An Ontological View of the RAMI4.0 Asset Administration Shell

Andreas W. Müller[1], Irlán Grangel-González[2] and Christoph Lange[3]
[1]*Data & Analytics Governance, Schaeffler AG, Herzogenaurach, Germany*
[2]*Corporate Research, Robert Bosch GmbH, Renningen, Germany*
[3]*Fraunhofer FIT, Sankt Augustin, Germany*

Keywords: Industry 4.0, Asset Administration Shell, Information Integration, Industrial Ontology.

Abstract: Information interoperability is of paramount importance to Industry 4.0 (I4.0). To this end, the Reference Architecture Model for I4.0 (RAMI4.0) defines the Asset Administration Shell (AS) concept as a core element for interoperable descriptions of assets. Assets, such as Cyber-Physical Systems, are building blocks that need to be interchangeable between various industrial systems but are heterogeneously modeled. In order to achieve the required interoperability between these systems, semantics is key. Typically, two types of systems occur in I4.0 scenarios: legacy systems not considering explicit semantics, and a new generation of ontology-based systems. However, existing approaches for modeling the AS do not explicitly address this situation of interoperability between systems of such different types. In this article, we develop an ontological view of the AS concept to bridge the gap between these types of systems. This results in an ontology that is intended to be likewise realizable with both ontology-based and non-ontology-based systems. We present this ontology together with a suggested two-phase engineering process for its application.

1 INTRODUCTION

The digitization of industry requires integrated information models describing the assets and information sources of companies to enable semantic integration and interoperable data exchange. The Industry 4.0 vision (Lee et al., 2015) aims at creating *Smart Factories* by combining Internet of Things (IoT), Internet of Services (IoS) and Cyber-Physical Systems (CPS) technology. Smart Factories promise to improve the production process by adding intelligence to industry data, such as orders or sensor data, and by coordinating the way humans and machines collaborate. Different standards, such as those published by bodies such as ISO or IEC, are used to describe information about manufacturing, security, identification and communication, among others.

Although the vision of digitizing production and manufacturing has gained much traction lately, it is still not clear how this vision can actually be *implemented* in an interoperable way using concrete standards and technologies (Brettel et al., 2014). A key challenge is to enable smart industrial devices to communicate and to *understand* each other as a prerequisite for cooperation scenarios (Kharlamov et al., 2016). Of paramount importance for achieving interoperability in this context is the RAMI model along with its Asset Administration Shell (AS) concept (Marseu et al., 2017). The AS is intended as a defined access point to the variety of information and functionality offered by smart devices. The contribution of this article is to develop an ontological formalization of the AS concept that supports realization with and interoperability of different types of information systems. To foster acceptance and implementation of the AS concept, we consider it essential to design the AS in a way that can likewise be integrated with existing systems and devices as well as with future ontology-based systems. That way, implementers can homogeneously extend their portfolios without information breaches.

The paper is organized as follows. Section 2 provides a definition of the problem. Section 3 gives necessary background information. Section 4 describes related work in this field. Section 5 introduces our approach, while section 6 concludes the paper.

2 PROBLEM DEFINITION

Assets are viewed as industrial devices, ranging from simple components to complex CPS. They exhibit

Müller, A., Grangel-González, I. and Lange, C.
An Ontological View of the RAMI4.0 Asset Administration Shell.
DOI: 10.5220/0010111101650172
In *Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2020) - Volume 2: KEOD*, pages 165-172
ISBN: 978-989-758-474-9
Copyright © 2020 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

(a) RAMI4.0. (b) Administration Shell (AS).

Figure 1: **The RAMI4.0 model and the Administration Shell.** RAMI4.0 includes a three-dimensional model with core processes and concepts for I4.0, e.g., the Administration Shell (AS) (Adolphs, et al., 2015); The manifest, the component manager, and the Asset are included as main parts of the AS.

various capabilities, e.g., communication or execution of production steps, which may change during the individual lifecycles. Hence, they may be equipped with AS depending on the respective capability set.

