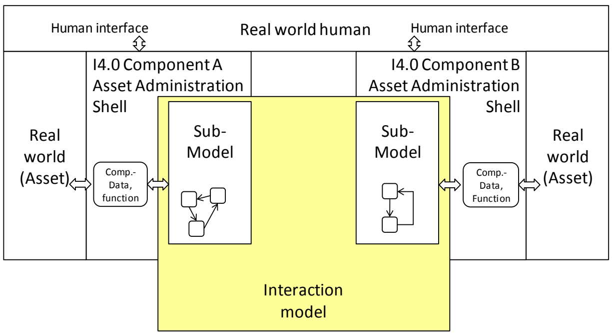

Figure 6 Placement of interactions between I4.0 components

3 Interaction concept

The AAS offers data and functions via sub-models to external parties for cooperation. Examples for negotiation oriented sub-models are "drilling", "transporting", "nego-

tiating", "concluding of agreements", "controlling a production process" (e.g., by using PackML), "diagnosing".
The functional and non-functional properties of the sub-models of the I4.0 components should be available and readable for user properties, property value statements (PVS), parameters and characteristics. Property descriptions have a fixed structure which is based on IEC 61360 that defines attributes of properties. This results in a template for properties that contains identifying, descriptive and value-related attributes. In addition, there are attributes of the properties that represent the assignment to one or more views or references to documents in which the specific property is standardized.

Table 1 Attributes of Properties

Field	**Description**	**Degree of supp-supp-port**	**Example**
Attribute nach IEC 61360-1 / ISO 13584-42			
ID	Identifier according to ISO 29002-5 [9] or URI	Required	ACB723
Version	Version of the attribute value	Required	V4.3.03.15
Name	Descriptive Name	Required	Leistungsaufnahme max.
Definition	Long description for classification of the attribute	Required	Maximum allowable power consumption of the device at its input terminals meassured in Watt
Shortname	technical acronym	Optional	
Symbol	technical abbreviation	Optional	i.e. P for electric power
SI-Unit	Standard-SI-Unit of the attribute	Required	[W]
Datatype	IT datatype to represent the attribute	Required	Real
Data representation	Binary representation of the attribute	Required	ASCII, Hex, decimal, …
Valid range	valid values or value ranges	Required	0..2000
Value	current value of the attribute	Optional	24,57
Additional I4.0-specific attributes			
….			

The interaction between I4.0 components are commonly realized by means of messages communicated via a communication system. Messages have a defined structure that is depicted in Figure 8. The message contains control information (sender ID, message type, address of the sub-model (submodel ID), control flags (if the message needs confirmation) and security and quality of service (QoS) properties), as well as the properties of the sub-models. Properties are listed in the manifest part of the sub-model in order to compare them to the received message. Agreements on additional "private" properties are possible in bilateral negotiation between the involved partners.

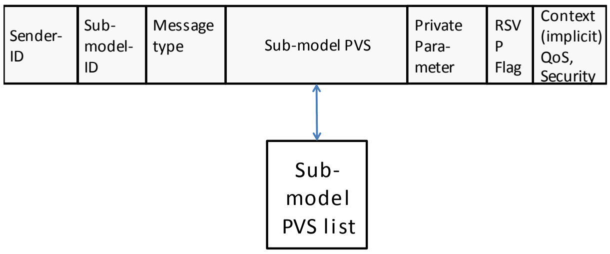

Figure 8 Basic message structure

In general, the interaction model consists of the definition of interacting partners (here the software components that implement the sub-models of the AAS in I4.0 components), the message content (e.g. properties of sub-models) and the sequences (represented by the sequence diagram). The sequences shown in the sequence diagram are part of the interacting automata [3]. A general concept is developed which consists of the message with its content, sequences of messages and rules for controlling alternatives in the sequence order. This concept is described by formal and semi-formal models and methods, e.g., sequence diagrams, state machines, classifications as well as property catalogs and, if necessary, ontologies, or, to enable interoperability on semantic level, too.
The later are formal vocabularies for specific knowledge domains, often modelled using directed and labeled multigraphs; e.g. by applying Semantic Web Technologies such as the Resource Description Framework (RDF) and by reusing and combining numerous existing ontologies.

