Industrie 4.0 service architecture

Semantic interoperability in Industrie 4.0 service systems

A key-enabler for the *self-organization of production systems* proclaimed by the Industrie 4.0 is a standardized core *service system*. This requires a certain degree of interoperability between participants of the service system that allows to perform *Plug and Produce* on top of it. The main expectations regarding such as service system are therefore to allow for easy discovery as well as machine readable semantics of data. The service architecture serves as basis for implementing the *Administration Shell* as well as Industrie 4.0 compliant communication. The underlying service architecture needs to describe the static base structure of data (the Industrie 4.0 Information Meta-Model), the interactive access interfaces for browsing and modifying data (the Industrie 4.0 Information Services) as well as addressing services within their service system. Based on this architecture high-level services for organizing the actual production and handling of products (the Industrie 4.0 Application Services), as well as the self-organization of resources and assets in the automation system (the Industrie 4.0 Platform Services) can be created in a t*echnology-independent and interoperable* way.

KEYWORDS Industrie 4.0 / service architecture / information model / semantics / meta-modeling / discovery

Industrie-4.0-Dienstearchitektur –
Semantische Interoperabilität in Industrie-4.0-Dienstesystemen

Die bei Industrie 4.0 angestrebte *Selbstorganisation der Produktion* wird im Wesentlichen ermöglicht durch ein im Kern normiertes *Dienstesystem*. Dieses benötigt dabei einen Grad von Interoperabilität zwischen Dienstesystemteilnehmern, der tatsächlich *Plug-and-produce* ermöglicht. Die Haupterwartungen an ein solches Dienstesystem liegen beim einfachen Auffinden (Erforschungsmechanismen) und Verstehen von Daten (maschinenlesbare Semantik). Die Dienstearchitektur, die dieser Beitrag behandelt, dient sowohl als Basis für die Implementierung der *Verwaltungsschale*, als auch der Industrie-4.0-konformen Kommunikation. Die zugrundeliegende Dienstearchitektur beschreibt dabei die statische Grundstruktur der Daten (das Industrie-4.0-Informations-Meta-Modell), die interaktiven Zugriffsschnittstellen zur Erforschung und Änderung der Daten (die Industrie-4.0-Informations-Dienste) sowie die Adressierung von Diensten in einem Dienstesystem. Auf Basis dieser Architektur können höherwertige Dienste zur Organisation der Produktion und Handhabung der Produkte (die Industrie-4.0-Anwendungsdienste), aber ebenso zur Selbstverwaltung der Ressourcen/Assets des Automatisierungssystems (Industrie-4.0-Plattformdienste) *technologieunabhängig und interoperabel* erstellt werden.

SCHLAGWÖRTER Industrie 4.0 / Dienstearchitektur / Informationsmodell / Semantik / Metamodellierung / Discovery

DIRK SCHULZ, THOMAS GOLDSCHMIDT, JOHANNES SCHMITT, ABB

Many key advantages of future Industrie 4.0 systems such as self-organization of production systems can only be achieved through a high degree of interoperability between production assets which participate in the Industrie 4.0 service system. Therefore, basic functionality such as discovery of services and semantic comprehension of data needs to be based on a common foundation. Furthermore, production-related and application-specific services also require such a common foundation in order to be interoperable.

The GMA FA 7.21 expert committee has outlined the cornerstones of such a service architecture for Industrie 4.0 in 2016 [2] [19]. Throughout 2017, individual aspects have been refined through GMA FA 7.21, the BaSys 4.0 public-funded project, and in part GMA FA 7.20. BaSys 4.0 [https://www.basys40.de/] has the objective to develop an Industrie 4.0 basis-operating-system for adaptable production. There are still unresolved questions on architecture aspects, but at the same time we find that the concepts in the initial publications are still valid.

We begin by relating the service architecture to published concepts of Industrie 4.0 such as the Reference Architecture Model Industrie 4.0 [1] and the Industrie 4.0 Administration Shell [3] in. Toward a technical architecture, we define a *service hierarchy* in accordance with ISO/IEC 7498-1 [4] and DIN SPEC 16593 [5] which separates low-level services for data access (information services) and higher-level services for platform management and applications (self-organization, production functions). The initial discovery of service participants is then introduced. The need for a common *information meta-model* is outlined in chapter 4 as an essential “common language” to describe the subjects of interaction (asset properties, skills, conditions, measurements, etc.). We also introduce resource- and message-oriented paradigms to access information, both of which are needed to interact with asset skills in a standardized manner.

1. ADMINISTRATION SHELL AND I4.0-CONFORMING COMMUNICATION

1.1 Administration shell

The main objective of the AS is to provide access to all information (data and descriptions of functions) which is relevant in the life-cycles of products and production systems (assets) [3]. Thus, the AS is *the* digitalization concept of Industrie 4.0, providing an information façade through which Industrie 4.0 components can mutually interact to advertise and use their skills. From the perspective of the service architecture, the administration shell (AS) is

1 | The sum of all information on an asset (*body*)
2 | Which can be accessed through Industrie-4.0-compliant communication (see section 1.2 and chapter 3 regarding information models and services)
3 | Which can be understood through an Industrie-4.0-defined semantics (manifest) or which follow a defined complementing data format (Industrie 3.0 standards)
4 | Within a defined organizational scope, e.g. one enterprise
5 | Discoverable through a defined mechanism (see chapter 3)
6 | Based on common asset identification data (*header* functionality)
7 | Regardless of the deployment of the individual *views* on (other) assets in that domain.

This implies that information related to an asset which is exposed through Industrie-4.0-compliant communication within an organizational domain is inherently part of the AS.

To implement the AS for particular types of assets like field devices, controllers, machine cells, or package units, the information defined by existing standards must be mapped into the AS body. The exact mapping – particularly the decision to describe information using a common meta-model (see section 4.2) or maintaining a complementing format (e.g. CAD files, manuals) – is outside of the scope of the service architecture.