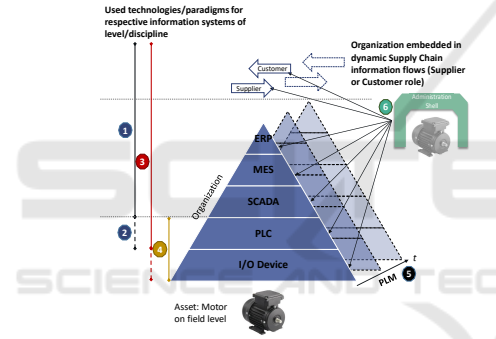

Figure 2: **Role of the AS in manufacturing processes.** The AS plays a key role in the Manufacturing process on all levels of the automation pyramid, while the associated asset can be located on the field level. The automation pyramid denotes a main principle of organization of systems in the industrial domain. The AS is key in the Product Life-cycle Management (PLM) process as well as in Supply Chains. For vertical and horizontal use it needs to be implemented differently due to the predominantly used technologies and paradigms on the respective layers. The numbers denote (1) Semantic technologies; (2) Extensions of semantics towards the field-level, e.g., via OPC UA; (3) OO (-based implementations); (4) Non-ontology, non-OO (-based implementations); (5) PLM-supported changes in the organization; (6) Arrows indicate the relevance of the AS model in various contexts. The dotted lines for Organization and Customer/Supplier indicate respective variability over time.

The AS provides methods for interaction with an asset and capabilities that are required for its use. Hence, an AS depicts a *smart interface* to an asset, which may vary in its asset-specific extent but always provides a standard access point for knowledge discovery and asset utilization.

Further, the information conveyed by the AS needs to be horizontally and vertically used across all layers of the automation pyramid (cf. Figure 2). Since the systems on the different levels could be legacy or ontology-based, AS manifestations also need to be either object-oriented or ontology-based, respectively. Hence, the AS structures need to be seamlessly applicable to all layers and thus a common understanding of information structures is required in this setting.

Considering the dynamics of assets with their capabilities and along their lifecycles, an AS needs to be understood by all actors involved in handling an asset (cf. Figure 2). Following this, and considering the scenarios of AS usage (ZVEI, 2016) as well as our own experience in the field, we propose an AS realization to meet two basic requirements: 1) The AS requires an ontology that permits modeling with both ontology-based and non-ontology-based systems alike; and 2) The ontology shall permit separation of the AS information from the description of the asset, enabling independent refinement of the models of AS and assets.

3 BACKGROUND

RAMI4.0 describes basic aspects of Industry 4.0 (Adolphs, et al., 2015) (see Figure 1a). The vertical axis (left) represents the IT perspective, with layers ranging from the physical device (*asset*) to complex functions in ERP systems (*functional*). These layers correspond to the idea of decomposing complex projects into smaller manageable parts. The horizontal axis on the left indicates the product lifecycle with *Type* and *Instance* as the main concepts.

In RAMI4.0, the AS concept is pivotal for achieving the desired interoperability between assets. An

asset is a physical or logical object that is managed by an organization and has value for it (Adolphs, et al., 2015; DIN, 2016). Further, an I4.0 component can be a production system, a machine, or an assembly inside a machine. This comprises two basic elements: an asset and its AS. Every asset that is equipped with an AS becomes an I4.0 component. The AS should be able to represent the different kinds of information about an asset that occur during its lifecycle. Further, the AS contains the integral parts *manifest*, i.e. a catalog of the meta-information about the AS, and *component manager*, which serves as a basis for SOA-based access to the I4.0-component (cf. Figure 1b).

The information systems used by the target audiences of RAMI4.0 are typically long-lived, heterogeneous and follow organizational principles such as the automation systems pyramid or are integrated in process networks such as supply chain management. For these audiences ontology-based approaches to RAMI4.0 constitute disruptions in existing application architectures. Hence, with regard to the long-term focus of RAMI4.0 and the endeavours required to realize a universal AS model, we provide an implementation-agnostic basic structural design that can meet current system architectures and can thus be applied to various types of interoperating information systems based on different technological foundations.