4 Example

A product requests the initiation of a production step, e.g. it needs to be transported from a conveyor belt to the processing station. For this purpose, a robot with grippers is used. The production line is configured such that it may produce a variety of products. It is therefore necessary to check if the technical resources (here the robot with gripper) is available to handle the product. Therefore, the I4.0 component of the product (as described in Section 2) interacts with the I4.0 component of the robot and requests handing in accordance to the properties of the product and the desired movement. For example, properties can be information about mass, material and start and end positions of a pick-and-place task.
First of all, a valid contract is required that allows collaboration between the product and the robot. The AAS of

the robot checks the contract if a request from the product's AAS is received. The received message contains contract number (TaskRefNo in Figure 10) and properties with the requested property value statements that characterizes the product and the requested handling.

Now the AAS must check, if:

(1) the agreement number is valid and active (contract active?),

(2) if the requested properties are known and supported (match with the entries in the manifest (CheckProperties), and

(3) if the requested properties value statements can be transformed by the asset (CheckProperties).

The checks of (1) to (3) can be positive or negative. If the contract number is active, the characteristics check is started simultaneously. If this is not the case, an error message is issued. The properties are checked in a similar way. Unsupported properties and property value statements are reported.
If there is no positive answer the calling I4.0 component sub-model has to decide what to do, change the property values of stop the negotiation.

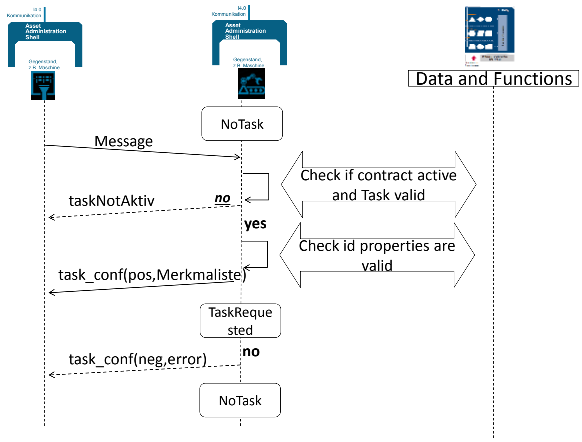

Figure 9 Interaction pattern: request using an existing agreement

The described sequence shows the short part of the interaction pattern example "Product Handling". There are even more details to be agreed upon in case of cooperation. Such cooperation is expressed in an automata, that describes (Figure 10) permitted messages and initiated state transitions, as well as possible reactions. Special interactions with the asset or other functions inside the AAS are typically deposited in accompanying statements and rules.

Table 2 shows some properties with their attributes according to IEC 61360.

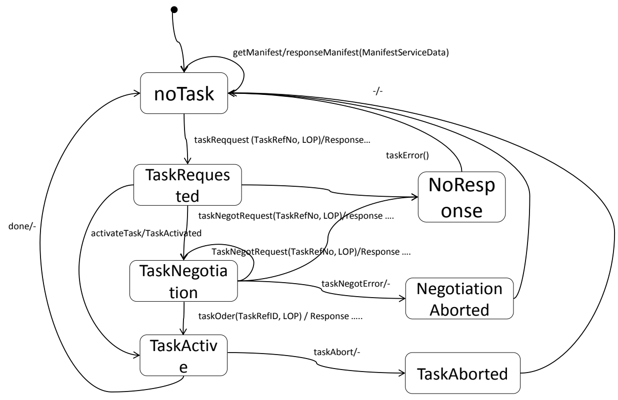

Figure 10 Negotiation behavioral model in the requested I4.0 component

Table 2 Example Definition of Properties

Attribute	TaskRef No	Attribute "Mass"	Attribute "Material of the clamping surface"
Attribute nach IEC 61360-1 / ISO 13584-42			
ID	http://plattform-i40.de/ag1/contract/TaskRefNo	0173-1#02-BAB576#005	0173-1#02-AAG804#003
Version	V0.9	V9.1	V9.1
Name	TaskRefNo	Mass (weight)	Material of the clamping surface
Definition	Alphanumeric number, referencing an existing contract	Force that garavitation exerts upon a body	Description of the material composition of the clamping surface
Short-name	TaskRefNo	Mass	Material
Symbol	-	m	-
SI-Unit	-	kg	-
Datatype	String	Float	String
Data Representation	ASCII	IEEE	ASCII
Valid Range	5-20 ASCII Characters	0 … 100	Materialcode
Wert (Example)	ABC_2457_DEF	15,4	0173-1#07-WAA113#003 - Plastic

