An RDF-based Approach for Implementing Industry 4.0 Components with Administration Shells

Irlán Grangel-González, Lavdim Halilaj, Sören Auer, Steffen Lohmann, Christoph Lange, Diego Collarana
Enterprise Information Systems, University of Bonn, Germany
{grangel,halilaj,auer,langec,collaran}@cs.uni-bonn.de
Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS)
{irlan.grangel.gonzalez,lavdim.halilaj,soeren.auer,steffen.lohmann,christoph.lange-bever,diego.collarana.vargas}@iais.fraunhofer.de

***Abstract*—Industry 4.0 is a global endeavor of automation and data exchange to create smart factories maximizing production capabilities and allowing for new business models. The Reference Architecture Model for Industry 4.0 (RAMI 4.0) describes the core aspects of Industry 4.0 and defines Administration Shells as digital representations of Industry 4.0 components. In this paper, we present an approach to model and implement Industry 4.0 components with the Resource Description Framework (RDF). The approach addresses the challenges of interoperable communication and machine comprehension in Industry 4.0 settings using semantic technologies. We show how related standards and vocabularies, such as IEC 62264, eCl@ss, and the Ontology of Units of Measure (OM), can be utilized along with the RDF-based representation of the RAMI 4.0 concepts. Finally, we demonstrate the applicability and benefits of the approach using an example from a real-world use case.**

I. Introduction

Industry 4.0 is a global endeavor leveraging the development of smart factories based on fully computerized, software-driven automation of production processes as well as the integration of software components. In smart factories, software systems monitor and control physical processes, effectively cooperate with each other and with humans, and make decentralized decisions. In order to realize this Industry 4.0 (I4.0) vision, a variety of areas related to manufacturing, security, and machine communication, among others, need to interoperate and align their respective information models.

As a response to these requirements, the *Reference Architecture Model for Industry 4.0 (RAMI 4.0)* and the concept of the *Administration Shell (AdminShell)* have been devised [1]. The Administration Shell is intended to provide a digital representation of all information being available about and from an object, which can be a hardware system or a software component.

In [2], we presented a first version of our approach to implement the RAMI 4.0 model and the Administration Shell concept within an Industry 4.0 context. The approach uses the data exchange standard *Resource Description Framework (RDF)* and builds upon a representation of the RAMI 4.0 model as an RDF *vocabulary*. Furthermore, we identified six challenges for Industry 4.0—interoperability, globally unique identification, data availability, standards compliance, integration, multilinguality—and pointed out how RDF can be utilized to solve them. However, that first version missed an alignment of our *rami* vocabulary with the hierarchy levels specified by the RAMI 4.0 model, which are based on the IEC 62264 standard. Furthermore, crucial concepts, such as units of measurement, provenance information, and a product catalog, were not yet included in our first version of the *rami* vocabulary.

In this paper, we extend our initial work towards a Semantic Administration Shell with the following contributions:

- A significant extension of the *rami* vocabulary[1] covering sensor data, units of measurement as well as product and provenance information.
- An RDF-based vocabulary representing the *IEC 62264* standard[2] and its alignment with the *rami* vocabulary.
- A real-world use case showing the applicability and benefits of implementing Industry 4.0 components with the proposed RDF-based approach.

The remainder of this paper is structured as follows: First, we provide essential background information and terminology in Section II. In Section III, we describe our RDF-based approach of implementing Industry 4.0 components utilizing the concept of Administration Shells. A concrete example showing the benefits of our approach in a real world use case is given in Section IV. Section V provides an overview of related work, before we conclude the paper in Section VI.

II. Background

This section introduces terms and concepts that are relevant to our approach.

A. Reference Architecture Model for Industry 4.0 (RAMI 4.0)

The Reference Architecture Model for Industry 4.0 (RAMI 4.0) encompasses the core aspects of Industry 4.0 in a three-dimensional layer model [1], [3]. It illustrates the connection between IT, manufacturers/plants and the product life cycle in a three-dimensional space. Each dimension shows a particular part of these domains divided into different layers, as depicted in Figure 1. The model extends the hierarchy levels

[1] http://w3id.org/i40/rami/
[2] http://w3id.org/i40/iec/62264/

– Preprint of a paper accepted for presentation at ETFA 2016 in Berlin –

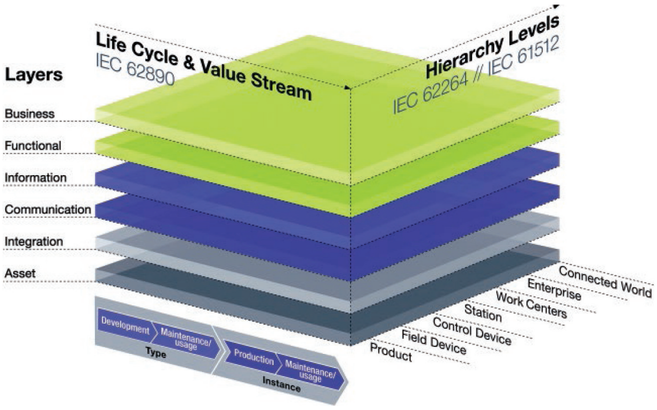

Fig. 1: Reference Architecture Model for Industry 4.0 (RAMI 4.0), comprising the three dimensions *layers*, *life cycle* and *hierarchy levels* (taken from [1]).

defined in *IEC 62264/61512* by adding the concepts *Product* on the lowest level and *Connected World* at the top level, which goes beyond the boundaries of an individual factory.