4 RELATED WORK

Grangel-González et al. have implemented the AS concept in the context of the RAMI as an ontology (Grangel-González et al., 2016). The ontology considers the AS in relation with the asset according to the first specification of the RAMI model (Adolphs, et al., 2015). Further, classifications of data are introduced as classes, e.g., application data, structural data are related to the asset to represent distinct types of data to which an asset should be related to. Core concepts of the AS specification, e.g., the manifest, header, and body were not considered in this work. Tantik et al. (Tantik and Anderl, 2017) proposed an integrated data model and structure for the AS in I4.0 contexts. Pethig et al. (Pethig et al., 2017) developed a data model for the AS to be applied in conditioning monitoring services for I4.0. Diedrich et al. (Diedrich et al., 2017) proposed a model for semantic interoperability for communication of assets within the smart factories context. In (Platform Industrie 4.0, 2019) a broad and detailed overview of AS representation, but without formal logical foundations, is given.

As shown above, most studies examined the semantic modeling of the AS concept. So far, however, there has been little discussion regarding the AS as a smart interface capable to deal with ontology-based systems.

5 METHODOLOGY

This section presents the steps and design choices taken to develop an ontology of the AS that meets the requirements given in Section 2.

5.1 General Approach

In creating our design we followed 1) the specification of the RAMI4.0 (DIN, 2016) as the normative guideline, and 2) an approach driven by initial object-oriented analysis and design (OOAD) (Booch et al., 2008), inspired by a similar conceptual approach (Negri et al., 2017), instead of mainly ontology-oriented approaches, such as competency question-driven authoring (Ren et al., 2014). This we consider essential for acceptance in the industrial community to foster integration into today's predominantly OO-based industrial system landscapes as well as to provide a bridge to ontology-based systems in the field.

In this sense, we first identified class candidates in the textual descriptions and definitions of the RAMI4.0 specification from nouns, and candidates for interrelations and attributes from verbs and adjectives, respectively. Generalizations, associations and parthood roles were constructed from textual indicators such as *is a*, *of*, or *consists of*, respectively. Following (Bradner, 1997), cardinalities were derived from textual qualifiers such as *must*, *can*, or similar for distinguishing mandatory from optional parts or interrelations. From this, an initial view, corresponding to an analytical model in OOAD, for the AS was derived, i.e., the basic set of terms, classes and their configuration was identified. In view of the suggested AS usage scenarios in (ZVEI, 2016), we refined and extended this model, and cross-checked with different descriptions of the AS in further non-normative publications. This refined model corresponds to a design model in the OOAD sense.

We then provided semantic grounding for the ontology following (Berardi et al., 2005) for seamless use with ontology-based systems as well as OO systems. During the entire course, best practices for ontology building and naming guidelines for entities were followed with regard to long-term usability of the ontology.

5.2 Terms and Interrelations

With the basic idea of the AS as an interface, two separate initial terms/classes can be identified, i.e., AS and Asset. The AS represents the Asset within an *I4.0 system* (third class). It is assumed that these classes denote different, individually manageable entities. With the notion of the presence of an associated AS enabling an Asset to become an *I4.0 component* (fourth class), which may be part of an I4.0 system, these four classes constitute the core of the target ontology (cf. Figure 3).

Figure 3: Initial analytical view of the four core terms of the AS model with their basic associations.

From the lifecycle and utilization perspectives both Assets and ASs may exist independently in terms of management by information systems. The membership of an Asset in the class of I4.0 components depends on its classification according to the communication capabilities and publicity level (CP) matrix (Adolphs et al., 2016). An Asset does not have to meet the requirements of an I4.0 component from the very beginning. Instead, it may be later equipped with properties or capabilities that permit its CP reclassification as I4.0 component, which only then requires an AS to be present. Additionally, I4.0 systems may contain Assets that are not I4.0 components. Hence, information systems must be able to identify and manage such non-I4.0 components as well. Using this initial analytical model, the requirements for the AS (cf. (Adolphs et al., 2016)), together with its scenarios for utilization described by (ZVEI, 2016), are now examined. This results in the following observations and considerations. From these, competency questions (CQ) are derived for the basic tasks of subsequent validation and model checking of the target ontology and thus for AS handling.