The first property (second column) contains a reference to a contract, which regulates that this product type and the robot may cooperate. This example shows that URIs can be used for identification purposes. The name space "plattform-i40.de" indicates the name space of that contract area and the corresponding properties belong to a specification of the Plattform Industrie 4.0, namely a working group of the Plattform called “AG1 . A specification is needed defining this reference. The properties "mass" (third column) and "material of the clamping surface" (fourth column) are contained in ecl@ss [12]. In this example, it is assumed that the AAS of the robot responds with the material of the gripper to allow the product to prove the compatibility with the material of the product. For these two properties, the typical ecl@ss IDs according to ISO 29002-5 [9] can be used. The values, if they consist of a set of countable valid values of a property, can also comprise IDs.

A request message, which the product's AAS sends to the robot's AAS in order to initialize a production step, could look as follows:

“SID: I40_comp_Product_xyz; TID: http://plattform-i40.de/interaction_pattern/as/TaskNegotiation; MID: http://plattform-i40.de/interaction_pattern/message_type/ TaskReq=req; PID: http://plattform-i40.de/ interaction_pattern/contract/property_type/TaskRefNo; VALUE = ABC_2457_DEF; 0173-1#02-BAB576#005=21,2; “

The abbreviations have the following meanings: SID: Sender-ID; TID: Sub-Model-ID; MID: Message type; PID: property ID; AS – Administration Shell

5 Mapping to OPC UA

In order to implement interactions between AASs, a number of potential technical candidates exist. Besides the OMG Data Distribution System (DDS) and the oneM2M architecture, the Open Platform Communications Unified Architecture (OPC UA), standardized in IEC 62541 [6], is the major contender for I4.0 Communication. OPC UA defines services for the interaction with a server-side object-oriented information model. In order to use the information model, an OPC UA Client has to establish a secure connection to an OPC UA Server first. OPC UA information models are based on an address space consisting of uniquely identifiable nodes. Nodes are connected via references, that are hierarchical (i.e. parent-child relationships) or non-hierarchical (i.e. type-of relationship). The general OPC UA information model is defined in IEC 62541-5, and describes standardized nodes of a server’s (empty) address space. Based on the standardized address space and general information model, custom information models can be defined. The information model can be introspected and modified during runtime via the services standardized in IEC 62541-4. Reusable information models for specific domains of interest are released as Companion Specifications for increased interoperability.
OPC UA can be used as an example for the implementation of the interaction concept described in this paper.
Although the focus of this paper is on semantic interoperability, underlying communication should be considered as well.
The current state of the OPC UA standard is based upon the client/server communication paradigm. For this reason, I4.0 components, acting in OPC UA server and client roles, have to establish a secure connection before the messages specified in this paper can be exchanged. If the client knows the server, the connection can be established directly. However, before this is possible, the communication partners, here I4.0 components, have to get known to each other. This is typically performed by means of discovery services that enable a publish-find-bind architecture pattern. OPC UA includes discovery mechanisms that can be used to discover OPC UA endpoints. An OPC UA endpoint includes the security configuration of a certain communication channel to an OPC UA Server. For this reason, current OPC UA discovery cannot be used to discover functionality offered by I4.0 components that is included in the OPC UA information model. Nevertheless, after an OPC UA connection has been established OPC UA clients can browse the OPC UA information model, read, and write values. The OPC UA information model has to include the sub-models embedded in an AAS. With regards to the current state of I4.0 standardization, sub-models are based on Property Value Statements (PVS) about properties according to IEC 61360 [7]. A possible implementation scenario based on the current state of I4.0 standardization is depicted in Figure 11.