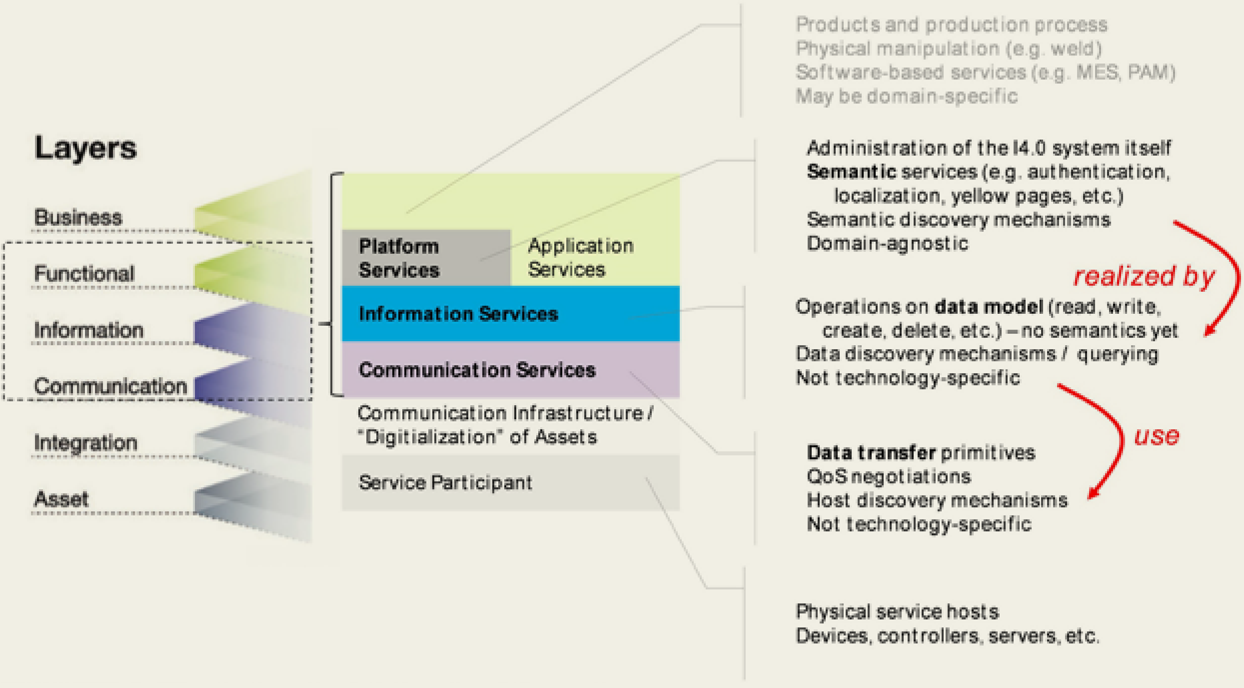

FIGURE 1: The Industrie 4.0 Service Hierarchy

The task of the service architecture is to provide means for accessing and representing any kind of asset-related information; common low-level services and information meta-models are essential to ensure interoperability for higher-level services. Interoperability is defined as a level of functional cooperation of service participants including *common protocols, interfaces, data access methods and data types, and common parameter semantics and application functionality* [6]. In the service architecture, this refers to the properties 2, 3, 5, and 6 above.

Regarding a systematic approach toward a technical specification of the structure of the administration shell, we refer to an upcoming article to be published in *atp edition*.

1.2 Industrie-4.0-Compliant Communication

To participate in Industrie-4.0-compliant communication, an component needs to

1 | Implement the information services (model access) defined in sections 4.3 or 4.4 and
2 | Integrate with the communication services (data transport)

using particular M2M protocols based on Internet Protocol (IP) technology (e.g. OPC UA).

2. CONCEPTUAL SERVICE ARCHITECTURE FOR INDUSTRIE 4.0

The RAMI4.0 and the I4.0 Component are the main guiding concepts of Industrie 4.0. The proposed Industrie 4.0 Service Architecture is therefore defined with reference to the main concepts.

2.1 Service Hierarchy

The Industrie 4.0 Service Architecture distinguishes between four types of services that are built on top of the actual automation and communication assets (see Figure 1). The *Communication Services* define the primitives for data transfers (connect/disconnect, transmit/receive, etc.) considering the negotiation of the required quality of service (QoS). The communications services themselves are defined in a technology independent way in order to allow mappings to M2M protocols such as OPC UA or DDS.

On top of the communication services there are services for information access – the *Information Services*. These services define the basic functions required to interact with information models (read/write, create/delete, etc.). This paper will focus on these services which will be detailed in section 4.2. It is important to keep in mind that the information service themselves

