The SemIoE Ontology: A Semantic Model Solution for an IoE-Based Industry

Marco Arazzi, Antonino Nocera, *Member, IEEE*, and Emanuele Storti

***Abstract*—Recently, the Industry 5.0 is gaining attention as a novel paradigm, defining the next concrete steps toward more and more intelligent, green-aware, and user-centric digital systems. In an era in which smart devices typically adopted in the industry domain are more and more sophisticated and autonomous, the Internet of Things and its evolution, known as the Internet of Everything (IoE, for short), involving also people, robots, processes, and data in the network, represent the main driver to allow industries to put the experiences and needs of human beings at the center of their ecosystems. However, due to the extreme heterogeneity of the involved entities, their intrinsic need and capability to cooperate, and the aim to adapt to a dynamic user-centric context, special attention is required for the integration and processing of the data produced by such an IoE. This is the objective of the present paper, in which we propose a novel semantic model that formalizes the fundamental actors, elements and information of an IoE, along with their relationships. In our design, we focus on state-of-the-art design principles, in particular reuse, and abstraction, to build "SemIoE," a lightweight ontology inheriting and extending concepts from well-known and consolidated reference ontologies. The defined semantic layer represents a core data model that can be extended to embrace any modern industrial scenario. It represents the base of an IoE knowledge graph, on the top of which, as an additional contribution, we analyze and define some essential services for an IoE-based industry.**

***Index Terms*—Industry 5.0, Internet of Everything (IoE), Internet of Things (IoT), knowledge graph, ontology.**

I. Introduction

THE INDUSTRY 5.0 paradigm emphasizes the shift from traditional industries to intelligent, people-oriented, and environmentally conscious industrial ecosystems. The concept of Industry 5.0 unravels through many recent research contributions analyzing the evolution of the modern industry and technology integration [13], [22], [29], and has attracted the interest of the European Commission who provided a thorough definition outlining its main characteristics and objectives [10]. According to this definition, the Industry 5.0 builds upon five main points, namely: 1) human-centric point of view; 2) mass customization and optimization; 3) sustainability; 4) resilience; and 5) strong technology integration. According to this perspective, the focus of production processes should always revolve around the experiences and needs of human beings, individualizing product offerings on the customer side. The Internet of Things (IoT) has strengthened the relationship between humans and intelligent devices, and the expansion of this network, known as the Internet of Everything (IoE), integrates devices, processes, data, and people into a single, highly interconnected environment. This technology allows for a smooth transition toward Industry 5.0, as the IoE natively builds a collaborative ecosystem involving people, autonomous devices (such as smart objects, robots, cobot, and so forth), and smart platforms. However, one of the main challenges of an Industry 5.0 scenario is to redefine the working context by putting human resources at the center of the production chain, so that all the services, facilities, data, and tools, available in the industry environment, are dynamically adapted to their needs. Moreover, modern IoT technology, which is part of the more general IoE ecosystem considered in this article, provides access to increasingly autonomous smart devices that can often even act on behalf of their human owners. This leads to a very complex reality to which the industry should adapt, not only to surround human resources easing their access to the industrial environment, but also to allow a smooth shift from direct manual actions and tasks carried out by human resources to automatic actions carried out by suitably delegated smart devices. Such an opportunity implicitly requires a slight paradigm change from human-centric to user-centric, in which users can be either human resources or autonomous smart devices acting on their behalves.

In such a scenario, one of the main problems is how to integrate data produced by intelligent and autonomous devices with data produced by the actions carried out by humans in order to enable self-organization, self-optimization, and self-healing for the whole IoE. In this setting, a model-based strategy may support the integration effort providing fundamental reference information to enable the declination of the above mentioned Industry 5.0, at least from a data management point of view. However, such a strategy should answer the following research question: how to model heterogeneous information from such a complex context in a flexible, modular, and extensible way?

In this article, we embrace the novel scenario introduced above and strive to address the mentioned question by

Manuscript received 23 July 2024; accepted 25 August 2024. Date of publication 2 September 2024; date of current version 6 December 2024. This work was supported in part by the PRIN 2022 Project "HOMEY: a Human-Centric IoE-based Framework for Supporting the Transition Toward Industry 5.0" funded by the European Union—Next Generation EU under Grant 2022NX7WKE and Grant CUP: F53D23004340006. *(Corresponding author: Emanuele Storti.)*

Marco Arazzi and Antonino Nocera are with the Department of Electrical, Computer and Biomedical Engineering, University of Pavia, 27100 Pavia, Italy (e-mail: marco.arazzi01@universitadipavia.it; antonino.nocera@unipv.it).

Emanuele Storti is with the Department of Information Engineering, Polytechnic University of Marche, 60121 Ancona, Italy (e-mail: e.storti@univpm.it).