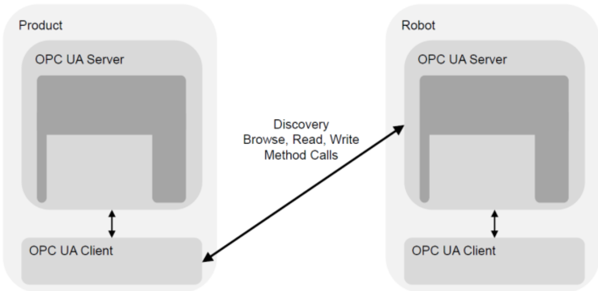

Figure 11: Implementation Scenario based on OPC UA Client/Sever

As introduced in Section 4, messages need to be exchanged, e.g. for the negotiation of a production order. For this purpose, the openAAS implementation for the AAS implements a message exchange scheme that is based on OPC UA method calls [7]. The AAS in the OPC UA information model comprises a mailbox. Messages can be delivered to this mailbox by calling a method of the OPC UA server. Nevertheless, usually a sender of a message does not need an established connection to the receiver of the message in message based interaction schemes (lose coupling) [11].
In our opinion, it should be possible to discover functions of I4.0 components in the future, too. Not only OPC UA endpoints should be discoverable. In addition to the client/server paradigm, OPC UA supports the publish/subscribe communication pattern. Publish/subscribe could be an interesting implementation option for I4.0 communication. I4.0 components could publish functionality to unknown receivers. A receiver could then initiate

the negotiation process via a standard OPC UA connection.
Additionally, it would be valuable to relate I4.0 standardization upon the OPC UA standardization approach. Existing OPC UA Companion Specifications already cover many aspects needed of I4.0 sub-models. One major aspect that Companion Specifications are missing is the representation of standardized information elements (PVS based on IEC 61360). Maybe PVS types could be included in a standardized OPC UA information model for I4.0 in the future by adding a dedicated relation called “semanticRef” for objects which contains PVS. The vision for future I4.0 communication based on OPC UA is depicted in Figure 12.

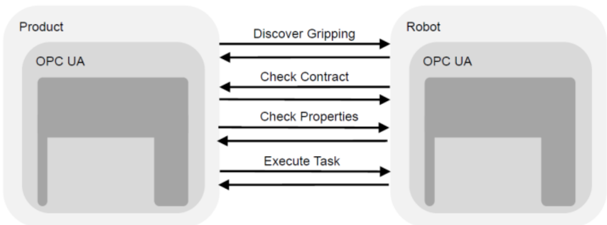

Figure 12: Future Scenario for OPC UA based I4.0 communication

6 Mapping to AutomationML

In addition to describe how to communicate (e.g. interactions mapped to OPC UA as discussed above) it is essential to also discuss what to communicate. This refers to semantics of the data being exchanged in interactions between I4.0 components. As described in section 3 the data transferred via messages need to be semantically related to data and function elements of the sub-models used in the AAS. The first step towards a common understanding is the use of the property identifiers as defined in the IEC 61360 series, and used, e.g., in eCl@ss, a cross-industry product data standard for classification and clear description of products and services [12]. A second step towards semantic interoperability is to agree upon data models that allow to describe asset characteristics and capabilities. Here, the IEC 62714 standard AutomationML [11] gains more and more attention and momentum. Being defined as an engineering data exchange format for use in industrial automation systems engineering, it may be used to describe the various aspects (sub-models) of the manifest within an AAS. AutomationML is a container standard. IEC 62714-1 defines the architecture, general requirements and basic concepts, among which is a profile of CAEX [14] as the so-called top-level format (core meta-data model based upon XML) that allows a modeler to express topological relationships between nodes (e.g., description of a plant structure, compositions between assets). Currently, the following libraries exist as extensions:

- IEC 62714-2: Role class libraries to derive necessary roles (links) for semantic definitions in individual application cases.
- IEC 62714-3: Geometry and to specify the modelling of geometry and kinematics information based on COLLADA.
- IEC 62714-4: Logic to specify the modelling of logics, sequencing, behaviour and control related information based on PLCopen XML.

Ongoing work is the integration of eCl@ass property definitions into AutomationML. Further parts may be added in the future in order to interconnect additional data standards to AutomationML.
As a consequence, it is to be discussed to which degree the requirements upon the AAS manifest may be currently realized by a mapping to AutomationML and which aspects are missing or may have to be defined.
Recently, an OPC UA companion specification as well as a DIN SPEC 16592 [13] has been published that defines a mapping from AutomationML models to an OPC UA information model. Hence, since then, a model-driven engineering of an I4.0 component may be realized. Starting from an AutomationML model expressing the properties and capabilities of an I4.0 component, finally an OPC UA server may be generated that may act as interface to an AAS implementation for the I4.0 component.