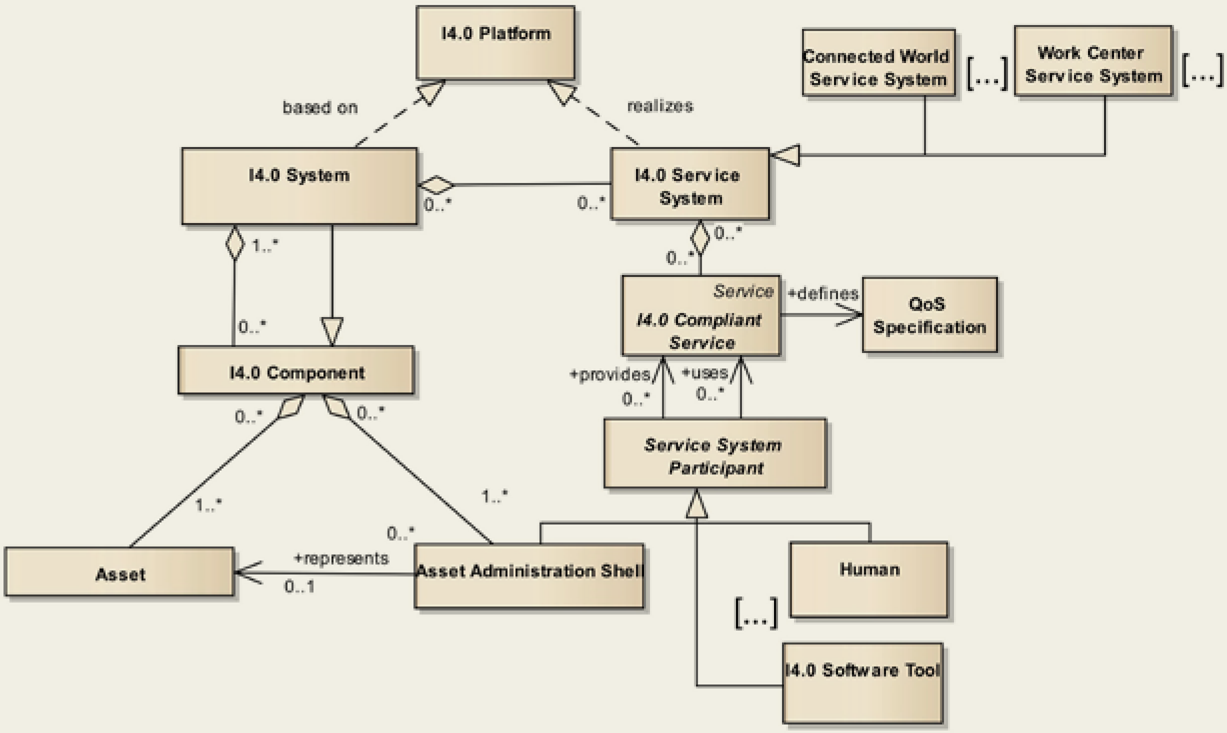

FIGURE 2: Overview model of the Industrie 4.0 Service System

do not define data semantics but only provide operations for interaction with "meaningless" data. These services are also specified in a technology-independent manner and must be mapped onto M2M protocols such as OPC UA or DDS.

Using these low-level services, higher-level services are built: *Platform Services* supporting the self-management of an I4.0 system and the *Application Services* exposing the actual production-related functionality (e.g. drilling, welding, material transport, etc.).

Whereas the platform services are designed to be domain-agnostic and provide generic functionality such as *discovery* of application service providers, the actual content of each application service is highly domain-specific and must be defined by the corresponding user- or vendor-organization..

Although the RAMI 4.0 layers are not intended as hierarchical "(N)-services" in the sense of ISO/IEC 7498-1, their functionalities can easily be mapped onto a service hierarchy as illustrated in Figure 1. As stated above, the resulting service system is technology-independent, and while it provides the *mechanisms* for defining the semantics of data, it does not actually define specific content and its semantics. Rather, a suitable information meta-model is proposed as a "modeling-language" so *parts of any RAMI 4.0 layer can be described by means on the information layer* (which is in fact its main purpose).

Note that all service layers come with their own search and discovery functionality. Particularly on communication and information layers, the basic discovery of network hosts and services (address and endpoint information) requires dedicated mechanisms outside of the Administration Shell, which at this point is not yet accessible. In fact, the purpose of these low-level discovery functions is exactly to provide the entry points into AS information models to further search for specific properties or skills based on data semantics. The discovery topic is further introduced in chapter 3.

2.2 Description of the Industrie 4.0 Service System

Previous publications on Industrie 4.0 either focused on the high-level reference architecture (RAMI 4.0) [1] or the description of particular parts of the Industrie 4.0 ecosystems such as the Industrie 4.0 component and the Administration Shell [2]. However, in order to build a service architecture a more concrete specification on how these things relate to each other is required. Thus, FIGURE 2 depicts a more formal overview model of the already defined artifacts of Industrie 4.0 and how they relate to each other. The elements described in this model are based on the official articles on Industrie 4.0 mentioned above as well as the glossary defined by GMA 7.21 that is extracted from these publications and agreed on in the consortium [10].

As depicted in Figure 2 the top level elements of the I4.0 service architecture are defined by the I4.0 Platform, I4.0 System(s) and the actual I4.0 Service System. An I4.0 Platform defines the basic and standardized communications and system infrastructure that enables the efficient creation of I4.0 Systems in a particular domain. Thus, actual I4.0 Systems are based on such a platform. The system itself is comprised of I4.0 Components which in turn can be systems themselves. An I4.0 component is defined as the combination of one or more assets together with their Administration Shell(s).

The I4.0 Service System now flanks this hierarchy of elements and brings in the service orientation side of the I4.0 System. There might be different notions of a service system depending on the scope to which services need to be exposed. For example, a site-internal service system would probably involve a different set of services than a global I4.0 Service System. An I4.0 Service System comprises of a set of I4.0 Compliant Services that have a defined Quality of Service (QoS) specification. These services are provided and/or used by I4.0 Service System Participants. The most prominent of these participants are naturally the services offered by the Administration Shell of the I4.0 Components that take part in the system. However, human actors or tools that are not I4.0 Components (e.g., for engineering systems or tools) can act as I4.0 Service System Participants by interacting with these services.

3. DISCOVERY SERVICES

An I4.0 System consists of I4.0 Components that can be distributed over networks or multiple I4.0 Components can coexist on the same system element (e.g. as independent services on a device or a cloud). In order setup a system and exchange information between the I4.0 Components, they have to determine the address of the targeted I4.0 Component. In I4.0 use-cases it is targeted to deploy and rearrange I4.0 Components flexibly in a system, i.e. efforts for manual configuration should be avoided, such as entering addresses. Discovery Services address this aspect. Their task can be defined by the following description: Discovery Services are used to exchange the basic information (announcement) that is required to setup the further information exchange between I4.0 Components – i.e. Discovery Services are used for bootstrapping the information exchange for the Information Services. The information required to setup further information exchange typically uses URLs. Such announcement typically carries an URL that refers to a protocol and a network endpoint, but it can also provide additional meta-data that can be used for filtering (e.g. device-type, vendor) or connection setup (e.g. protocol specifics, server-capabilities).