The vertical axis on the left hand side of Figure 1 represents the IT perspective, comprising layers ranging from the physical device (*asset*) to complex functions as they are available in ERP systems (*functional*). These layers correspond to the IT way of thinking, where complex projects are decomposed into smaller manageable parts. The horizontal axis on the left hand side indicates the product life cycle where *Type* and *Instance* are distinguished as two main concepts. The RAMI 4.0 model enables the representation of data gathered during the entire life cycle. The horizontal axis on the right hand side organizes the locations of the functionalities and responsibilities in a hierarchy.

B. Industry 4.0 Component

A component is a core concept in the Industry 4.0 context. As defined in [1], an I4.0 component constitutes a specific case of a Cyber-Physical System (CPS). It is used as a model to represent the properties of a CPS, for instance, real objects in a production environment connected with virtual objects and processes. An I4.0 component can be a production system, an individual machine, or an assembly inside a machine. It is comprised of two foundational elements: an object and its Administration Shell. Every object or entity that is wrapped by an Administration Shell becomes an I4.0 component, as illustrated in Figure 2. In the following, the different parts of I4.0 components are presented in more detail.

1) Object: In [1], the term *object* is used to refer to an individual physical or non-physical entity. An object can be an entire machine, an automation component, or a software platform; it can be a legacy system or a new system. The industry should be able to integrate and benefit from these objects in I4.0 contexts, independently of their type and age.

2) Administration Shell: The Administration Shell is used to store all important data of an object. Its goal is to create

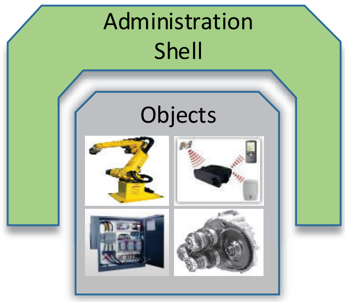

Fig. 2: Industry 4.0 components are objects wrapped by an Administration Shell (adapted from [1]).

benefits for all participants in networked manufacturing [1], including:

a) Data Management: The Administration Shell provides mechanisms to manage large amounts of data and information generated by manufacturers and other stakeholders. For instance, it stores and manages information related to configuration, maintenance, or connectivity with other devices.

b) Functions: Different functions, such as operations, maintenance tasks, or complex algorithms implementing business logic, can be provided by the Administration Shell. These functions facilitate the interaction between the I4.0 component and other actors, including human users.

c) Services: Although the information of a component is stored only once, it can be used beyond the boundaries of the component, within enterprise networks or in a cloud. The information can be made available to different users and can be accessed in various use cases.

d) Integration: The Administration Shell, in combination with communication protocols, offers the possibility of easy integration of I4.0 components.

e) Modularity: Each specific part of an object should be able to store information in the Administration Shell. This ensures that all information is saved and ready to be used for subsequent analysis.

C. Resource Description Framework (RDF)

The Resource Description Framework (RDF) is a semantic data model for interchanging data on the Web recommended by the World Wide Web Consortium (W3C)[3]. In particular, it facilitates the integration of data when the data schemas vary. The flexibility of RDF also allows for the evolution of data and schemas over time, without requiring all data consumers to adapt immediately.

In RDF, information is represented as triples consisting of subjects, predicates and objects, which can be combined to directed graphs composed of vertices (representing subjects and objects) and edges (representing predicates). An example RDF graph representing information about an enterprise is shown in Figure 3. International Resource Identifiers (IRIs) are used to identify resources unambiguously, while *literals*

[3] https://www.w3.org/RDF/

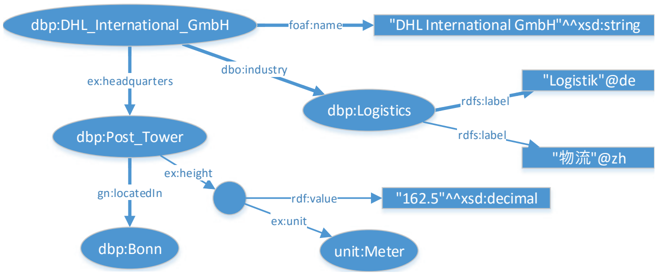

Fig. 3: Example of an RDF graph representing information about an enterprise.

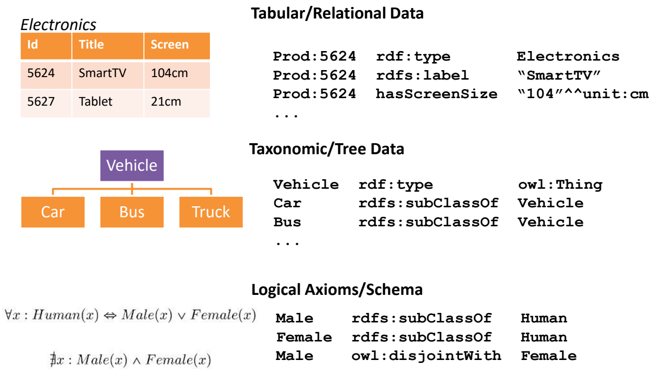

Fig. 4: RDF mediates between different data models and bridges between conceptual and operational layers.