- Both Asset and AS are separately identifiable. **CQ1**: Which AS refer to an associated Asset?
- The AS consists of a *Header* and a *Body*, with the Body containing information about the I4.0-component, while the Header contains information about its utilization. Hence, both Header and Body are viewed mandatory. **CQ2**: Does an AS contain both Header and Body?
- The AS contains the *Component Manager*. There is a looseness in the specification: While the Component Manager is mostly described as a mandatory singleton, there are examples that imply that in situations no dedicated Component Manager needs to be defined as part of an AS. Instead, an AS could act as a proxy to the AS containing the Component Manager to be used. **CQ3**: Does the AS contain the Manifest? **CQ4**: Which ASs contain a Component Manager?
- An I4.0 component exposes domain-specific functionality and a continuous *State Model*, making it accessible in a combined fashion (Adolphs et al., 2016). For this, the mechanism of the AS's Component Manager is assumed. **CQ5**: Does a given Component Manager offer the State Model?
- The AS uses different views for exposing the properties and functions of the represented Asset to the client. **CQ6**: Does the AS offer views that organize properties and functions of the Asset?
- An AS contains multiple *Partial Model*s composed from hierarchical *Properties* referring to data and functions of the Asset. **CQ7**: Does the AS contain Partial Models with hierarchically organized Properties that refer to data or functions of the associated Asset?
- Following the requirements of nesting and combining an AS may access or be part of other ASs, or may combine many Assets with various CP classifications. **CQ8**: Can an AS be nested with another AS? **CQ9**: Can an AS refer to Assets with different CP classifications?
- An Asset as an I4.0 component may be linked to more than one AS during its lifecycle, implying versioning. **CQ10**: Is a given Asset identifiable as an I4.0-component? **CQ11**: Can the relation between an Asset and its AS be redefined?
- An AS shall be extensible by various types of properties or models for exposing specific aspects of an asset. **CQ12**: Does the AS permit integration of specific models for exposing aspects of the Asset?

From these observations, the revised, larger analytical model as shown in Figure 4 is derived.

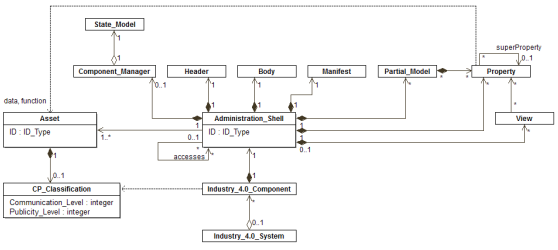

Figure 4: Extended analytical view of the basic terms.

5.3 Modes of Use

The functions of I4.0 components and modes of use for the AS are described in (Adolphs, et al., 2015; DIN, 2016). The following scenarios are considered to capture the dynamics of the AS during the lifecycle of an asset.

An AS may be supplemented to an existing asset:

$$\begin{aligned} AS &\sqsubseteq \geq 1\ refers\_to.Asset \\ a &: Asset \\ as &: AS \sqcap \exists refers\_to.\{a\} \end{aligned} \tag{1}$$

An AS may be replaced by a newer version:

$$\begin{aligned} as1 &: AS \sqcap \exists refers\_to.\{a\} \\ as2 &: AS \sqcap \exists refers\_to.\{a\} \end{aligned} \tag{2}$$

with the $as1 : \exists refers\_to.a$ needing to be retracted in advance and AS instances being subject to versioning.

An AS may be created together with an I4.0 component representation:

$$\begin{aligned} as &: AS \sqcap \exists refers\_to.\{a\} \\ i &: I4.0\_Component \sqcap \exists has\_part.\{as\} \end{aligned} \tag{3}$$

An AS may be extended to functionally cover additional I4.0 components:

$$\begin{aligned} AS \sqsubseteq\ & (\geq 1\ refers\_to.Asset) \sqcap (\geq 0\ accesses.AS) \sqcap \\ & (= 1\ has\_part^{-}.I4.0\_Component) \\ a1 &: Asset \\ a2 &: Asset \\ as1 &: AS \sqcap \exists refers\_to.\{a1\} \\ as2 &: AS \sqcap \exists refers\_to.\{a2\} \end{aligned} \tag{4}$$

It is assumed that both $a1$ and $a2$ qualify as I4.0 components, thus this inference shall be possible:

$$\begin{aligned} i1 &: I4.0\_Component \sqcap \exists has\_part.\{as1\} \\ i2 &: I4.0\_Component \sqcap \exists has\_part.\{as2\} \\ as2 &: \exists accesses.\{as1\} \sqcap \exists has\_part^{-}.\{i2\} \\ &\models i2 : \exists has\_part.\{as1\} \end{aligned} \tag{5}$$

ASs may be nested:

$$\begin{aligned} AS &\sqsubseteq \exists accesses.AS \\ as1 &: AS \\ as2 &: AS \sqcap \exists accesses.\{as1\} \end{aligned} \tag{6}$$

An AS may be (un-)linked from/to another AS: This scenario is basically identical to the nesting scenario. However, for unlinking, asserted knowledge needs to be retracted first.