It ist possible to differentiate between Local and Global Discovery: Local Discovery can make use of mechanisms that are available within a local network (e.g. IP-broadcast, -multicast). Global Discovery is typically required when no such mechanism can be applied – e.g. in multiple networks with barriers that filter out IP-broadcast or -multicast. Global Discovery typically makes use of a registrar with a common well known address that can be pre-entered in an I4.0 Component.

Multiple technologies exist that can be used to implement such Discovery Services: For Local Discovery mDNS (Multicast DNS also known as part of zero-conf or Bonjour) is often used [11]. The basic concept of mDNS follows DNS using UDP multicast, which provides a peer-to-peer based exchange of URLs combined with additional field that can be used to transport the aforementioned meta-data. OPC UA provides a Local and Global Discovery Server. The basic OPC UA Local Discovery Server (LDS) follows the concept of a local registrar without the ability to announce that information in a network. However, the new specification of LDS-ME (Multicast Extension) extends LDS by means of mDNS. The OPC UA Global Discovery Server (GDS) could be used as technology for Global Discovery – however it might not be suited for all use cases (e.g. for Cloud Integration) as it is often difficult to integrate the required OPC UA based protocol with the network infrastructure (e.g. Firewalls, HTTP/REST based frameworks).

4. INFORMATION SERVICES

Information services provide basic mechanisms for data access and manipulation. To be stable under the evolution and replacement of middleware technologies, and to allow interoperability between different middleware technologies, information services are defined in a technology-independent manner. They can be considered as the conceptual, technology-independent interface of the information layer, which is then mapped onto technology-specific protocol(s) such as OPC UA. Applications, i.e. the elements of the Industrie 4.0 functional layer, can be defined against a kind of technology abstraction layer (this is a core feature of the information layer).

4.1 Industrie 4.0 Information Meta-Model for Interoperability

One of the major efforts of Industrie 4.0 is to ensure interoperability between its systems. Thus, the service architecture is defined on a level which allows the integration of its underlying technical systems in a flexible and sustainable manner. This is achieved by defining a technology-independent vocabulary for *Information Services* proposed in section 2.1. On a formal level, this vocabulary forms a common information meta-model.

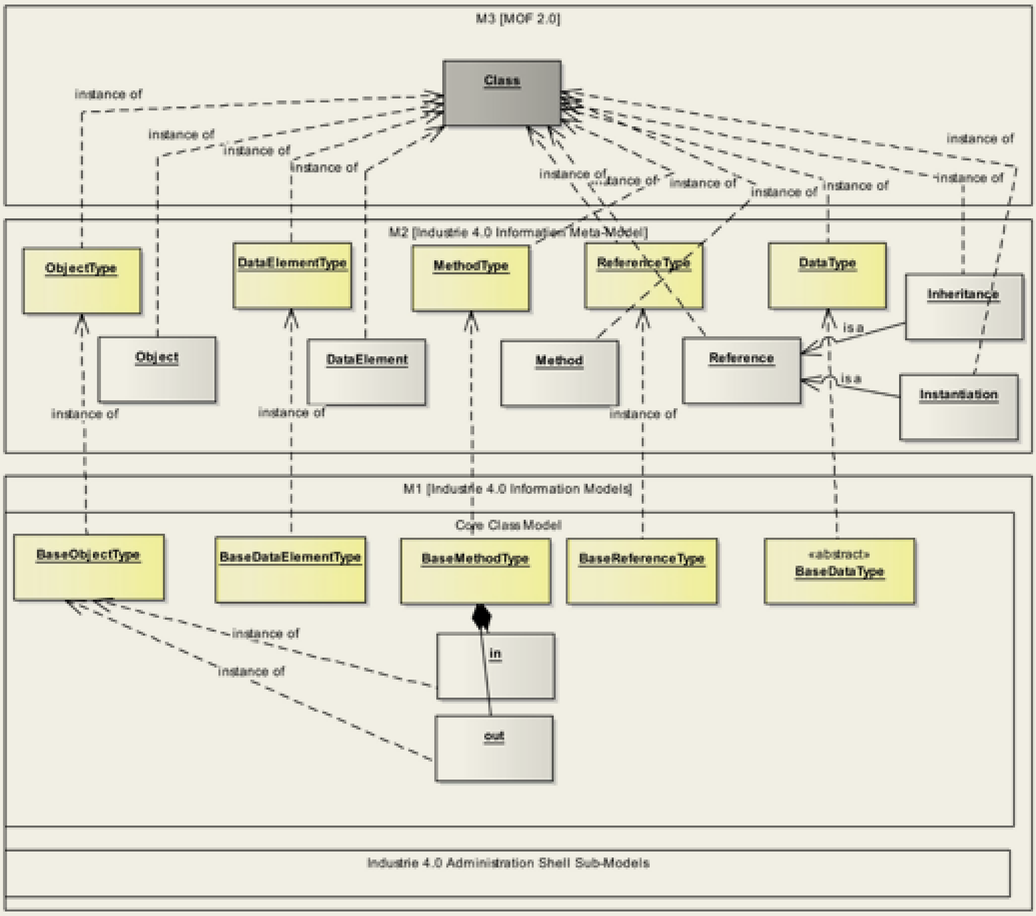

FIGURE 3: Required elements of an Industrie 4.0 Information Meta-Model

It is defined on an abstract level so that it provides sustainability under evolution, basically aiming at the "extinction" of specific technologies. Still, the meta-model defines the core concepts for information modelling that can be mapped to underlying technology-specific meta-models such as the one of OPC UA.