(consisting of either a string and language tag or a value and datatype) describe concrete data values. Formally, an RDF dataset D is defined as a set of triples: $D \subset I \times I \times (I \cup L)$, where I represents the set of IRIs and L the set of literals.[4]

RDF resources (which appear as subjects or objects in triples) are typed by simply adding a triple with the `rdf:type` property as predicate and a suitable object representing the class the resource belongs to (cf. the first triples in the first two examples of Figure 4).

Properties and classes required to describe and structure the data in a certain domain can be defined in RDF themselves. They can be arranged in property or class hierarchies, as in the example representing taxonomic data in Figure 4. Such descriptions of classes and properties are called vocabularies, RDF schemas, or ontologies. On the one hand, they represent a semantic model of a certain domain. On the other hand, they can be directly used to represent and integrate data and to execute queries (e.g., using the W3C-specified SPARQL query language). Thus, RDF bridges between the conceptual and operational levels of information and data representation. Furthermore, RDF can be used to easily represent various types of information and data, including taxonomic/tree data, tabular/relational data, logical axioms, etc. (cf. Figure 4). Since all schema and data entities have IRI identifiers that are world-wide unique, it is easy to link to other data (instance level) or to reuse vocabulary elements from existing vocabularies (schema level).

The flexible data model, the ability to define interlinked domain-specific vocabularies, the world-wide unique identification of entities using IRIs as well as the possibility to represent various types of information make RDF perfectly suited for representing, interlinking and integrating data in enterprise and manufacturing settings. In particular, we propose to employ RDF as the *lingua franca* to represent and integrate information in Industry 4.0 contexts. In the following sections, we discuss this approach in more detail.

III. An RDF-based Approach for Industry 4.0

Semantic technologies play a crucial role with regard to the description and management of things, devices, and services [4], [5]. Moreover, it has been recognized that I4.0 components and their contents should follow a common semantic model [1]. Therefore, we propose an RDF-based approach to pave the way towards a common semantic model for Industry 4.0. Figure 5 depicts the architecture of our RDF-based approach, which extends the Administration Shell idea to enable the integration of different I4.0 components. For representing the hierarchy levels of the RAMI 4.0 model, we created an RDF-based vocabulary conforming to Part 1 of the IEC 62464:2013 standard (cf. subsection III-B).

A. *The* rami *Vocabulary*

Our approach defines a semantic vocabulary for the Administration Shell concept by providing an ontological formalization of the elements that describe I4.0 components. Since the Administration Shell is a key concept of the RAMI 4.0 model, we decided to use the namespace *rami* for the vocabulary—also, since the vocabulary implements further concepts of the RAMI 4.0 model.

The core classes of the vocabulary are `rami:BasicData`, `rami:AdminShell` and `rami:Object`. The class `rami:AdminShell` represents the Administration Shell concept and its properties. The objects in the RAMI 4.0 model are described by the `rami:Object` class. In addition, properties like `rami:name`, `rami:isPartOf` and `rami:description` are created to represent the characteristics and features of the object. The basic data associated with the object are represented by the `rami:BasicData` class. This allows to add different types of data, such as sensor, mechanical, electrical, or physical data, as subclasses to the basic data class.

Further, it permits to incorporate existing models, such as the *Object Memory Model* (OMM), which supports the creation of digital memories for manufacturing objects [6]. OMM provides *blocks* for grouping data on a certain aspect of an object. These blocks contain metadata for describing the object, for instance, its ID, description, or format. In the *rami* vocabulary, we included some of the OMM concepts, such as identification and type description, and organize them in the `rami:BasicData` class. In this way, different types of data associated with the object, for instance,

[4]For simplicity, we omit the consideration of blank nodes here.

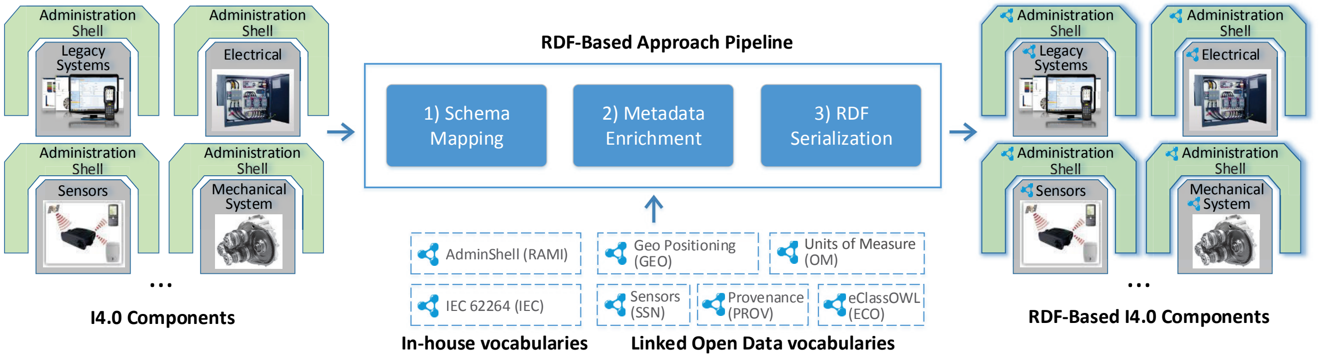

Fig. 5: Architecture of the proposed RDF-based I4.0 components comprising vocabularies and RDF representations of relevant standards for representing information about a wide range of components.