The AS as well as other involved entities are modeled as distinct individuals for easy information management.

5.4 Ontological Representation

A principle followed in this work is that textual parthood indicators such as *contains*, *consists of*, *has part* can be modeled via a closure leading to meet the desired requirements while neglecting the distinction between structural and functional parthood for now.

First, we addressed the question of how to model an asset as the basic entity for an information system. Here, a partitioning problem exists since an asset shall at any time be either a *template* or *instance* while being at the same time either *tangible* or *intangible*, each being mutually exclusive. This allows for different interpretations and hence OOA models.

With regard to the intended uniform cross-technology knowledge representation, the expected large numbers of assets and types to be managed, the possible modifications an asset and its representation may be subject to during its lifecycle, and the long-term use of the resulting ontology, we found that only one individual should be required to model an asset. Hence, we opted for preserving the distinction of assets as tangible or intangible. This classification remains static during the entire lifecycle of an asset. Specification of an asset as a template or an instance that can be individually managed and tracked is expressed via an *is_template* flag. That way, an asset can be modeled using a single individual. Evolution of assets can be expressed by the *is_directly_based_on* role (see Figure 5[1], following the semantics for UML class diagrams (Berardi et al., 2005)). This asset concept can be extended to employ further models for representing the data, functions and facets of assets (outside the scope of this paper).

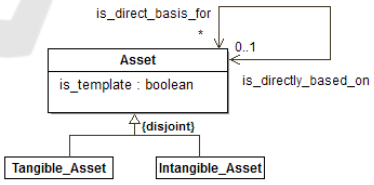

Figure 5: Selected approach for modeling an Asset.
$Asset \sqsubseteq (Intangible\_Asset \sqcup Tangible\_Asset) \sqcap$
$(\leq 1\ is\_directly\_based\_on.Asset)$
$Template \sqsubseteq Asset \sqcap$
$(\leq 1\ is\_directly\_based\_on.Asset) \sqcap \overset{\exists}{\forall} is\_template.\{true\}$
$Instance \sqsubseteq \neg Template.$

An asset is the first entity to be considered in the process of AS management. Thus, we developed the overall ontology as follows. According to RAMI4.0 an Asset may be classified by its CP levels as well

[1]Here and in the following we employ $\overset{\exists}{\forall} r.C$ as a shorthand notation for $\exists r.C \sqcap \forall r.C$.

as by the layer and hierarchy levels it is assigned to. For being classified as an I4.0 component, the CP classification is mandatory. We introduced an abstract superclass *Managed Representative* as the disjoint union of I4.0 components and non-I4.0 components. (Non-)I4.0 components can be modeled as well as I4.0 systems, which are I4.0 components with at least one other I4.0 component and may also contain non-I4.0 components. Hence, only I4.0 components are equipped with AS while all Managed Representatives refer to underlying assets. This can be axiomatized as follows.

$$\begin{aligned} &RAMI\_Communication\_Capability\_Level \equiv \\ &\{Active\_Communication, \\ &Industry4.0\_Communication, \\ &No\_Communication, \\ &Passive\_Communication\} \end{aligned} \tag{7}$$

$$\begin{aligned} &RAMI\_Publicity\_Level \equiv \\ &\{Known\_Anonymously, Known\_Individually, \\ &Managed\_As\_Entity, Unknown\} \end{aligned} \tag{8}$$

$$\begin{aligned} &I4.0\_Component \equiv \\ &Managed\_Representative \sqcap \exists refers\_to.(Asset \sqcap \\ &\underset{\forall}{\exists} has\_RAMI\_C\_Level.(Passive\_Communication \sqcup \\ &Active\_Communication \sqcup \\ &Industry4.0\_Communication) \\ &\sqcap \underset{\forall}{\exists} has\_RAMI\_P\_Level.Managed\_As\_Entity) \\ &\sqcap (= 1\ contains\_Shell.Administration\_Shell) \end{aligned} \tag{9}$$

