

# Semantic Asset Administration Shells in Industry 4.0: A Survey

Sadeer Beden  
Department of Computer Science  
Swansea University  
United Kingdom  
sadeer.beden@swansea.ac.uk

Qiushi Cao  
Department of Computer Science  
Swansea University  
United Kingdom  
qiushi.cao@swansea.ac.uk

Arnold Beckmann  
Department of Computer Science  
Swansea University  
United Kingdom  
a.beckmann@swansea.ac.uk

**Abstract**—The Asset Administration Shell (AAS) is a fundamental concept in the Reference Architecture Model for Industry 4.0 (RAMI 4.0), that provides a virtual and digital representation of all information and functions of a physical asset in a manufacturing environment. Recently, Semantic AASs have emerged that add knowledge representation formalisms to enhance the digital representation of physical assets. In this paper, we provide a comprehensive survey of the scientific contributions to Semantic AASs that model the Information and Communication Layer within RAMI 4.0, and summarise and demonstrate their structure, communication, functionalities, and use cases. We also highlight the challenges of future development of Semantic AASs.

**Index Terms**—Industry 4.0, RAMI 4.0, Semantic Asset Administration Shell, Knowledge-driven manufacturing

## I. INTRODUCTION

Digital transformation in manufacturing is enabling a shift in paradigm towards smart manufacturing, which makes use of new technologies and concepts. The advancements in Cyber-Physical Systems (CPS), Industrial Internet of Things (IIoT), Cloud Computing, Artificial Intelligence (AI), and others key enabling technologies, has lead to a vision that machines, processes, and products are connected via intelligent networks utilizing information and communication technologies. This vision is called the fourth industrial revolution, commonly referred to as *Industry 4.0* [1]. To realise this paradigm shift, there are many initiatives developing architecture models to support Industry 4.0. The Reference Architecture Model for Industry 4.0 (RAMI 4.0), developed by Plattform Industrie 4.0 and other associations including ZVEI and VDMA [2], includes the fundamental concept of an *Asset Administration Shell (AAS)* that serves as the virtual and digital representation of all information and functions of a physical asset in a manufacturing environment. Within an AAS, big industrial data is stored in a strict format through one or more sub-models to enable real-time condition monitoring of machines and machine tools.

As the manufacturing domain is highly data and knowledge-intensive, uniform knowledge representation of physical re-

sources and seamless integration of heterogeneous data for analytic tasks are needed as the basis to automate decision making processes for production systems. To address these challenges, knowledge-driven manufacturing methods have received significant attention in recent years. Normally, these methods involve knowledge models during decision making processes in order to provide rich data semantics throughout the manufacturing activities [3]. This trend has enabled the adoption of *Semantic Asset Administration Shells*. A Semantic AAS uses knowledge representation formalisms such as Resource Description Framework (RDF) and Web Ontology Language (OWL) to create digital representations of physical assets [4]. Semantic AASs enable interoperable communication among machines, and compatibility with diverse digital frameworks and architecture.

Although Semantic AASs are often mentioned in the context of knowledge-driven smart manufacturing, to the best of our knowledge, there lacks a survey contribution to summarise and demonstrate their structure, functionalities, and use cases. In this paper, we aim to fill this gap by providing a comprehensive survey on this subject. The aim of the survey is to provide a vision and outlook on the critical and cutting-edge technologies used within Semantic AASs.

The rest of the paper is structured as follows. Firstly, Section II gives a brief introduction to RAMI 4.0 and Industry 4.0 components. In Section III, we introduce the survey methodologies used in this paper. Afterwards, in Section IV, we present an in-depth construct of Semantic AASs by introducing their structure, communication technologies, characteristics, and functionalities. Section V demonstrates some typical application use cases of Semantic AASs. Finally, Section VI concludes the paper and outlines open challenges within this research field.

## II. BACKGROUND

This section gives a brief description of relevant topics and concepts coined for Industry 4.0.

### A. RAMI 4.0 Model

The Reference Architecture Model for Industry 4.0 (RAMI 4.0) is a 3D model frame, in the form of a Smart Grid Architecture Model (SGAM) that highlights the most

S. Beden was supported by the Engineering and Physical Sciences Research Council [grant number EP/T517537/1] and by Tata Steel.

Q. Cao and A. Beckmann (in part) were supported by the Engineering and Physical Sciences Research Council [grant number EPSRC EP/S018107/1].

important and key aspects of achieving Industry 4.0 [2]. The model uses a divide and conquer approach to tackle the complex technical to business processes and architecture of Industry 4.0. Figure 1 displays the RAMI 4.0 model.

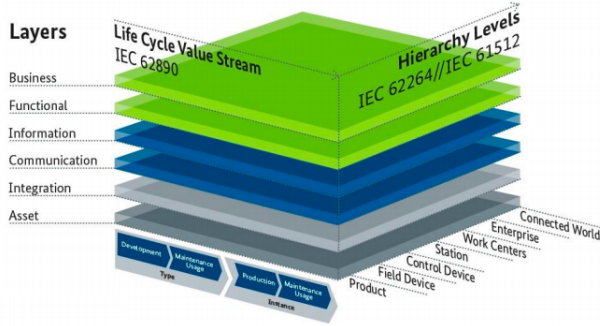


Fig. 1. RAMI 4.0 portrayed by Plattform Industrie 4.0 and ZVEI [2].