`rami:EngineeringData`, inherit attributes that have been defined for `rami:BasicData`. Additionally, they inherit attributes specifically defined for `rami:EngineeringData`, in this case, `standard name`, `version`, etc. The `rami:AdminShell` class is used to connect the object with its basic data. Figure 6 depicts the main classes and properties of the *rami* vocabulary.

In order to realize the Industry 4.0 vision, it is crucial to allow the integration of existing standards that already specify certain aspects and are used in industrial contexts. We have therefore created the class `rami:Standard` in our vocabulary in accordance to [3]. In addition, we have added instances describing existing standards, such as `rami:IEC61784` for communication, `rami:IEC61360` for engineering, and `rami:IEC61508` for safety. Following this approach allows to connect the object with the standard that describes it via the aforementioned Administration Shell concept. If further standards need to be considered, they can be easily added as instances of the `rami:Standard` class in the same way.

In manufacturing processes, provenance is of great importance [7]. For instance, authenticating a specific product with regard to its manufacturer, the date it was manufactured, etc., are critical information to record within the manufacturing context. For this reason, we reused the W3C *Provenance Ontology*[5] to track the creator and contributors of an object in the *rami* vocabulary.

With the goal of aligning the RAMI 4.0 model with the IEC 62264 hierarchy levels, we defined the class `rami:RAMIHierarchyLevel`. Instances of this class represent the RAMI 4.0 hierarchy levels (`rami:Station`, `rami:WorkCenter`, etc.). This allows to link concepts, such as the IEC 62264 *Storage Unit*, which is a type of *Work Center*, as shown in Listing 1.[6]

Listing 1: Alignment of RAMI 4.0 and IEC 62264 concepts

```
@prefix iec62264: <https://w3id.org/i40/iec/62264/> .
@prefix rami:     <https://w3id.org/i40/rami/> .

iec62264:StorageUnit  rami:RAMIHierarchyLevel  rami:WorkCenter .
```

[5]https://www.w3.org/TR/prov-o/

[6]The listings use the Turtle syntax for serializing RDF graphs as triples.

Listing 2: Representing lengths using the OM ontology

```
@prefix om:   <http://www.wurvoc.org/vocabularies/om-1.8/> .
@prefix rami: <http://w3id.org/i40/rami/> .
@prefix eco:  <http://www.ebusiness-unibw.org/ontologies/eclass/5.1.4/#> .
@prefix ex:   <http://example.org/data/> .

ex:object1      eco:P_BAA018001      ex:lengthOfObj1 .
ex:lengthOfObj1 om:numerical_value   "42.72" .
ex:lengthOfObj1 om:units_of_measure_or_measurement_scale
                                     om:centimetre .
rami:object2    eco:P_BAA018001      ex:lengthOfObj2 .
ex:lengthOfObj2 om:numerical_value   "18" .
ex:lengthOfObj2 om:units_of_measure_or_measurement_scale
                                     om:inch-international .
```

Also, units of measurement are of paramount importance in manufacturing environments for the correct function and coordination of processes. Units are required for the specification of products as well as for representing the data produced by measuring devices (e.g. sensors). Often, units are represented as simple strings, e.g., °C, mm, kg, etc. This has the drawback that the semantics of the units are not machine-readable and sometimes unknown or ambiguous. For example, both “18 in” and “45,72 cm” are referring to the same length.

For properly representing units, we aligned the *rami* vocabulary with the *Ontology of Units of Measure (OM)* [8][7]. This ontology provides global identifications and definitions for units of measurement, including quantities, measurements, and dimensions. By using the *in*[8] and *cm*[9] concepts from the OM ontology, the semantics of the units can be *understood* by a machine because their formal definitions can be looked up in the ontology via the IRIs of the concepts as well as processed and interpreted by software. For example, “centimetre” is defined as a unit in the dimension of length, amounting to 1/100 of the SI unit “metre”. Listing 2 illustrates how data values can be represented using the OM ontology.

Moreover, we considered the alignment with eCl@ss [9]. eCl@ss is a cross-industry classification system to describe products and services using unique identifiers. In the context of Industry 4.0, eCl@ss performs a crucial function by providing