Although this information can be inferred, explicit modeling for easier information management is proposed. In view of its use in information systems, the AS is modeled to consist of *Header*, *Body* and *Manifest* only. That way, a client can browse the AS since *Header* and *Manifest* are intended for directory purposes only. The *Component_Manager* is modeled to be contained by the *Body*. Additionally, the *Component_Manager* contains the required *State_Model* along with the set of *Partial_Model*s. Basically, all three elements *Header*, *Manifest* and *Partial_Model* contain *Property*s. In accordance with the specification, only the *Manifest* exposes the whole set of *Property*s, while the others expose only partial sets and combinations thereof. A *Property*, with its four defined subtypes, is considered a representative of either a data element or a function of the referenced *Asset*. *View* offers an extensible set of predefined subtypes for providing logical groupings of *Property*s.

Since an AS may logically contain other AS, this is expressed via the *accesses* relationship. For this, implementations need to prevent cycles. In OWL this is supported by defining the relationship as irreflexive and transitive[2]. Figure 6 shows the followed modeling approach.

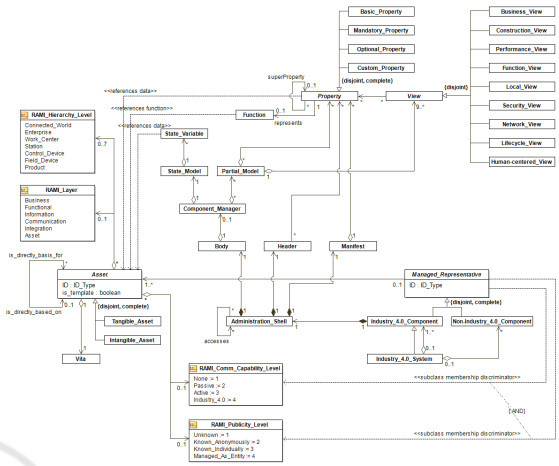

Figure 6: Basic model of the RAMI4.0 AS.

The AS concept can be axiomatized as follows:

$$\begin{aligned} &Administration\_Shell \sqsubseteq (= 1\ \overline{contains\_Shell}^{-}. \\ &I4.0\_Component) \sqcap \\ &(= 1\ has\_Shell\_Part.Header) \sqcap \\ &(= 1\ has\_Shell\_Part.Body) \sqcap \\ &(= 1\ offers\_Manifest.Manifest) \end{aligned} \tag{10}$$

5.5 Application

With the AS as an interface to an asset, the existence of the latter is a prerequisite for modeling the former. Further, both models are subject to changes during their individual lifecycles. With regard to the intended usability of the ontology for model-based systems engineering, we propose to explicitly distinguish two separate engineering phases in providing an AS for an asset (cf. Figure 7).

The first phase is called *asset engineering phase*. In this, an asset is modeled according to its requirements. Typically, such representations would involve specific standards, which might only be internally visible and thus result in asset-specific structures only. In this phase the asset is also unaware of its potential classification as an I4.0 component, as this depends on its capabilities. The result of this phase is required to be a self-contained model of the asset.

[2]The OWL implementation can be obtained from https://figshare.com/articles/dataset/rami_as/6808988.

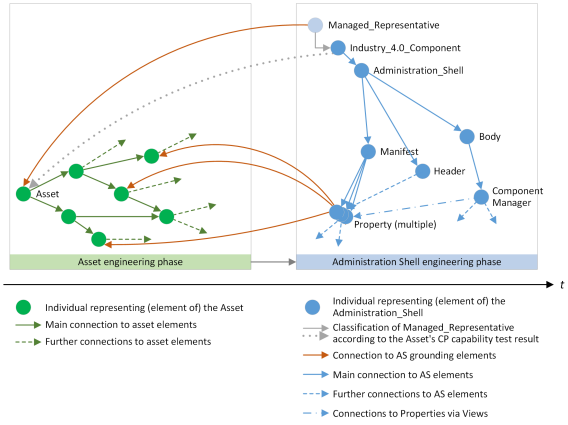

Figure 7: Conceptual phases for AS engineering. The existence of an engineered asset is prerequisite for the engineering of its associated AS.