Firstly, the *Layers* axis, as seen on the left hand side of Figure 1, represents the six processes that describe the technical and business perspectives of a product [5]. The asset layer represents the *things* that are in the physical world whereas the communication, information, functional, and business layers represent the asset’s form in the digital world. The integration layer is responsible for bridging these two worlds together by reading and storing all digital information of assets, typically through sensors.

The second axis is the *Life Cycle and Value Stream* axis. This axis expresses the lifecycle and associated value stream of products, machinery, factories, and other assets according to the IEC 62890 standard. The purpose of this axis is to record all data of an asset, from its first idea and initial design, all the way until the asset is scrapped.

The final axis, the *Hierarchy Levels* axis, describes the necessary advancements in hierarchy required for achieving Industry 4.0. The core structure of this axis is based on the IEC-62264 (Enterprise-control system integration) and IEC-61512 (Batch control) standards within a manufacturing environment. These two standards cover many sectors of a factory, from process industry to factory automation. The RAMI 4.0 model adds two additional layers:

- *Product*: The product itself is a key aspect of Industry 4.0 and still part of the model even after it has been manufactured and sold.
- *Connected World*: Connected world portrays cloud storage and collaboration beyond one factory, between other factories, suppliers, customers, and so forth, to achieve greater connectivity than at factory level.

### B. Industry 4.0 Components

An Industry 4.0 component (I4.0 component) serves as a model, the purpose of which is to hold all data of assets that can describe properties, configuration parameters, and functions of other assets [6].

The British Standards Institution (BSI) group, known for developing standards in Manufacturing and Engineering, de-

finer an I4.0 component as a *globally and uniquely identifiable participant that is capable of communication within an I4.0 system* [2], [7]. This can range from a module within a machine to a whole production system [4]. In general, an I4.0 component is composed of an *Asset* and an *Asset Administration Shell* as described below:

The **Asset** layer, as previously mentioned, represents all real, physical things within a manufacturing environment that can be connected digitally. Taking cold rolling mills as an example, this will include physical equipment or products such as the rolling mills, the rolls, the coils, and other machinery. This also includes immaterial items such as documents, plans, orders, and even the shift workers involved.

An **Asset Administration Shell (AAS)** serves as a virtual and digital representation of all information and functions of an asset, and acts as an administration interface accessible within an Industry 4.0 network [8]. Communication between assets occurs between their AAS, whereby the AAS provides and moderates controlled access to the data [9]. Additionally, an AAS contains an administration interface where its data may be accessed through external application services, e.g. by calling an Application Programming Interface (API) from an Enterprise Resource Planning (ERP) system. The data itself is stored in a strict format through one or more sub-models. The sub-model approach of storing data is expandable and flexible, and storage location on the data model is of free choice [10]. Thereby, similar assets from different vendors may contain different sub-models where the AAS simply acts as the data manager [10]. The data set should contain the full lifecycle of the asset, from its early stages of being planned and manufactured until it is scrapped.

The stored data is processed by data-driven methods for understanding manufacturing processes, thus enabling real-time condition monitoring and maintenance of the physical assets. The AAS is the key enabler of transforming legacy industrial components into I4.0 components.

As knowledge-based frameworks, **Semantic AASs** equip traditional AASs with knowledge representation formalisms to allow semantic metadata modelling. Resource Description Framework (RDF) and Web Ontology Language (OWL) are prime examples of widely used models for standardizing metadata and semantics [11]. Benefiting from the explicit data semantics specified in semantic models, these Semantic AASs enable interoperable machine-to-machine data exchange and communications [4].

## III. SURVEY METHODOLOGY

### A. Research Questions

This survey paper will investigate research literature related to Semantic AAS as an I4.0 component. There is much literature on Industry 4.0 and I4.0 components, but the understanding of Semantic AASs is often overlooked. This review aims to provide a clearer and more concrete understanding of this topic.

The work of this survey paper is guided by answering the following research questions:

- What are the key characteristics of Semantic AASs?
- What sub-domains of manufacturing have been studied in the literature?
- What are the typical application cases?

In section IV, we answer the first two research questions by reviewing the existing literature on Semantic AASs. In section V, we answer the third research question by summarising the application cases of Semantic AASs.

#### B. Inclusion & Exclusion Criteria

In this paper, we focus on the Semantic AASs that model the Information Layer and Communication Layer within RAMI 4.0. Indeed, these two layers are of vital importance for the digital representation and real-time data analytics of physical assets. Although there are Semantic AASs that address the challenges in other layers, the amount of papers found are almost negligible compared to the two layers mentioned above. Because of this, research papers that focus on other layers are excluded from this study.

During this study, research papers, technical reports, and white papers are considered valuable resources. However, duplicates among these resources are excluded. This survey primarily focuses on English papers only.

#### C. Corpus & Web search engine

To derive greater understanding of Semantic AASs, an analysis of all relevant publications was carried out. This topic contains strong blends of scientific literature and engineering principle. Because of this, this survey used different bibliographic databases, including Institute of Electrical and Electronics Engineers (IEEE) Xplore for the engineering side, and ACM for the scientific literature side. Finally, Scopus was used which provided a combination of literature found in both IEEE and ACM, as well as other valid sources.