[7]http://www.wurvoc.org/vocabularies/om-1.8/

[8]http://www.wurvoc.org/vocabularies/om-1.8/inch-international

[9]http://www.wurvoc.org/vocabularies/om-1.8/centimetre

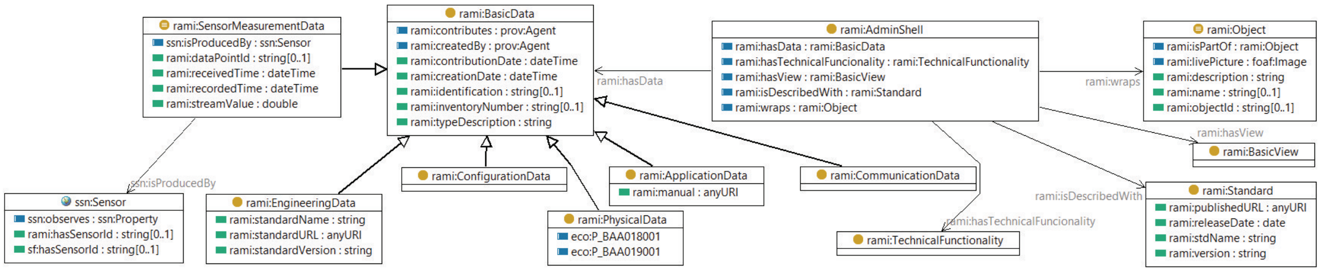

Fig. 6: Overview of the core classes and relationships of the *rami* vocabulary.

common definitions of a vast amount of products and services. eCl@ss is available as an RDF-based vocabulary[10] and can therefore be easily reused and aligned with our approach. To describe units of measurement, the eCl@ss vocabulary incorporates the *GoodRelations* vocabulary[11]. Since the OM ontology contains more specialized and rich descriptions for units, we propose to use both (i.e., the eCl@ss and *GoodRelations* vocabularies) jointly. Based on this, we recommend to align the eCl@ss concepts to our definitions in the *rami* vocabulary.

B. IEC 62264 Vocabulary

The RAMI 4.0 model builds on the IEC 62264 standard to define hierarchy levels for the manufacturing domain. Next to the hierarchy levels, this standard specifies core concepts for the development of manufacturing companies, such as work centers, production lines, and storage zones. Based on these definitions, we developed an RDF-based vocabulary that models the structure as well as the concrete semantics of these concepts. Figure 7 depicts the core classes and properties of this IEC 62264 vocabulary.

Our RDF-based approach allows to align the information models of different companies with the proposed standard. For example, the term *Plant* is commonly used in the manufacturing world. The meaning of this term is equivalent to the term *Site* according to the standard. Instead of changing the meaning, an alignment of the terms can be expressed as shown in Listing 3.

Following the same idea, also other cases, i.e., expressing that one concept is broader or narrower than some other, can be addressed by reusing specialized vocabularies, such as the Simple Knowledge Organization System (SKOS)[12].

Listing 3: Alignment of concepts

```
@prefix owl:        <http://www.w3.org/2002/07/owl#>.
@prefix enterprise: <http://enterprise.com/vocabularies/>.
@prefix iec62264:   <https://w3id.org/i40/iec/62264/> .

enterprise:Plant  a                    owl:Class .
enterprise:Plant  owl:equivalentClass  iec62264:Site .
```

[10] http://www.heppnetz.de/projects/eclassowl/
[11] http://www.heppnetz.de/projects/goodrelations/
[12] http://www.w3.org/TR/skos-reference

IV. Use Case

The vision of Industry 4.0 is centered around the concept of decentralized production and smart objects that participate in the production in terms of autonomy and decision-making. To accomplish this goal, object metadata, data, and relations with other objects need to be semantically described with the *rami* vocabulary. By doing so, the information provided by one object can be understood and exploited by other smart objects in the production chain. To illustrate the applicability of our approach, we detail a use case in this section, where our *rami* vocabulary is used to describe the data of a legacy system and some of its basic relations.

For the following scenario, we used the *AirProbe* dataset [10] which is provided as an SQL dump. It contains information about sensors, their geospatial locations and measurements of black carbon concentrations, temperature, and humidity. Such types of data can also be found in industry contexts, for instance, if sensors are installed in a factory, machine, or carrier. Sensor data is covered by our *rami* vocabulary by subsuming the basic data concept. Furthermore, and following best practices in vocabulary design, we aligned our description of the sensor data with the *Semantic Sensor Network (SSN)* ontology[13].

A necessary step to support the mapping of relational data to the *rami* vocabulary was the use of the W3C R2RML mapping language with the mapping tool D2RQ [11][14]. First, a mapping file was created between the dataset and the vocabulary. Table I shows the mapping from the main columns of the AirProbe dataset table to the concepts defined in our *rami* vocabulary.

We used D2RQ on top of a MySQL server to make the dataset accessible as RDF. D2RQ acts as a middle layer between the SQL-based data and the SPARQL queries. As a result, it is possible to perform queries and to receive real-time information about particular events. In this way, the data in a legacy system can be used and exploited by other, RDF-aware software agents without the need to transform all of them into RDF following an ETL (extract-transform-load) approach, which is expensive if the data source is updated frequently.

[13] https://www.w3.org/2005/Incubator/ssn/ssnx/ssn
[14] http://d2rq.org

Fig. 7: Overview of the core classes and relationships of the IEC 62264 vocabulary.

TABLE I: Mapping of database columns and concepts.

Column from AirProbe DB	*Vocabulary concept*
meta_device_id	rami:hasSensorId
meta_timestamp_recorded	rami:recordedTime
meta_timestamp_received	rami:receivedTime
geo_lat	geo:lat
geo_lon	geo:long
…	…
data_temp_1	om:Temperature
data_hum_1	om:Relative_humidity

Listing 4 shows the query used to obtain the measured temperature for a specific time interval. The result of the query is depicted in Figure 8.