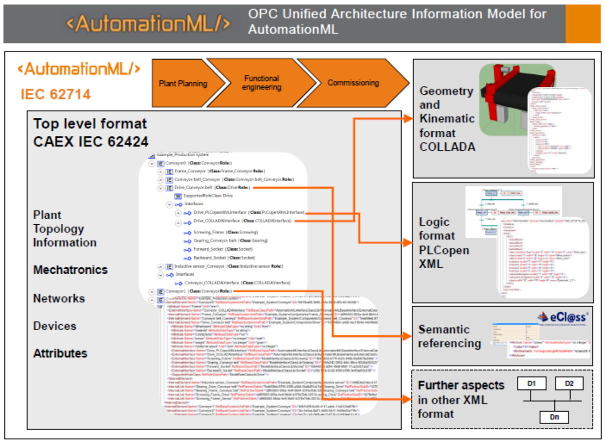

Figure 13: AutomationML base structure

7 Conclusion

The article shows the basic principles of an interaction model between I4.0 components which are the basic abstraction concept of manufacturing assets as defined in RAMI4.0. The I4.0 components together with the interaction model make up an I4.0 system. The article discusses required interactions and the essential elements of an interaction model. It is shown, that the interaction model consists of the definition of interacting partners by means of AAS and their sub-models, the message contents, the sequences of the messages for the special task execution. A small example is presented to show several elements of the interaction model. Furthermore, mappings to the two IEC standards OPC UA and AutomationML are presented as candidate solutions for the technology level.

8 Literature

[1] Industrie 4.0 Working Paper: Structure of the Administration Shell - Continuation of the Development of the Reference Model for the Industrie 4.0 Component. Published by Federal Ministry for Economic Affairs and Energy (BMWi) Public Relations 10119 Berlin. April 2016.

[2] http://iec2016.org/index.php/industry-4-0.html and DIN SPEC 91345 „Referenzarchitekturmodell Industrie 4.0 (RAMI4.0) April 2016. DIN ICS 03.100.01; 25.040.01; 35.240.50

[3] Bangemann, F., Reich, J., Diedrich, Ch.: A Grammar for the Semantics of Component Interaction of Cyber-Physical Systems. IEEE Interantional Symposium on Industrial Electronics, June 8-10. 2016 SF-006319 proceedings.

[4] GMA 7.21: Industrie 4.0 Service Architecture – Basic Concepts for Interoperability. VDI/VDE 2016.

[5] Christian Diedrich, Thomas Hadlich, Mario Thron: Semantik durch Merkmale für I40. Beitrag in in B. Vogel-Heuser et al. (Hrsg.), Handbuch Industrie 4.0, Springer NachschlageWissen, DOI 10.1007/978-3-662-45537-1_63-1. Online ISBN 978-3-662-45537-1.

[6] IEC/TR62541-1:2010: OPC Unified Architecture - Part 1: Overview and concepts, IEC Std., 2010.

[7] Palm et. al., "Standards für Industrie 4.0 - Dienste der Verwaltungsschale, Version 0.4", Whitepaper RWTH Aachen University, ACPLT - Lehrstuhl für Prozessleittechnik, 2016

[8] DIN SPEC 16593-2017-04 Reference Model for Industrie 4.0 Service Architectures – Basic concepts of an Interaction-based Architecture, Beuth Verlag, April 2017.

[9] ISO/TS 29002-5:2009 Industrial automation systems and integration -- Exchange of characteristic data -- Part 5: Identification scheme, 2009.

[10] IEC 61360-1 Standard data elements types with associated classification scheme for electric items - - Part 1: Definitions - Principles and methods, 2009.

[11] IEC 62714:2014 Engineering data exchange format for use in industrial automation systems engineering - Automation markup language, 2014.

[12] eCl@ass Classification and product description. http://www.eclass.eu.

[13] DIN SPEC 16592-2016-12, Combining OPC Unified Architecture and Automation Markup Language, Beuth Verlag, December 2016.

[14] IEC 62424:2016: Representation of process control engineering - Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools, 2016.