#### D. Survey results

At the initial stage, the main keywords searched were *Asset Administration Shell*. Scopus provided a result of 61 unique hits from 2017 to 2020; 70.4% of which were conference papers, 26.2% Articles and 3.4% categorized as others. The vast majority of these publications were from IEEE and only a handful from ACM. Very few results were from other bibliographic databases, such as Science Direct and Web of Science. As a result, this paper includes all relevant results but will primarily focus on the results found from IEEE as well as ACM. These keywords resulted in 37 papers on IEEE Xplore from the years 2017 to 2020. This small number conveys that this field of research is new and upcoming. From 2017 to 2020, the average annual number of papers are nine. In 2017, four papers mentioned “Asset Administration Shell”, followed by eight papers in 2018. This value jumped to 16 papers in 2019 and dipped to seven from January to October 2020. Furthermore, there are no papers prior to 2017 which further emphasises that this field is in early stages of research.

In comparison, ACM only contained six papers from 2017 to 2020. This conveys that most research is happening from an engineering point of view.

When expanding the search to include *Semantic AASs*, *RDF-based AASs*, *knowledge-based AAS*, we found under ten relevant papers from the above digital libraries. Thus, we extended our search with Google Scholar and other available sources, and discovered seventeen relevant papers in total. We will introduce these papers in the next section.

### IV. SEMANTIC ASSET ADMINISTRATION SHELLS: THE STATE OF THE ART

This section gives a detailed description of the Semantic AASs that appear within the papers mentioned in the previous section. The papers are categorized according to which layer the Semantic AASs serve within the RAMI 4.0 architecture. In this paper, we focus on two layers: the Information Layer and the Communication Layer. Most of the reviewed literature contribute to these two layers.

#### A. The Information Layer: RDF-based AASs

Normally, Semantic AASs use formal knowledge modelling languages such as RDF and OWL to create digital representations of physical assets. These type of AASs enable industrial devices to communicate and understand each other, for the goal of semantic interoperability.

The first RDF-based data model for AAS was introduced by Grangel-González et al. in 2016 [4]. This paper proposed to add a semantic layer to AAS, stating the advantages of adopting an RDF-based approach. The developed AAS benefited from the traits of RDF schema as it provided decentralized and extensible global *Identification*, unified data *Integration*, and *Coherence* among new taxonomies, vocabularies, and ontologies. Because of the uniform information representation, existing standards and asset data could easily be integrated and referenced. To access relevant asset information, SPARQL was used as a query language to retrieve data captured by the Semantic AAS. The authors demonstrated the Semantic AAS using a motor controller as a use case. This Semantic AAS was extended by the same authors later in [12]. Here, they proposed three improvements: 1) A significant extension of the RAMI 4.0 vocabulary for describing sensor data, units of measurement, and product information; 2) An RDF-based vocabulary which incorporated the international standard IEC 62264 that aligned with the RAMI 4.0 vocabulary; 3) A real-world use case on black carbon monitoring in industry. The prominent characteristic of their work is the translation of IEC 62264 standard-based RAMI 4.0 model to an RDF-based uniform vocabulary. This translation provided a common description of Industry 4.0 components using a unified knowledge representation language.

Similarly, Bader et al. introduced Semantic AASs in more detail [13]. In their paper, the authors filled the gap between industrial reference frameworks and semantic description of the physical world. To achieve this goal, the authors mapped the latest AAS data models into RDF format, and used Shapes Constraint Language (SHACL) shapes to enable schema validation. Firstly, when mapping to RDF, they stated that the AAS object is the root of every AAS. Thereby, it could also be the

entry for traversing the Semantic AAS graph. To cover the semantics of an asset, the authors created *rdf:type*, *rdfs:label*, *rdfs:comment* to cover the class assertions, name, description, and kind attributes of the asset. Secondly, a fundamental requirement of an AAS is to have a unique identifier — the authors declared that it is possible to use the RDF's URI as a possible identifier. Otherwise, custom formats such as International Registration Data Identifiers (IRDI) are allowed. Submodels and SubmodelElements of AAS were realized and modelled using Operations, ReferenceElements, Files, binary objects and Properties classes. *rdf:property* was developed to align the Property class with RDF's graph model. To execute this mapping, the authors used an open-source tool named RMLMapper. They also illustrated how the transformation to the semantic data model was able to decrease the amount of required storage space.

Tantik and Anderl aligned the Plattform Industrie 4.0 AAS guidelines with the World Wide Web Consortium (W3C) specifications in [14]. W3C provides the Object Memory Model (OMM) which contains block-based digital object memories (DOMe) to present highly standardized meta-information. Combining OMM with an integrated component data model produces a new data model: the Component Data Model (CDM). The authors used CDM to present data of nested I4.0 components. Using a robot arm as a use case, they presented an integrated data model that used a central remote maintenance platform (CRMP) as a form of communication between AASs.

Hua and Hein investigated AutomationML from a semantic point of view in [15]. They argued there was a lack of semantic support for automated machine processing, and thereby, used OWL to transform AutomationML data to a formal and declarative semantic representation. They combined Inductive Logic Programming (ILP) techniques with automated reasoning to obtain meaning of system unit classes. To demonstrate this, they used a machine learning example using OCEL and CELOE algorithms to demonstrate concept learning in AutomationML using DL-Learner.

Thuluva et al. dived into the Web of Things (WoT) and Semantic Web Technologies to address cross-domain interoperability problems in Automation Systems (AS) [16]. The WoT standard provides an interface named *Thing Description (TD)* that was developed to describe an object, as well as its metadata and interaction. This paper employed SWT in WoT and used TD as the basis to model a WoT-enabled AAS.