Listing 4: Querying temperature in a specific time interval.

```
PREFIX xsd:  <http://www.w3.org/2001/XMLSchema#>
PREFIX om:   <http://www.wurvoc.org/vocabularies/om-1.8/>
PREFIX rami: <http://w3id.org/i40/rami/>

SELECT ?time ?value WHERE {
  ?measurement  rami:recordedTime  ?time ;
                om:Temperature     ?value .
   FILTER (xsd:dateTime(?time) >= "2015-01-29T10:00:00Z"^^xsd:dateTime)
   FILTER (xsd:dateTime(?time) <= "2015-01-29T11:00:00Z"^^xsd:dateTime)
}
ORDER BY DESC(?value) LIMIT 20
```

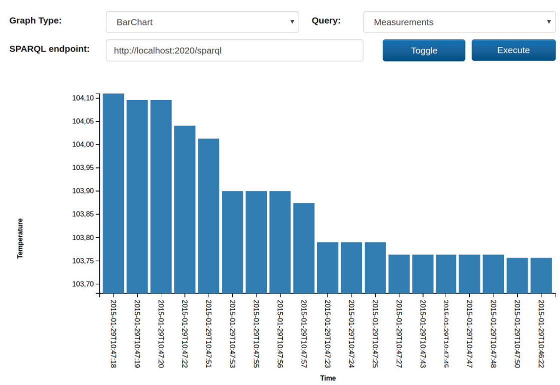

Fig. 8: Temperatures measured in a given time interval.

The query defined in Listing 5 returns all geospatial information about those sensors that transmitted data in a particular interval of time.

Figure 9 shows these geographical coordinates on an interactive map, where the user is able to navigate and obtain more information about the sensors.

One of the main advantages of the *rami* vocabulary is the uniform data representation according to the RDF model, which enables efficient integration and querying of the data comprised in the Administration Shell. Listing 4 and Listing 5 provide exemplary SPARQL queries that illustrate the uniform and integrated data retrieval possible with our approach.

The use case points out how the *rami* vocabulary enables a flexible semantic representation of data, which helps to overcome the challenges related to the integration of heterogeneous data sources that I4.0 is facing.

Listing 5: Querying sensors active in a given time interval.

```
PREFIX om:   <http://www.wurvoc.org/vocabularies/om-1.8/>
PREFIX rami: <http://w3id.org/i40/rami/>
PREFIX geo:  <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX ssn:  <http://purl.oclc.org/NET/ssnx/ssn#>

SELECT ?sensorID ?lat ?lng
WHERE {
  ?measurement  a                  rami:SensorMeasurementData .
                rami:recordedTime  ?time ;
                geo:lat            ?lat ;
                geo:long           ?lng ;
                ssn:isProducedBy   ?sensor ;
   ?sensor      hasSensorId        ?sensorID .
   FILTER (xsd:dateTime(?time) >= "2015-01-29T10:00:00Z"^^xsd:dateTime)
   FILTER (xsd:dateTime(?time) <= "2015-01-29T11:00:00Z"^^xsd:dateTime)
}
```

V. Related Work

There are several works investigating the use of ontologies for representing manufacturing standards, also with regard to IEC 62264 [15], [16], [17]. However, these approaches lack an adequate design to permit the reuse and integration of other vocabularies, and they have not been published according to the Linked Data [18] principles. Some benefits of using Linked Data to integrate industrial engineering data are described in [19].

In this paper, we addressed the heterogeneity of industrial data and showed how to use semantic technologies, i.e., how

TABLE II: Comparison with related I4.0 component description approaches.

Approach	*Basic Concept*	*Identification*	*Data model*	*Organization Type*	*Serialization*
EDDL [12]	Device	n/a	Object	n/a	Text
OMM [6]	Physical artifact	Primary ID and IDs for blocks	Element	Hierarchical	XML
DOMe [7]	Object	Primary ID and IDs for blocks	Object	Hierarchical	XML
PML [13]	Physical object	XML tag ID	Object	Hierarchical	XML
SPDO [14]	Product	URI/IRI	Resource	Hierarchical	OWL-DL
Our RDF-Based Approach	Administration Shell/Object	URI/IRI	Resource	Hierarchical	RDF, RDF Schema, OWL

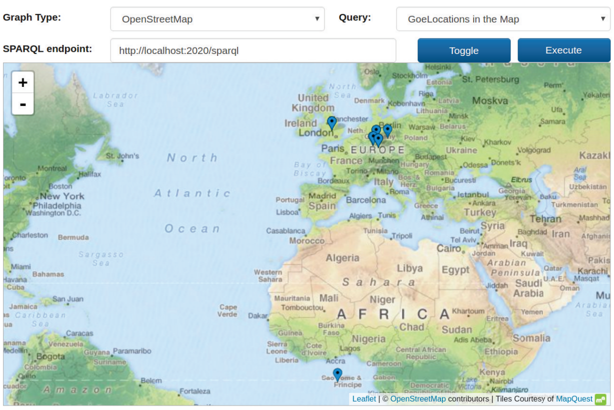

Fig. 9: Geolocations of the queried sensors.

to utilize vocabularies and query languages like SPARQL, to overcome this issue. So far, no attention has been paid to the development of a core vocabulary that can serve as a central hub and be linked with other vocabularies in the domain of industrial engineering.