Object-orientation has proven to be a useful concept in order to describe information on an abstract level. From abstract modelling languages such as the UML to domain specific information modelling languages such as OPC UA or IEC 61850 similar concepts are used to define and structure information models. These conceptual models are mostly based on the same paradigms such as objects (classes, elements, etc.) that have references to each other (associations, links, etc.) and provide data elements (variables, attributes, fields, etc.). On a more general level, these are the concepts we find in the state of the art of meta-modelling with the Resource Description Framework (RFD), the Object Management Group (OMG) Meta-Object Facility (MOF), and specifically with the UML2 core model [12].

In addition to the operational use of information models as artifacts of service interaction, the specification and implementation of models in standards and products are essential use-cases that must be supported. To this end, the vocabulary given by the meta-model must be easy to use if it is to serve as a modeling language that is usable in practice. *Easy to use* in this context means *minimal* (avoiding uncertainties about which entity to choose), *concise* (giving any entity a perfectly clear meaning), and *complete* (ensuring all needed information content can be expressed). In other words: vocabulary or modelling language must be *easy to use for an application-domain expert* wishing to state information about a domain-subject (e.g. a robot), not an M2M protocol expert or software architect. To this end, the use of simplified tabular formats has proved to be very successful in the past.

TABLE 1: Signature of information services

Signature (o = optional)	Description
Source Addressing	**Who** is soliciting the access?
Operation	*What* is being done to the entity at the given **address**, using the specified **information** payload
Target Addressing	Identifying (and eventually addressing) the target of the service operation, i.e. attributes to set, parent objects to receive a new child object
Information Payload	The information to be transferred, i.e. attribute values to assign, object types to instantiate
RSVP Flag (o)	Indicate whether a response/acknowledge on the information layer is expected. This is unrelated to the implementation of the transport layer (layer 4), where UDP (connection-less) or TCP (connection-oriented) might be used.
Context (implicit)	The context in which the service is called, i.e. the application, user, the local I4.0 network, a global (discovery) scope, etc.
Context: QoS (o)	The **expectation** on the reliability and timeliness that the communication layer needs to provide for this service call.

While the specific industry formats given by IEC 61360 [13] or IEC 61987 [14] are *concise*, they are *not minimal* in the sense that a domain-expert faces a large number of optional meta-attributes to supply for any object or variable, and being tailored for order-related properties, the vocabulary is *not complete* e.g. with regard to generic reference mechanisms within the data model. Some of these issues could be remedied with proper tool support, but in our view it shows that the mentioned formats were intended for data exchange, not for a dynamic, sustainable modeling *process* performed by humans. However, the use of a tabular format is appreciated for its efficiency.

UML as the state of the art from the software domain is certainly *complete* with regard to object-orientation as at the beginning of this section, without further restriction it is *not concise* and *not minimal enough*. One could model a device type as a class, the device instance as an object from that class. However, UML complements the tabular formats of IEC 61360 and IEC 61987 with a graphical notation. In our experience, the practical use of tabular and graphical notations is about 9:1. Both have their value, but while graphical notations are good to *visualize* structural relationships, the original *creation* of that content is better done using plain tables.

As an example of M2M protocols in the automation domain, OPC UA is *complete* and - as an implemented protocol – it also is *concise*. However, already the basic type-space (namespace 0) of an otherwise empty server is *not minimal*, offering a huge variety of object and reference types which would confuse a domain-expert. Furthermore, OPC UA information models are by nature protocol-specific.

In summary, we need a modelling language that is

- Easy to use for a domain-expert as part of a *model creation process* (meaning it has a minimal, concise, and complete vocabulary).
- Independent of specific M2M protocols, but can be mapped to them for *operational/online model exchange* (meaning we can describe a mapping of the Industrie 4.0 information meta-model to the M2M protocol meta-model). Here, information is made "serviceable" through resource- or message-oriented services.
- Supported by machine-readable entry and serialization formats (table-based, XML, JSON, etc.).

4.2 Industrie 4.0 Information Meta-Model – Vocabulary for Information Modelling

Figure 3 outlines the key vocabulary we require to describe information models. Both a formally complete specification and a "short grammar" for model engineers are needed. In fact, the purpose of this section is to state concise requirements for this "short grammar" rather than proposing it as formally complete specification. To arrive at this specification, the exact overlaps and gaps with existing specifications like UML need to be identified.

From a model engineer's perspective, the following elements along with their respective types make up the *Industrie 4.0 information meta-model*:

- Objects: main building blocks of information models representing conceptual entities.
- Data Elements: holding and referring static as well as dynamic data values.
- Data Types: describing the possible value sets for data elements.

TABLE 2: Essential service operations

Operation	Description	Target Entity[1]	Response
Get / Set	Get or set values of variables, topics, or alarm objects (defined through the target address). In effect read/write, publish/subscribe, and general event-based communication can be realized by calling get/set operations on a data element address, a publication topic, or an event object.	Data elements	Value (mandatory for read), confirmation (optional for write)
Create / Delete	Insert or remove object instances into the information model (including the creation of references to structure the model)	Objects (and References)	Optional
Browse	Traverse information model	References	Object(s)
Method Call	Transport a complex payload whose contents are interpreted by business-logic, i.e. unlike a read or write operation, the effects are method-specific.	Methods	Optional, depends on semantics of the method call

TABLE 3: Examples for contexts of service calls in Industrie 4.0 systems

Context Facet	Example
Application	Augmented reality application during device replacement
Users	Service engineer
Industrie 4.0 Access Network	Local I4.0 Network #123
Discovery Scope	Plant-local
System State	Operation vs. maintenance phase of the process

- References: Common rules for expressing semantics and references between distributed models. Additionally, together with objects, references can be used for structuring the information models.
- Methods: Representing complex functionality that can be triggered in the context of individual objects in the information model. Methods may receive and return data elements (parameters, return values) without changing the state of information model.