### B. The Information Layer: OWL ontology-based AASs

As another type of semantic models, OWL-based ontologies also play an important role for describing manufacturing entities within a Semantic AAS. In computer science, an ontology is considered as “an explicit specification of a conceptualization for a domain of interest” [17]. Within this definition, specification refers to an act of describing or identifying something precisely. This requires the concepts and relationships in ontologies to be clearly defined by using formal logic. Since ontologies are developed based on formal logic foundations, they have been pervasively used in industry

to ensure the semantic interoperability among different systems and users. In the manufacturing domain, ontologies play a key role in many distributed intelligent systems as they provide a shared, machine-understandable vocabulary for information exchange among dispersed agents [18]–[21]. Large ontologies are designed in a modular structure to enhance their reusability, extendability, and easy maintenance.

To model the concepts and relations within ontologies, the W3C developed a formal ontology language named OWL. OWL is a component of SemanticWeb that explicitly represents the meaning of terms in vocabularies and the relationships between those terms. The representation of terms and their interrelationships form an ontology. In the following section, we review the existing OWL ontologies and their rule-based extensions that are relevant to Semantic AASs.

The first group of studied ontologies were developed to model *product-related* concepts for manufacturing. Vegetti et al. [22] proposed a PProduct ONTOLOGY (PRONTO) for the domain of Complex Product modelling. With primary focus on product structure, this ontology considered different abstraction levels of product concepts such as *Family* and *Variant*. It also extended the conventional product structure representations (BOMs) with considering composition and decomposition structures of products within a variety of manufacturing environments. However, PRONTO was not capable of referring to the existing international standards related to the modeling of product structure, processes, and features. To address this weakness, Panetto et al. [23] developed ONTO-PDM, which is an ontological model considered as a facilitator for interoperating all application software that shared information during the physical product lifecycle. The distinctive merit of this ontology was its incorporation of standardisation initiatives, such as International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) standards. This nature of ONTO-PDM allowed the management of heterogeneous information scattered within organizations, by formalising the knowledge related to product technical data [23]. The MASON ontology [24], developed by Lemaignan et al., is another prominent ontology for manufacturing. This OWL ontology was developed to draft a common semantic net in manufacturing. It conceptualised three core concepts: *Entities*, *Operations*, and *Resources*. In more detail, the product information was specified by three sub-concepts: *Geometric Entities*, *Raw Material*, and *Cost Entities*. The representation of product-related concepts were linked to the description of manufacturing processes and resources under the proposed semantic net in manufacturing.

The second group of reviewed ontologies focused on *process-related* concepts for manufacturing. A manufacturing process is a sequence of activities through which the raw materials are assembled, integrated, and transferred into a final product. To model manufacturing processes, Grüninger et al. [25] proposed the Process Specification Language (PSL) Ontology, which is a semantic model to facilitate correct and complete exchange of process information among manufacturing systems. Within the PSL ontology, the authors

formalized the concept of *process* in the form of first-order logic theories. This formalisation has been widely used in domain applications such as process modeling, process monitoring, process planning, simulation, project and workflow management [25]. Another process-related OWL ontology is the one developed by Cao et al. [26]. In their work, a domain ontology was developed to formalise essential concepts and relations about condition monitoring. The ontology was structured into three sub-modules, namely the *Manufacturing* module, the *Context* module, and the *Condition Monitoring* module. This ontology was used within a Cyber Physical System to enable real-time predictive maintenance. A case study on a conditional maintenance task of bearings in rotating machinery was performed to evaluate the proposed ontology. This ontology was extended by the same authors in their recent work [27]. The extended new ontology was named *Manufacturing Predictive Maintenance Ontology (MPMO)*. MPMO was used together with Semantic Web Rule Language (SWRL) rules to enable ontology reasoning, for detecting and predicting possible anomalies that may happen within manufacturing processes. The effectiveness and usefulness of MPMO was tested on a real-world data set collected from a semi-conductor manufacturing process.

The third group of ontologies addressed *resource-related* concepts for manufacturing. In general, the concept *Resources* is defined as physical objects that can execute a range of operations during a manufacturing process. Borgo and Leitão [28] formalized the concept of *Resources* by developing a core ontology for manufacturing scheduling and control environments. In their work, a *Resource* was defined as “an entity that can execute a certain range of jobs, when it is available, as long as its capacity is not exceeded” [28]. The core ontology was implemented as part of a multi-agent manufacturing control system by using the Java Agent Development Framework (JADE) framework. The author concluded that an established foundational ontology plays an important role in handling heterogeneous data generated by manufacturing control applications, especially those built upon distributed approaches such as multi-agent systems. The notion of *Resources* was also studied in the MASON ontology [24], where *Resources* are further classified into four sub-notions: 1) *Machine-tools* (e.g., turning machines, drilling machines, milling machines); 2) *Tools* (e.g., forging die and punch, turning tool, founding pattern and mould); 3) *Human Resource* (e.g., procedure expert, handling operator, programming operator); and 4) *Geographical Resources* (e.g., plants, workshops). The modelling of manufacturing resources was of vital importance for estimating the total cost for manufacturing activities.

### C. The Communication Layer: Semantically-enhanced OPC-UA

To tackle the Communication layer of RAMI 4.0, a joint working group including OPC Foundation, ZVEI and VDMA have chosen the OPC Unified Architecture (OPC-UA) as the standard for machine to machine communication [29]. OPC-UA is an enhanced, platform-independent, connection-oriented

communication protocol based on service-oriented architecture (SOA) developed by the OPC Foundation [30]. The OPC-UA protocol follows the IEC 62541 standards [31] which are derived from the OPC-UA’s Core, Access and Utility specifications [32].