The second phase is referred to as *AS engineering phase*. In this, the model of the asset is evaluated, i.e., capability-tested, according to the criteria for I4.0 components (this process is not in scope of this paper). If the asset qualifies as an I4.0 component, the AS model can be created, along with its mandatory structural parts. Further, AS elements such as properties or views are added and linked to the underlying elements of the asset model, effectively resulting in the AS as a wrapper. The outcome of this phase is a self-contained AS with its elements linked to those elements of the asset that it needs to expose in order to be usable as I4.0 component.

This two-step process decouples asset modeling from AS modeling, allowing for independent engineering, management and distributions of assets, their AS and combinations thereof.

5.6 Evaluation

In Section 5.2, main CQs for basic AS model checking are identified. These are to ensure that AS instances transferred between ontology-based and non-ontology-based systems follow identical structures. We analyze how these CQs can be answered by the developed ontology with its intended application (cf. Figure 6). Table 1 reports on the results for each CQ. The list, however, shows that it is not possible to answer **CQ10** by the ontology itself. Rather, additional business logic is needed for performing the required tests. Valid AS instances can then be handled using structurally identical query patterns on ontology-based and OO systems. This is exemplified by the access path in Listing 1 for accessing an asset's AS's *Performance_View*. This further reveals the need for additional but in (DIN, 2016) insufficiently specified explicit access to overall views on an asset via an AS's *Header*.

Listing 1: SPARQL access.

```
SELECT ?pv WHERE {
?a a :Administration_Shell;
   :has [a :Body;
   :has [a :Component_Manager;
   :has [a :Partial_Model;
   :has ?pv]]].
   ?pv a :Performance_View. }
```

Table 1: Competency questions and answers.

CQ	Answers to CQs
CQ1	An AS is part of an I4.0 component, which is subclass of an identifiable representative that is associated with an Asset.
CQ2	Header and Body are parts of an AS.
CQ3	Manifest is part of an AS.
CQ4	An AS may access other AS. An AS contains a Body. A Body can contain a Component Manager.
CQ5	A State Model is part of a Component Manager.
CQ6	An AS contains a Body, which may contain a Component Manager, which contains Partial Models. A Partial Model is composed of Properties and Views, with the latter referring to the former.
CQ7	An AS contains a Body, which may contain a Component Manager, which contains Partial Models. A Partial Model is composed of Properties. A Property references data elements or functions of the Asset associated with the AS, and it may have a *Property* that is its *superProperty*.
CQ8	An AS may access other AS or be part of another AS.
CQ9	An AS may refer to more than one Asset at one time.
CQ10	Via testing, an Asset with its characteristics can be examined for its RAMI Communication Capability Level and its RAMI Publicity Level. The results of this examination can be stored as part of the Asset. According to these levels, a Managed Representative, which can be classified as I4.0 component, can be assigned to the Asset.
CQ11	An AS is part of an I4.0 component, which is a subclass of Managed Representative. This holds the reference to the associated Asset. This reference can be retracted and reasserted.
CQ12	Via Body and Component Manager, an AS contains Partial Models. Specific models can be integrated as specializations of Partial Model, which may contain Properties and Views that reference specific functions or data of the Asset.

6 CONCLUSION AND FUTURE WORK

In this article, we presented an ontology for the RAMI4.0 AS concept. This ontology can be used with systems based on different technologies or paradigms. By applying OOAD to the descriptions in the RAMI4.0 standard documents we developed the ontology to be implementable with OO systems. Afterwards, we extended it with its logical underpinnings to bridge the gap to ontology-based systems, permitting a structurally identical implementation. The approach introduces a strict separation of the concepts of Asset and AS and of their respective models, using only loose couplings in between. In an attempt to create and manage these models, we defined them to be engineered in two consecutive phases, which enables selective exposure of Asset information via the AS.

We see this approach as an extensible basis for information systems to enhance the interoperability of assets from different manufacturers as well as the supply of necessary asset information to customers. For future developments we consider the provision of standard software components, which encapsulate the required functionality so that the involved complex model-based processing becomes easily usable. This is essential for a wide acceptance of the AS concept.