The concept of objects with references can be used to describe extended use cases present in object oriented modelling/programming:

- Instantiation (type-instance relationship between ObjectTypes and Objects): By using a special “is-type-of” reference it is possible to distinguish between object instances and types in an information model.
- Inheritance: Is realized by using a specialized reference that also have this fixed semantics. This also allows to build extended concepts like polymorphism on instance level.
- Containment and Aggregate Structures: Again, by defining specific references between entities, aggregation semantics can be created.

To make the types available in the modeling process and support reflection mechanisms, they could be mirrored into to model layer as a kind of core class library. This corresponds to the approach OPC UA takes in namespace 0. The concrete Industrie 4.0 administration shell sub-models can then be defined using these base classes.

Main points for discussion still are the exact selection of *data types*, treatment of *enumerations*, creation of *data sets*, or definition of *access patterns* (cyclic, on-demand, event-based).

4.3 Service Signature for Resource-Oriented Information Access

The service calls have a common signature as described in Table 1. For *resource-oriented access*, all information entities are individually addressable and accessible, thus spanning up the address space of the information model. The *functionality* of each service described by the data operation, the target address, and the accessed information content. In addition, there are *non-functional* aspects like the service quality, which mostly define the expectations on the underlying communication layer [7].

TABLE 4: Schema for tag values of data elements

<table>
<tr><td rowspan="3">“tag:”</td><td colspan="3">taggingEntity</td><td rowspan="3">“:”</td><td rowspan="3">specific</td><td rowspan="3">[“#” fragment]</td></tr>
<tr><td>authorityName</td><td rowspan="2">“,”</td><td>date</td></tr>
<tr><td>DNSname | emailAddress</td><td>Year [„-“ month [„-“ day]]</td></tr>
</table>

TABLE 5: Example for multi-language semantic references

```
variable.is_tagged_by[0] = “tag:iso.org,2017:iec61987:0112/2///61987#ABB215”
variable.is_tagged_by[1] = “tag:eclass.org,2017:0173-1#02-BAE430”
```

TABLE 6: Examples of local and global references to data points

Generalized address	Technology-specific example
Fully qualified dns host name + service end-point + local data path or identifier	opc.tcp://dms.local/FDI/Objects/DeviceSet/TTH300/ParameterSet/PV

The specific operations are explained in Table 2 The given information payload already indicates how each operation is responsible for one specific type of entity in the information meta-model described in Section 4.2. On a technology-independent level publish/subscribe may be considered as a mere read/write operation on topic identifiers. However, we expect this type of data distribution to become very relevant in distributed, self-configuring Industrie 4.0 systems and therefore propose these dedicated (convenience) operations. The matter of entity addresses in a distributed system is subject of Section 4.4.

Instead of defining different operations for unconfirmed and confirmed (acknowledged) types of communication, we propose an RSVP flag to indicate if the service should be executed with or without a confirmation. As stated in Table 2, some operations such as *read* on data elements naturally require a response to be sent.

As stated above, the expected service quality (QoS) is a requirement on the Industrie 4.0 communication layer. For our considerations, we presume that this layer offers a *transmit* and a receive service, which are configured using the *target address* and *QoS* parameters from the information service call. A first proposal of a reference architecture for the communication layer comes from the *Platform Industrie 4.0 AG 1, UAG (sub-working-group) Network-Based Communication* [7]. QoS is subject to the QoS working-group in GMA FA 7.21. Overall, we expect typical quality indicators to be supported such as:

- Availability and reliability. In practice, a redundancy factor of the underlying infrastructure might be much more usable. Still, it must be considered that the asset (data source) itself may not be redundant to begin with.
- Timeliness (latency, cycle- or round-trip-time and related jitter). From the perspective of the functional layer, latency-related properties would need to include processing time in the end-points. It must be considered that e.g. process instruments may only support measurement cycles far lower then a network is able to deliver.
- Update frequency, maximum age of information
- Required data-rate/bandwidth (per cycle)

In any case, the deployed physical communication resources limit what can be configured in software, availability and latency considerations must be taken into account already at network design time. It makes great sense to be able to query the communication layer for the maximum configurable quality of service properties.

For security, it is important to distinguish between

- Authentication and authorization, i.e. confirming the (claimed) identity of the service provider/consumer and enforcing access rights accordingly. In the service hierarchy, *authentication* is a good example for a higher-level *platform service*.
- Confidentiality and integrity, i.e. ensuring that information is never disclosed to any third party and that manipulations by third parties can always be detected (if not avoided). Data confidentiality an example for a function of the *communication layer*.

Defining the details of secure identities and confidential communication, in particular in a long-term sustainable manner, is subject to *Platform Industrie 4.0 Working-Group 3, Security* [8].

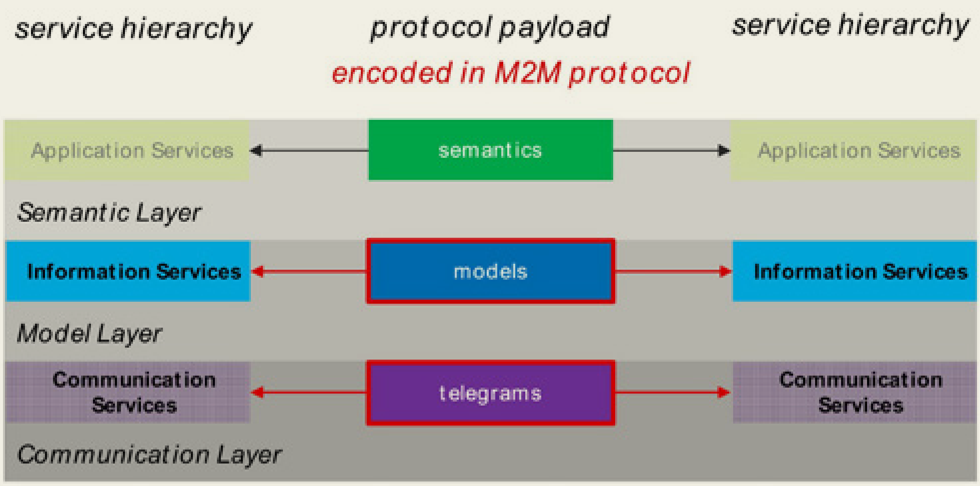

FIGURE 4: Industrie 4.0 service hierarchy and payloads of the interaction protocols

Any service call invariably takes place in a determined context. Following the examples in Table 3, this context is formed by the environment of the service participant. In particular the Industrie 4.0 Network as a resource and security policy context have a strong impact on discoverability and accessibility of distributed information models. Standardizing the view on this service context is a matter of security and convenience for service orchestration in Industrie 4.0.