Within RAMI 4.0, the Communication Layer provides components for the communication between machines, devices, production lines, and products. To enable this goal, data communication standards are used to offer required communication between sensors, actuators, and smart devices. In Industry 4.0, OPC-UA has emerged as a widely used standard for data exchange and communication in smart factories [33]. Recently, semantic technologies are considered as a solution to annotating data and provide unambiguous data semantics. The provided data semantics are machine-readable information that allows AASs to perform required actions intelligently without human intervention.

When tackling the communication aspects, Grangel-González et al. used RDF, and STO ontology to describe I4.0 communication standards in [34]. They provided building blocks for the implementation of knowledge graphs for Smart Factory Standards to enable mapping and semantic integration. They used existing ontologies such as *MUTO* (for tagging), *FOAF* (to represent agents & linking documents), and *DCTERMS* (for document meta-data and *RAMI* (for vocabulary). Afterwards, they described the main STO classes. Some examples included *sto:Standard*, *sto:SDO*, *sto:Domain*, *sto:isPartOf*, *sto:relatedTo* e.t.c.. They used VoCol as an integrated environment to view and explore these ontology classes. Using these classes, the authors provided a semantic description of the OPC Foundation as well as OPC-UA. They also provided a use case for searching standards and their metadata.

Katti et al. [35] focused on the concept of the Semantic Web Services (SWS), well known for allowing machines to connect without human intervention. The authors integrated OWL-S concepts into the OPC-UA specification, expanding to the last layer of the automation pyramid, the shop floor. To achieve this, they integrated the edge component *GeSCo*, which also tackled the connectivity and network latency challenges in Cloud MES manufacturing systems. *GeSCo* communicated with the manufacturing resources through the OPC-UA protocol. This approach added flexibility in generating orchestration plans, to overcome any unpredicted events in production. Their work was further developed in [36] where they introduced *Semantic Annotations (SAWSDL)* concepts for more added benefit. They named this approach *Semantically Annotated OPC-UA (SA-OPC-UA)*. They stated that the underlying enabling technology for SAWSDL was WSDL. WSDL contains extension attributes that enable semantic annotations to describe the syntax of the web services and their operations. They stated that there was no equivalent facility in OPC-UA that could achieve the same goal.

#### D. The Communication Layer: the Semantic Web of Things

In Industry 4.0, the Internet of Things (IoT) aims to create a network of physical objects that are embedded with smart sensors, actuators, and software. To enable real-time data analytics, AASs are required to be equipped with low-latency data exchange and communication capabilities. IoT is a promising solution for this task. However, IoT suffers from a lack of interoperability which leads to its weakness in data management in pervasive and heterogeneous environments [37], [38].

One of the earliest contributions to the Semantic Web of Things was proposed by Pfisterer et al. in [39]. In their work, a service infrastructure named SPITFIRE was developed as an architecture of the Semantic Web of Things. SPITFIRE provides uniform vocabularies to integrate descriptions of sensors and physical objects with the linked open data (LOD) cloud, by which the analysis of data is accomplished on the web. It also provides a comprehensive representation and integrated abstractions for physical objects, their high-level states, and how they were linked to sensors. By this, users were able to access the current status of real-world entities. This vision was achieved by embedding a semantic search engine in the architecture which integrated different static and dynamic data sources in a seamless way [39].

Another framework for the Semantic Web of Things was developed by Ruta et al. in [40]. To associate semantic annotations to real-world objects, locations, and events, the authors outlined a novel general framework for the Semantic Web of Things. The framework *ubiquitous Knowledge Base (u-KB)* is based on an evolution of the classical Knowledge Base model [40]. To enable data annotation, domain ontologies were used as conceptual models for a particular domain of interest. The data annotation allows semantic-based dynamic resource dissemination and discovery within a mobile communication network.

In a more recent work, Jara et al. analysed the IoT convergence issue by presenting the different architecture levels of the Semantic Web of Things [41]. The focus was addressed on the trends for capillary networks and for cellular networks with standards such as IPSO, ZigBee, OMA, and the oneM2M initiative [41]. The work paved the way for developing a semantic layer for the IoT by giving a comprehensive analysis of each technology. The analysed technologies are mainly the common internet protocols, including IPv6, Hypertext Transfer Protocol (HTTP), Constrained Application Protocol (CoAP), the Internet Engineering Task Force (IETF) protocols.

The reviewed articles in this section mainly contributed to the Information and Communication Layers of RAMI 4.0. Table I summarises our survey results regarding the domains, information and communication models, and achieved goals of the reviewed papers.

#### V. SEMANTIC ASSET ADMINISTRATION SHELLS: APPLICATION CASES

In this section, we abstract from technologies and focus on application cases of Semantic AASs.

Grangel et al. demonstrated the semantic implementation of a *Servo Motor Controller* used for automation within an Engineering and Manufacturing domain in [4]. The same authors also semantically described industry sensors in legacy systems in [12]. Within these legacy systems, the authors used the industrial data set *AirProbe* from an SQL dump. This dump contained sensor information such as geospatial locations and measurements of black carbon concentrations, temperature and humidity [12]. From this data, they were able to query active sensors within a given time interval and displayed them as geographical coordinates on an interactive map.