Cheng et al. [20] present guidelines to properly choose the level of semantic formalization for representing different types of Industry 4.0 projects. The crucial role of semantic technologies for mass customization is discussed in [5]. The work emphasizes that semantic technologies can serve as a glue to connect smart products, data, and services.

Table II provides a comparison of our approach with related I4.0 component description attempts. The *Electronic Device Description Language* (EDDL) is a language to describe information related to digital components [12], [21]. EDDL is available for a large number of devices that are currently utilized in the process industry. EDDL provides a text-based description of devices and their properties, describing the data and how they should be displayed.

The *Object Memory Model* (OMM) is an XML-based format that allows for modeling information about individual physical artifacts [6]. The memory is partitioned into blocks to enable various actors to read and write different aspects of information about an artifact. The conceptual approach in that work is to bring a semantic layer to the physical components, but its implementation suffers from the syntactic limitations of XML. However, it is envisioned that blocks of an OMM contain RDF and OWL payload data.

Extending the concept of OMM, *Domeman* [22] is a framework for the representation, management, and utilization of digital object memories. The idea of using semantic descriptions of physical artifacts by combining OMM and a server realization has been proposed by Haupert and Schneide [23]. They developed an Object Memory Server as an index server for product memories, based on the same set of metadata as the block format. However, this approach is focused on the identification of artifacts and still exposes the OMM limitations mentioned above.

A similar approach is proposed with *DOMe* in [7]. DOMe is a Digital Object Memory which allows automated interaction between workpieces and machine tools using an RFID-based smart environment. It also relies on the metadata proposed by the OMM approach to describe the manufacturing object. The application of ontologies is considered for representing rules of the manufacturing domain. However, the semantic description of the object itself, and the various types of data that exist in the manufacturing domain, are not addressed.

The *Physical Markup Language* (PML) is a common language for describing physical objects, processes and environments [13]. The goal of PML is to use these descriptions in remote monitoring and control of the physical environment.

Janzen and Maass [14] define smart products as a connection of physical products and information goods that allow the embedding of digital product information into physical products. They present the *Smart Product Description Object* (SPDO), a data model built on top of the *DOLCE* ontology [24] for describing smart products.

Bergweiler [25] defines an approach for distinguishing local and global data structures stored in Active Digital Object Memories (ADOMe), to extend so-called smart labels with memory and processing capabilities. According to the author, this can be realized by storing the data in a unified structured format.

VI. Conclusion and Future Work

An RDF-based approach plays an important role for the realization of the Industry 4.0 vision by means of the RAMI 4.0 model. To this end, the *rami* vocabulary and the associated Administration Shell concept have been introduced. Further, the hierarchy levels of the RAMI 4.0 model, which are defined by the IEC 62264 standard, have been translated into an RDF-based vocabulary. This permits to provide common descriptions of Industry 4.0 components along with different types of data represented by various standards applied in the domain. We also showed how relevant vocabularies like eCl@ss, the Ontology of Units of Measure (OM), or the Semantic Sensor Network (SSN) ontology can be utilized conjointly with our approach providing a common understanding of the terms relevant in an Industry 4.0 context.

We have demonstrated the applicability of our approach by implementing it in a real-world use case, where we aligned the RAMI vocabulary with sensor data from a legacy system. Discussions about the Industry 4.0 concepts and the RAMI model itself are still ongoing; therefore, many changes may occur. Despite this fact, we believe that the application of RDF paves the way for a concrete utilization of the RAMI and the IEC 62264 as well as the mentioned vocabularies.

We envision this work as an important step of a larger research and development agenda aiming at equipping manufacturing entities with semantics-based means for communication and data exchange. To further our research, we plan to bring more intelligence to the edge of production facilities, thus promoting self-organization and resilience.

Future work will focus on refining and extending the *rami* vocabulary in order to provide support for a wide range of objects and device types. Furthermore, we intend to develop a number of vocabularies representing important standards for Industry 4.0, such as IEC 61360, AutomationML, and OPC UA. Our aim is to integrate the *rami* vocabulary with the vocabularies for those standards in order to create a unified and semantically well-defined standard landscape for Industry 4.0.

Acknowledgments

This work has been supported by the German Ministry for Education and Research (BMBF) funded projects *LUCID* and *Industrial Data Space*, and by the European Commission under the Horizon 2020 Framework Program for the project *bIoTope* (grant no. 688203).

References

[1] P. Adolphs, H. Bedenbender, D. Dirzus, M. Ehlich, U. Epple, M. Hankel, R. Heidel, M. Hoffmeister, H. Huhle, B. Kärcher, H. Koziolek, 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. http://www.zvei.org/Themen/Industrie40/Seiten/Das-Referenzarchitekturmodell-RAMI-40-und-die-Industrie-40-Komponente.aspx

[2] 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 *10th IEEE International Conference on Semantic Computing (ICSC '16)*. IEEE, 2016, pp. 230–237.