In conclusion, quality of service, authentication and communication-layer security, along with the service context need further discussion. This requires close collaboration between the joint WG1/3 subgroup "secure communication" in Industrie 4.0 the GMA FA 7.20 and 7.21 expert committees.

4.4 Message-Oriented Access

Message-oriented services bind a specific functionality and its data context tightly together [15]. As discussed in the GMA FA 7.20, a particular *verb* representing a machine skill (e.g. "drill") is combined with properties of that skill (e.g. "diameter" of the hole to be drilled, drill "speed", etc.). This allows the drilling skill either to be executed with all given parameters considered – or not at all. Resource-oriented access always faces the challenge of leaving a parameter model in an inconsistent state. In other words, *transaction safety* is missing in the resource-oriented parts of M2M protocols such as OPC UA and has to be additionally specified and implemented. Therefore, for the atomic negotiation of machine skills/resources, message-oriented information access is perfectly suited.

We propose therefore to support both message- and resource-oriented data access (one might also say *random access*) but to organize the messages within the resource-oriented information models to define their execution context (which aspect of which asset) implicitly.

4.5 Addressing, Identification, and Semantics

As previously stated, an integral part of the service architecture are common rules for addressing, identification (see section 4.3), and understanding of the data exposed (see definition of interoperability in section 1.1) in the information model (see section 4.2). The information model needs to reflect the *where*, *who*, and *what* of its elements, so to speak.

It is important to distinguish between address and identifier of information. Addresses may be *transient*, but describe the location of information at a particular point in time; they enable seamless data *access* in a globally distributed system. Identifiers are *immutable* and always refer to the same object of interest. In particular, a fragment of an AS does not have to be directly aware of its own address but only its identity. Providing access (via addresses) to the AS of a particular asset (via identifiers) is then the task of the service system. For Industrie 4.0 components to interoperate, the meaning of data must also be represented in the information model in a machine-readable form.

In summary, addresses, identifiers, and semantics are essential meta-data of any piece of information in the AS. They can be realized using elements of the Industrie 4.0 meta-model in the following manner:

- addresses, implemented via local and global references
- identifiers, implemented via designated identifier data elements

- semantics, implemented either by semantic identifiers or semantic references to an object whose meaning might be known (or which provide further semantic references that have to be resolved)

The Industrie 4.0 service system must support the resolution of (semantic) identifiers to addresses.

To be able to integrate semantic definitions from existing standards, a kind of "multilingualism" is needed for the same data element. A data element must be able to hold multiple references/ identifiers (i.e. multiple "languages"), and each reference/identifier must state the namespace in which an address or identifier value is to be interpreted. It seems to be no viable option to base the entire definition of semantics in Industrie 4.0 only on object names or type systems. Table 4 specifies the creation of tag-values according to RFC 4151 [15] the generic URI specification in RFC 3986 [17]).

Table 5 illustrates this for the lower-range measuring-limit of a temperature transmitter. The example also shows that existing semantic dictionaries such as IEC 61987 or may eCl@ass may differ for individual physical quantities (the upper range measurement-limit

REFERENCES

[1] ZVEI - Zentralverband Elektrotechnik- und Elektronikindustrie e. V. (ZVEI). (2015). *Das ReferenzarchitekturmodEll Industrie 4.0 (RAMI 4.0)*. Retrieved from: https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2015/april/Das_Referenzarchitekturmodell_Industrie_4.0__RAMI_4.0_/Faktenblatt-Industrie4_0-RAMI-4_0.pdf

[2] VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik. (2016). Status Report: Industrie 4.0 Service Architecture. Retrieved from: https://www.vdi.de/fileadmin/vdi_de/redakteur_dateien/gma_dateien/END_3_Status_Report_Industrie_4_0_-Service_Architecture.pdf.

[3] Bundesministerium für Wirtschaft und Energie (BMWi). (2016).Structure of the Administration Shell – Continuation of the Development of the Reference Model for the Industrie 4.0 Component. Retrieved from: https://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/structure-of-the-administration-shell.pdf.

[4] ISO/IEC 7498-1:1994, Information technology – Open Systems Interconnection – Basic Reference Model: The basic model

[5] DIN SPEC 16593:2017-04, RM-SA - Reference Model for Industrie 4.0 Service architectures - Basic concepts of an interaction-based architecture

[6] DIN EN 61804-2:2007-10, Function blocks (FB) for process control; Part 2: Specification of FB concept

[7] Bundesministerium für Wirtschaft und Energie (BMWi). (2016). *Network-based Communication for Industrie 4.0 – Proposal for an Administration Shell*. Retrieved from: https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/network-based-communication-for-i40.pdf?__blob=publicationFile&v=7

[8] Bundesministerium für Wirtschaft und Energie (BMWi). (2016). *Sichere unternehmensübergreifende Kommunikation*. Retreived from: https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/sichere-unternehmensuebergreifende-kommunikation.pdf?__blob=publicationFile&v=10

[9] Schulz, D., & Schmitt, J. (2017) *Industrie-4.0-konforme Kommunikation*. Lecture presented at Kommunikation in der Automatisierung (KommA), Magdeburg.