Similarly, Pfisterer et al. also used sensors as a use case [39]. However, their application focused on connecting sensors to the internet and the web, as well as searching for specific semantic entities from the sensors. Their application valued and attempted to derive real world entities and their high-level status (e.g. meeting room that is occupied) over simply displaying the sensor and its raw data (sensor 536 with motion detection at time  $T$ ). To realise this, appropriate technologies of mechanisms that establish an explicit mapping between sensors and real life entities was looked into.

Katti et al. [36] made use of a manufacturing production scenario of a shop-floor. Their example included a production routing consisting of three unique operations: welding, color spraying, and quality checking. Firstly, these three tasks published their meta-data to the appropriate server that could send and retrieve routine details from a Cloud Manufacturing Execution System (MES), then back to the server. The server then assigned tasks and acknowledgements to each operation starting with welding.

Bader et al. [13] developed three examples of mapping Asset Administration Shells into Semantic Asset Administration Shells. The first AAS represented a Raspberry Pi (Pi 3B+) that contained three sub-models that stored technical characteristics data, documentation materials (product sheet and usage manual) and asset descriptions, constructed by a total of 52 SubmodelElements. In contrast, the second AAS represented an electronic Automation Controller for automation facilities, constructed by three sub models with more than 100 SubmodelElements. Finally, the third AAS represented a Multi-Protocol controller; this AAS contained eight submodels and over 150 SubmodelElements. When evaluating the mappings, they discovered that some expressions were not able to be transported from AAS to Semantic AAS as RDF struggled to sufficiently present some constructs e.g. the Property class in [13]. This also occurred as many input entities contained redundant information in their example.

Hua et al. [15] used *DL-learner* framework to adopt the OCEL and CELOE learning algorithms for concept learning in AutomationML. The authors were primarily interested in the accuracy and efficiency of these algorithms and thereby developed three scenarios that follow typical system unit class structures, each of which having unique complexities. The data set (AutomationML data) contained 222 classes, 63 properties and 61 individuals. Their tests derived that CELOE is much slower than OCEL, and not determined by the complexity of

TABLE I  
DOMAINS, INFORMATION AND COMMUNICATION MODELS, ACHIEVED GOALS OF THE REVIEWED PAPERS

Reference	Domain	Information Model	Communication Model	Achieved Goals
[4]	Automation control	RDF, RDFS, OWL	OPC-UA	Present the initial concept of Semantic I4.0 Component.
[12]	Industry sensors	RDF, D2RQ, SPARQL	Semantic Sensor Network (SSN)	Translate the IEC 62264 standard into RDF.
[13]	Automation control	RDF, SHACL	RMLMapper	Map the latest AAS data models into RDF.
[14]	Production robotics	Object Memory Model (OMM)	Central remote maintenance platform	Align AAS guidelines with W3C specifications.
[15]	Engineering systems	OWL	AutomationML	Transform AutomationML data to OWL.
[22]	Complex product modelling	Ontology	-	Model product information in different abstraction levels.
[23]	Complex product modelling	Ontology	-	Incorporate international standards in product modelling.
[24]	Smart manufacturing	Ontology	-	Draft a common semantic net for manufacturing.
[25]	Process modelling	PSL	-	Use PSL to enable the exchange of process information.
[26], [27]	Smart manufacturing	Ontology, logic rules	-	Enable ontology-based predictive maintenance.
[28]	Smart manufacturing	Ontology, JADE framework	-	Handle heterogeneous manufacturing data.
[34]	Smart manufacturing	RDF, Standards Ontology (STO)	OPC-UA, AutomationML	Facilitate the structuring, selection, and integration of standards.
[35], [36]	Smart manufacturing	OWL-S, SAWSDL	OPC-UA, GeSCo	Add flexibility in generating production plans.
[39]	Semantic sensor web	SPITFIRE	Linked Open Data (LOD) cloud	Link semantic sensor models to the cloud.
[40]	Semantic web of things	Ontology	ubiquitous Knowledge Base (u-KB)	Link semantic web to a mobile communication network.
[41]	Semantic web of things	Ontology	IPSO, ZigBee, OMA, oneM2M	Pave a way for developing a semantic layer for the IoT.

TABLE II  
APPLICATION CASE OBJECT AND PURPOSE

Reference	Physical Object	Application Purpose
[4]	Festo AG Motor Controller	Semantically describe an I4.0 component and some of its basic relations.
[12]	Industry Sensors in Legacy Systems	Display sensors on a global interactive map based on time stamps.
[13]	Raspberry Pi, Electronic & Multi-protocol Controllers	Map AASs into Semantic AASs.
[14]	Robot Arm	Simulate the adaptation and remote maintenance of a production robot.
[15]	KUKA Robot	Adopt OCEL and CELOE learning algorithms for concept learning.
[16]	Embedded Micro Reasoner	Detect overflow of water in a tank using sensors.
[36]	Shop Floor Production Machines	Welding, color spraying and quality testing processes on shop floor.
[39]	Sensors	Connect, search and display sensors to the web & internet.

the target concept. This was verified when their most simple test case in OCEL was the most computationally intensive.

Thuluva et al. [16] used an embedded micro reasoner to demonstrate the FESTO process on an automation workstation. The micro reasoner ran on resource-constrained devices with Unix/Linux OS and consisted of two main parts: a micro event processing engine and a datalog reasoner. The former managed event rules that were accessible over a RESTful API and were directly deployed on the automation system devices. The latter provided datalog reasoning, and was embedded in the edge device. The edge device itself was embedded on an automation system and acted as a gateway between the automation system and the cloud. The author's demo included a binary float sensor that included liquid level values on a tank to ensure the tank did not overflow.