REFERENCES

Adolphs, P., Auer, S., Billmann, M., Hankel, M., Heidel, R., Hoffmeister, M., Huhle, H., Jochem, M., Kiele, M., Koschnick, G., Koziolek, H., Linke, L., Pichler, R., Schewe, F., Schneider, K., and Waser, B. (2016). Structure of the administration shell. Status report, ZVEI and VDI.

Adolphs, et al. (2015). Reference Architecture Model Industrie 4.0 (RAMI4.0). Status report, ZVEI and VDI.

Berardi, D., Calvanese, D., and De Giacomo, G. (2005). Reasoning on UML class diagrams. *Artif. Intell.*, 168(1-2):70–118.

Booch, G., Maksimchuk, R. A., Engle, M. W., Young, B. J., Conallen, J., and Houston, K. A. (2008). Object-oriented analysis and design with applications, third edition. *ACM SIGSOFT Software Engineering Notes*, 33(5).

Bradner, S. (1997). Key words for use in RFCs to indicate requirement levels. RFC 2119, Internet Engineering Task Force (IETF).

Brettel, M., Friederichsen, N., Keller, M., and Rosenberg, M. (2014). How virtualization, decentralization and network building change the manufacturing landscape: An industry 4.0 perspective. *Int. Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering*, 8(1):37–44.

Diedrich, C., Belyaev, A., Schroder, T., Vialkowitsch, J., Willmann, A., Usländer, T., Koziolek, H., Wende, J., Pethig, F., and Niggemann, O. (2017). Semantic interoperability for asset communication within smart factories. In *22nd IEEE Int. Conf. on Emerging Technologies and Factory Automation, ETFA, Limassol, Cyprus, September 12-15*, pages 1–8.

DIN (2016). DIN SPEC 91345: Reference Architecture Model Industrie 4.0 (RAMI 4.0). Technical report, DIN.

Grangel-González, I., Halilaj, L., Coskun, G., Auer, S., Collarana, D., and Hoffmeister, M. (2016). Towards a Semantic Administrative Shell for Industry 4.0 Components. In *Tenth IEEE Int. Conf. on Semantic Computing, ICSC Laguna Hills, CA, USA, February 4-6*, pages 230–237.

Kharlamov, E., Grau, B. C., Jiménez-Ruiz, E., Lamparter, S., Mehdi, G., Ringsquandl, M., Nenov, Y., Grimm, S., Roshchin, M., and Horrocks, I. (2016). Capturing industrial information models with ontologies and constraints. In *15th Int. Semantic Web Conf. (ISWC)*, pages 325–343.

Lee, J., Bagheri, B., and Kao, H.-A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3:18–23.

Marseu, E., Kolberg, D., and Weyer, S. (2017). Exemplary transfer of the RAMI 4.0 Administration Shell to the SmartFactory KL System Architecture for Industrie 4.0 Production Systems. White paper, SmartFactoryKL.

Negri, E., Perotti, S., Fumagalli, L., Marchet, G., and Garetti, M. (2017). Modelling internal logistics systems through ontologies. *Computers in Industry*, 88:19–34.

Pethig, F., Niggemann, O., and Walter, A. (2017). Towards industrie 4.0 compliant configuration of condition monitoring services. In *15th IEEE Int. Conf. on Industrial Informatics, INDIN Emden, Germany, July 24-26*, pages 271–276.

Platform Industrie 4.0 (2019). Details of the Asset Administration Shell. Specification, Platform Industrie 4.0.

Ren, Y., Parvizi, A., Mellish, C., Pan, J. Z., van Deemter, K., and Stevens, R. (2014). Towards competency question-driven ontology authoring. In *The Semantic Web: Trends and Challenges - 11th Int. Conf., ESWC, Anissaras, Crete, Greece, May 25-29. Proceedings*, pages 752–767.

Tantik, E. and Anderl, R. (2017). Integrated data model and structure for the asset administration shell in Industrie 4.0. *Procedia CIRP*, 60:86–91.

ZVEI (2016). Beispiele zur Verwaltungsschale der Industrie 4.0-Komponente – Basisteil. White paper, ZVEI.