[3] P. Adolphs, S. Auer, H. Bedenbender, M. Billmann, M. Hankel, R. Heidel, M. Hoffmeister, H. Huhle, M. Jochem, M. Kiele, G. Koschnick, H. Koziolek, L. Linke, R. Pichler, F. Schewe, K. Schneider, and B. Waser, "Structure of the administration shell. continuation of the development of the reference model for the industrie 4.0 component," ZVEI and VDI, Status Report, 2016. http://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/structure-of-the-administration-shell.html

[4] M. Thoma, T. Braun, C. Magerkurth, and A. Antonescu, "Managing things and services with semantics: A survey," in *2014 IEEE Network Operations and Management Symposium (NOMS '2014)*, 2014, pp. 1–5.

[5] W. Wahlster, "Semantic technologies for mass customization," in *Towards the Internet of Services: The THESEUS Research Program*. Springer, 2014, pp. 3–13.

[6] J. Haupert, M. Seißler, B. Kiesel, B. Schennerlein, S. Horn, D. Schreiber, and R. Barthel, "Object memory modeling," World Wide Web Consortium (W3C), Incubator Group Report, Sep. 2011. [Online]. Available: https://www.w3.org/2005/Incubator/omm/XGR-omm-20111026/

[7] T. Aruväli, W. Maass, and T. Otto, "Digital object memory based monitoring solutions in manufacturing processes," *Procedia Engineering*, vol. 69, pp. 449–458, 2014.

[8] H. Rijgersberg, M. van Assem, and J. L. Top, "Ontology of units of measure and related concepts." *Semantic Web*, vol. 4, no. 1, pp. 3–13, 2013.

[9] eClass e.V., "eCl@ss standardized material and service classification," 2016. [Online]. Available: http://www.eclass.eu/

[10] A. Sîrbu, M. Becker, S. Caminiti, B. De Baets, B. Elen, L. Francis, P. Gravino, A. Hotho, S. Ingarra, V. Loreto, A. Molino, J. Mueller, J. Peters, F. Ricchiuti, F. Saracino, V. D. P. Servedio, G. Stumme, J. Theunis, F. Tria, and J. Van den Bossche, "Participatory patterns in an international air quality monitoring initiative," *PLOS ONE*, vol. 10, no. 8, 2015.

[11] C. Bizer and A. Seaborne, "D2RQ - treating non-RDF databases as virtual RDF graphs," in *Poster Proceedings of ISWC '04*, 2004.

[12] M. Riedl and F. Naumann, *EDDL – Electronic Device Description Language*, 2nd ed. München, Oldenburg Industrieverlag, 2011.

[13] D. L. Brock, T. P. Milne, Y. Y. Kang, and B. Lewis, "The physical markup language," Auto-ID Center, White Paper MIT-AUTOID-WH-003, 2001.

[14] S. Janzen and W. Maass, "Smart product description object (spdo)," in *Poster Proceedings of the 5th International Conference on Formal Ontology in Information Systems (FOIS '08)*, 2008.

[15] M. Dassisti, H. Panetto, A. Tursi, and M. De Nicolo, "Ontology-based model for production-control systems interoperability," in *5th CIRP Digital Enterprise Technology Conference (DET '08)*, 2008, pp. 527–543.

[16] 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.

[17] C. Legat, C. Seitz, S. Lamparter, and S. Feldmann, "Semantics to the shop floor: towards ontology modularization and reuse in the automation domain," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 3444–3449, 2014.

[18] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data–the story so far," *International Journal on Semantic Web and Information Systems*, vol. 5, no. 3, pp. 1–22, 2009.

[19] M. Graube, J. Pfeffer, J. Ziegler, and L. Urbas, "Linked data as integrating technology for industrial data," *International Journal of Distributed Systems and Technologies*, vol. 3, no. 3, pp. 40–52, 2012.

[20] C.-H. Cheng, T. Guelfirat, C. Messinger, J. O. Schmitt, M. Schnelte, and P. Weber, "Semantic degrees for industrie 4.0 engineering: Deciding on the degree of semantic formalization to select appropriate technologies," in *10th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 15)*. ACM, 2015, pp. 1010–1013.

[21] S. Runde, G. Wolf, and M. Braun, "EDDL and semantic web - from field device integration (FDI) to future device management (FDM)," in *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation, (ETFA '13)*. IEEE, 2013, pp. 1–8.

[22] J. Haupert, "DOMeMan: a framework for representation, management, and utilization of digital object memories," in *9th International Conference on Intelligent Environments (IE '13)*. IEEE, 2013, pp. 84–91.

[23] J. Haupert and M. Schneider, "The object memory server for semantic product memories," in *SemProM - Foundations of Semantic Product Memories for the Internet of Things*. Springer, 2013, pp. 175–189.

[24] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider, "Sweetening ontologies with dolce," in *13th International Conference on Knowledge Engineering and Knowledge Management*, ser. EKAW '02. London, UK, UK: Springer-Verlag, 2002, pp. 166–181. [Online]. Available: http://dl.acm.org/citation.cfm?id=645362.650863

[25] S. Bergweiler, "Intelligent manufacturing based on self-monitoring cyber-physical systems," in *9th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, C. B. Westphall, J. L. Mauri, and P. Pochec, Eds. IARIA, 2015, pp. 108–113.