Table II summarises our survey results including the purpose for each application, as well as the physical objects used for each use case.

## VI. CONCLUSION AND OPEN CHALLENGES

The Plattform Industrie group developed a reference architecture model named RAMI 4.0 to tackle the ongoing questions of Industry 4.0. This model introduces a new key technology known as the AAS that is the key enabler for providing a unified way of storing and communicating information between components. From our survey, we have discovered that there is plenty of ongoing research looking at the RAMI 4.0 model but only some containing implementations of AASs.

It is clear that there is a lack of formal standardization when implementing AAS as many papers use different standards and technologies for their AAS implementations. Furthermore, a key fundamental of Industry 4.0 that needs to be addressed is the semantic interoperability between machines. Many of the AAS implementations display ways of storing and communicating data between industrial devices but these devices lack to demonstrate the ability to *understand* the data itself. Thus, the direction towards Semantic-based AAS with the use of RDF, OWL and other semantic approaches have been investigated to achieve greater interoperability. In this paper, we have demonstrated existing research contributions regarding Semantic AASs, under the framework of RAMI 4.0. The main characteristics, functionalities, frameworks, and use cases have been covered in this survey.

The survey results expose three main open challenges. The first challenge is the lack of standardised information models for Semantic AASs. Most of the reviewed papers proposed fragmented frameworks or models with a focus on a specific sub-domain of manufacturing, while only a few of them addressed the standardisation issue at a high level. For future works, standardised information models need to be developed with referring to formal ontologies and international standards. The second issue is the need for statistical data processing capabilities for Semantic AASs. As semantic technologies mainly use logic formalisms to describe a certain domain of interest, they have an inherent weakness in processing numeric data. To address this issue, statistical methods such as machine

learning and big data technologies are required to equip Semantic AASs with strong mathematical data processing functionalities. The third open challenge is the lack of human behaviour modelling within Semantic AASs. Although fundamental machine operations can be executed autonomously without human intervention, many decisions for complex tasks need the involvement of humans. However, only a few of the existing research works have considered human behaviour within a decision making cycle. In the future, data from social perspectives need to be considered to improve the decision making of manufacturing systems.