[10] VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik, Fachausschuß 7.21 „Industrie 4.0" (2015). Glossar Industrie 4.0 des Fachausschuss VDI/VDE-GMA 7.21 „Industrie 4.0". Retrieved from: https://www.iosb.fraunhofer.de/servlet/is/48960/Begriffsdefinitionen_VDI_GMA_FA7-21_v4.pdf?command=downloadContent&filename=Begriffsdefinitionen_VDI_GMA_FA7-21_v4.pdf

[11] Cheshire, S., & Krochmal, M. (2013). *Multicast dns* (No. RFC 6762).

[12] Object Management Group. (2016). OMG Meta Object Facility (MOF) Core Specification. Retrieved from: http://www.omg.org/spec/MOF/2.5.1/PDF.

[13] IEC 61360-1:2017, Standard data elements types with associated classification scheme for electric items - Part 1: Definitions - Principles and methods

[14] IEC 61987, Industrial-process measurement and control - Data structures and elements in process equipment catalogues

[15] Bundesministerium für Wirtschaft und Energie, (BMWi). (2016) *Weiterentwicklung des Interaktionsmodells für Industrie 4.0-Komponenten*. Retrieved from: https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/interaktionsmodell-i40-komponenten-it-gipfel.pdf?__blob=publicationFile&v=13

[16] Kindberg, T., & Hawke, S. (2005). *The'tag'URI Scheme*. Retrieved from: https://www.ietf.org/rfc/rfc4151.txt

[17] Masinter, L., Berners-Lee, T., & Fielding, R. T. (2005). *Uniform resource identifier (URI): Generic syntax*. Retrieved from: https://www.ietf.org/rfc/rfc3986.txt

[18] Grangel-González, I., Halilaj, L., Coskun, G., Auer, S., Collarana, D., & Hoffmeister, M. (2016, February). Towards a semantic administrative shell for industry 4.0 components. In *Semantic Computing (ICSC), 2016 IEEE Tenth International Conference on* (pp. 230-237). IEEE.

[19] Goldschmidt, T., & Schulz, D. (2016). Industrie 4.0 Dienstearchitektur – Semantische Interoperabilität in Industrie 4.0 Dienstesystemen. In: Tagungsband Automation 2016 (pp. 107 – 108). Düsseldorf: VDI Verlag.

of a Coriolis mass-flow meter in eCl@ass is e.g. 02-AAQ480).

What is pragmatically required in the next steps is an agreement on *common practice address formats* acknowledging the use of Internet technology. DNS names or IPv6 addresses are good candidates for addressing service hosts. To address data within a host, local names or variable identifiers can be used. OPC UA already uses such a concept as illustrated in Table 6.

The presented approach allows objects (e.g. "measurement skill"), data elements (e.g. "measurement range limit"), or even methods (e.g. "calibrate") to be semantically annotated with references to multiple (existing) dictionaries based on one common mechanism. Existing semantics can be re-used, new definitions can be created in user- or vendor-organizations without any governance issues. This should greatly facilitate the introduction of machine-readable semantics.

SUMMARY AND OUTLOOK

We have presented the state of the discussion on the Industrie 4.0 Service Architecture. A service hierarchy is defined in compliance with ISO/IEC 7498-1 and DIN SPEC 16593. An information-meta model is outlined toward a description language for information models, which in turn can be serialized for interaction through M2M protocols as illustrated in Figure 4. This information meta-model and addressing scheme can serve as starting point for further work. The exact vocabulary and a usable entry format for model engineers are needed; furthermore, there are detail modelling challenges (e.g. treatment of enumerations) that must be resolved before any such format can be practically used. At runtime, both resource- and message-oriented information access must be supported so information content has to be defined only once but can be access in both manners. To this end, mappings to (existing) M2M protocols (e.g. OPC UA or OneM2M) and offline serialization formats (AutomationML or other XML dialects) must be defined.

Furthermore, a generalized approach for basic service discovery based on Internet technology is given. This approach must be validated and extended toward global search and discovery mechanisms that also cover the *connected world* of RAMI 4.0.

In conclusion, the topics of *discovery, specification, and access to distributed information models* must be completed as part of defining a technical specification of an Industrie 4.0 Administration Shell. To this end, the ongoing work within in the GMA FA 7.20 and 7.21 expert committees must be consolidated and integrated with the "Administration Shell in Detail" activity lead by Plattform Industrie 4.0 AG1.

MANUSKRIPTEINGANG
06.10.2016
Im Peer-Review-Verfahren begutachtet

AUTHORS

Dr. rer. nat. **DIRK SCHULZ** (geb. 1976) is working as Principal Scientist at ABB Corporate Research Center, Germany. His main focus is the Connectivity Architecture for digital automation systems in the context of Industrie 4.0 and IIoT. This includes technologies related to the communication and information layers such as SDN, 5G, and OPC UA. Dirk Schulz is leading the working-group on Industrie 4.0 Service Architecture in the GMA FA 7.21 expert committee. He has a background in computer engineering and holds a PhD from the University of Mannheim.

ABB AG Forschungszentrum,
Wallstadter Straße 59, 68526 Ladenburg,
Tel. +49 (0) 6203 71 62 67,
E-Mail: dirk.schulz@de.abb.com

Dr.-Ing. **THOMAS GOLDSCHMIDT** (geb. 1982) works as a Platform Architect at ABB Digital, Germany. The main focus of his work is in industrial internet of things, information modelling as well as cloud architectures. He studied computer science at the Hochschule Furtwangen University and received his PhD from KIT.

Dr.-Ing. **JOHANNES SCHMITT** (geb. 1979) Schmitt is senior scientist in the group Devices and Networks at ABB Corporate Research. His research focusses communication and especially information centric concepts for automation systems. He also investigates the usage of OPC UA on embedded or mobile devices up to cloud based systems. Johannes holds a PhD in Electrical engineering from the TU Darmstadt, Germany.