## REFERENCES

- [1] D. Horvath and R. Szabo, "Driving forces and barriers of industry 4.0: Do multinational and small and medium-sized companies have equal opportunities?" *Technological Forecasting and Social Change*, vol. 146, pp. 119–132, 06 2019.
- [2] P. Adolphs, H. Bedenbender, D. Dirzus, M. Ehlich, U. Epple, M. Hankel, R. Heidel, M. Hoffmeister, H. Huhle, B. Karcher, H. Koziolok, R. Pichler, S. Pollmeier, F. Schewe, A. Walter, B. Waser, and M. Wollschlaeger, "Reference architecture model industrie 4.0 (rami4.0)," zvei and vdi, status report, 2015," 2015.
- [3] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart factory of industry 4.0: Key technologies, application case, and challenges," *IEEE Access*, vol. 6, pp. 6505–6519, 2017.
- [4] I. Grangel-González, L. Halilaj, G. Coskun, S. Auer, D. Collarana, and M. Hoffmeister, "Towards a semantic administrative shell for industry 4.0 components," in *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*. IEEE, 2016, pp. 230–237.
- [5] F. Zezulka, P. Marcon, I. Vesely, and O. Sajdl, "Industry 4.0 – an introduction in the phenomenon," *IFAC-PapersOnLine*, vol. 49, no. 25, pp. 8 – 12, 2016, 14th IFAC Conference on Programmable Devices and Embedded Systems PDES 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2405896316326386>
- [6] J. Fuchs, J. Schmidt, J. Franke, K. Rehman, M. Sauer, and S. Karnouskos, "I4.0-compliant integration of assets utilizing the asset administration shell," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019, pp. 1243–1247.
- [7] B. S. I. (BSI), *Smart manufacturing — Reference architecture model industry 4.0 (RAMI4.0)*. BSI Standards Limited 2017, 2017.
- [8] G. Electrical and E. M. Association, "Examples of the asset administration shell for industrie 4.0 components – basic part," 2017.
- [9] P. I. 4.0, "Details of the asset administration shell from idea to implementation," 2018.
- [10] C. Wagner, J. Grothoff, U. Epple, R. Drath, S. Malakuti, S. Grüner, M. Hoffmeister, and P. Zimmermann, "The role of the industry 4.0 asset administration shell and the digital twin during the life cycle of a plant," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2017, pp. 1–8.
- [11] S. Decker, S. Melnik, F. Harmelen, D. Fensel, M. Klein, M. Erdmann, and I. Horrocks, "The semantic web: the roles of xml and rdf," *IEEE Internet Computing*, vol. 4, 10 2000.
- [12] I. Grangel-González, L. Halilaj, S. Auer, S. Lohmann, C. Lange, and D. Collarana, "An rdf-based approach for implementing industry 4.0 components with administration shells," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2016, pp. 1–8.
- [13] S. Bader and M. Maleshkova, *The Semantic Asset Administration Shell*, 11 2019, pp. 159–174.
- [14] E. Tantik and R. Anderl, "Integrated data model and structure for the asset administration shell in industrie 4.0," *Procedia CIRP*, vol. 60, pp. 86–91, 12 2017.
- [15] Y. Hua and B. Hein, "Concept learning in automationml with formal semantics and inductive logic programming," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, 2018, pp. 1542–1547.
- [16] A. S. Thuluva, D. Anicic, and S. Rudolph, "Semantic web of things for industry 4.0," *Semantic Web*, vol. 11, p. 883, 2017.
- [17] T. Gruber, "Ontology," 2018.
- [18] Q. Cao, C. Zanni-Merk, and C. Reich, "Ontologies for manufacturing process modeling: A survey," in *International Conference on Sustainable Design and Manufacturing*. Springer, 2018, pp. 61–70.
- [19] A. Gómez-Pérez and O. Corcho, "Ontology languages for the semantic web," *IEEE Intelligent systems*, vol. 17, no. 1, pp. 54–60, 2002.
- [20] A. Gangemi, "Ontology design patterns for semantic web content," in *International semantic web conference*. Springer, 2005, pp. 262–276.
- [21] Q. Cao, A. Samet, C. Zanni-Merk, F. d. B. de Beuvron, and C. Reich, "An ontology-based approach for failure classification in predictive maintenance using fuzzy c-means and swrl rules," *Procedia Computer Science*, vol. 159, pp. 630–639, 2019.
- [22] M. Vegetti, G. P. Henning, and H. P. Leone, "Product ontology: definition of an ontology for the complex product modelling domain," in *Proceedings of the Mercosur Congress on Process Systems Engineering*, 2005.
- [23] H. Panetto, M. Dassisti, and A. Tursi, "Onto-pdm: Product-driven ontology for product data management interoperability within manufacturing process environment," *Advanced Engineering Informatics*, vol. 26, no. 2, pp. 334–348, 2012.
- [24] S. Lemaignan, A. Siadat, J.-Y. Dantan, and A. Semenenko, "Mason: A proposal for an ontology of manufacturing domain," in *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06)*. IEEE, 2006, pp. 195–200.
- [25] M. Grüninger, "Using the psl ontology," in *Handbook on Ontologies*. Springer, 2009, pp. 423–443.
- [26] Q. Cao, F. Giustozzi, C. Zanni-Merk, F. de Bertrand de Beuvron, and C. Reich, "Smart condition monitoring for industry 4.0 manufacturing processes: An ontology-based approach," *Cybernetics and Systems*, vol. 50, no. 2, pp. 82–96, 2019.
- [27] Q. Cao, A. Samet, C. Zanni-Merk, F. de Bertrand de Beuvron, and C. Reich, "Combining chronicle mining and semantics for predictive maintenance in manufacturing processes," *Semantic Web*, no. Preprint, pp. 1–22, 2020.
- [28] S. Borgo and P. Leitão, "Foundations for a core ontology of manufacturing," in *Ontologies*. Springer, 2007, pp. 751–775.
- [29] Z. Platform Industrie 4.0, "Details of the asset administration shell - part 1. version 2.0," 2019.
- [30] X. Ye and S. Hong, "An automationml/opc ua-based industry 4.0 solution for a manufacturing system," 09 2018, pp. 543–550.
- [31] S. Grunner, J. Pfrommer, and F. Palm, "Restful industrial communication with OPC UA," in *IEEE Transactions on Industrial Informatics*, 2016, vol. 12, no. 5, pp. 1832–1841.
- [32] H. Haskamp, M. Meyer, R. Möllmann, F. Orth, and A. W. Colombo, "Benchmarking of existing opc ua implementations for industrie 4.0-compliant digitalization solutions," in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, 2017, pp. 589–594.
- [33] T. Hannelius, M. Salmenpera, and S. Kuikka, "Roadmap to adopting opc ua," in *2008 6th IEEE International Conference on Industrial Informatics*. IEEE, 2008, pp. 756–761.
- [34] I. Grangel-González, P. Baptista, L. Halilaj, S. Lohmann, M.-E. Vidal, C. Mader, and S. Auer, "The industry 4.0 standards landscape from a semantic integration perspective," 09 2017.
- [35] B. Katti, C. Plociennik, and M. Schweitzer, "Semopc-ua: Introducing semantics to opc-ua application specific methods," 06 2018.
- [36] B. Katti, C. Plociennik, M. Schweitzer, and M. Ruskowski, "Sa-opc-ua: Introducing semantics to opc-ua application methods," 08 2018.
- [37] F. Scioscia and M. Ruta, "Building a semantic web of things: issues and perspectives in information compression," in *2009 IEEE International Conference on Semantic Computing*. IEEE, 2009, pp. 589–594.
- [38] Q. Cao, C. Zanni-Merk, A. Samet, F. d. B. de Beuvron, and C. Reich, "Using rule quality measures for rule base refinement in knowledge-based predictive maintenance systems," *Cybernetics and Systems*, vol. 51, no. 2, pp. 161–176, 2020.
- [39] D. Pfisterer, K. Romer, D. Bimschas, O. Kleine, R. Mietz, C. Truong, H. Hasemann, A. Kröller, M. Pagel, M. Hauswirth et al., "Spitfire: toward a semantic web of things," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 40–48, 2011.
- [40] M. Ruta, F. Scioscia, and E. Di Sciascio, "Enabling the semantic web of things: framework and architecture," in *2012 IEEE Sixth International Conference on Semantic Computing*. IEEE, 2012, pp. 345–347.
- [41] A. J. Jara, A. C. Olivieri, Y. Bocchi, M. Jung, W. Kastner, and A. F. Skarmeta, "Semantic web of things: an analysis of the application semantics for the iot moving towards the iot convergence," *International Journal of Web and Grid Services*, vol. 10, no. 2-3, pp. 244–272, 2014.