Digital Object Identifier 10.1109/JIOT.2024.3452945

introducing a user-centric *semantic* model that formalizes and relates the fundamental actors, elements and information available in such an IoE environment for an Industry 5.0 ecosystem. After a thorough analysis of the reference domain, we recognize the following main entities for our model: 1) human resources and smart devices; globally referred to as *agents* of the IoE in the remaining of this article; 2) activities and workflows carried out by IoE agents; 3) relationships and collaboration actions among agents; 4) role and activity-related privileges; and 5) environment elements and physical building parts. On the top of these concepts, a lightweight and extensible ontology is designed following state-of-the-art design principles, in particular reuse, and abstraction. As for this final aspect, the ontology is built by inheriting concepts available in well-supported and acknowledged existing models, such as the semantic sensor network (SOSA/SSN) [11], the building topology ontology (BOT) [31], and the organization ontology (ORG) [12]. The proposed lightweight ontology, named "SemIoE," represents a semantic knowledge core layer providing a reference structure to build an IoE knowledge graph, which includes all the instances of the entities identified above for an Industry 5.0. Due to the extremely dynamic and heterogeneous domain, instead of focusing on specific industry ecosystems, we make an additional effort to generalize and identify the common actors and build a common core model, which can, hence, be extended to model any existing real-life industrial IoE scenario. As an additional important contribution, in our solution we define and include a set of services for the IoE built on top of the aforementioned IoE knowledge graph, namely: 1) IoE access control, managing and securing the access to IoE facilities only to sufficiently privileged agents; 2) IoE collaboration, providing a robust mechanism to enable collaboration among IoE agents toward the completion of specific IoE tasks; 3) IoE secure delegation, enabling the possibility of delegating complex tasks to supporter peers (both humans and smart devices); and 4) IoE Environment Setting, allowing for the adaptation of the surrounding smart environment to favor the activities/needs of IoE agents. Altogether, the semantic knowledge layer, the IoE knowledge graph, and the set of defined IoE services represent the building blocks for an IoE-based Industry 5.0.

The remaining of this article is organized as follows. In Section II, we analyze the related scientific literature. The reference scenario is detailed in Section III. Section IV is devoted to the description of the proposed semantic knowledge layer and the discussion of the proposed ontological model for IoE. In Section V, we describe in details the services for the IoE included in our solution. Section VI, instead, is devoted to analyze possible implementation strategies. Finally, in Section VII we draw our conclusion and discuss possible open points and future developments.

II. Related Work

The novel paradigm of Industry 5.0 identifies the transition from the traditional industry toward smart, eco-aware, and human-centric factories [25], [37]. In this futuristic, but still timely, context, some authors identify the IoT and, more in general, the IoE, as an enabling technology [22]. The IoE focuses on the interconnection between humans, data processes, and things in a single unified ecosystem and is referred to as the evolution of the consolidated concept of the IoT [32]. Due to the heterogeneity of the technologies used in IoT, and the additional complexity of combining data produced by IoT devices with those produced by humans, processes, and complex interaction among them, it is fundamental to define ontologies allowing semantic interoperability between different applications and services [30].

In the literature, a great effort has been devoted to defining semantic models to represent device characteristics and their relationships in the IoT. A major problem faced by the research community in this context is related to the heterogeneity of IoT devices and their continuous evolution over time. As a matter of fact, ontologies in this domain should be general enough to model network of sensors, their capabilities and the applications built on top of them. Moreover, they should also consider that these technologies are extremely mutable, as new devices are released on a daily basis, and that the difference between the applications and services in different industrial domain are, typically, very relevant [34]. One of the earliest approaches in this direction is described in [33]. Here, the authors designed a very general-purpose ontology, called OntoSensor, starting from the Web Ontology Language (OWL) [26] and the Suggested Upper Merged Ontology (SUMO) [27]. Building once again from the SUMO ontology, Eid et al. [15] designed an initial ontology to retrieve all-and-only relevant sensor data, following an evolving prototype life-cycle. According to the approach identified in [19], the building process is split into: common vocabulary collection, initial taxonomy identification, adding restrictions and axioms, checking for consistency, modifications, and evaluation.

At the time of writing, probably the most suitable sensor ontology for IoT is the semantic sensor network (SSN) ontology [11]. Several work built lightweight semantic models on the top of the concepts included in the SSN ontology, e.g., [6] and [20]. In particular, the IoT-Lite ontology described in [6] builds a core model containing only the main concepts, along with their relationships, to support the most standard queries for IoT solutions. Originally proposed by the W3C Semantic Sensor Network Incubator group, the SSN ontology has also been revised by [20] in the sensor, observation, sample, and actuator (SOSA) ontology. This proposal aims at a lightweight vocabulary, including broader concepts with respect to the SSN ontology, with the idea to provide a core model that can be integrated and aligned to other specifications, e.g., the OGC's observations and measurements (O&M) or the DOLCE-ultralite (DUL).

Researchers also focused on modeling the environments in which IoT sensors are deployed, e.g., [7], and [31]. One of the first contribution in this direction is represented by the DogOnt ontology [7]. DogOnt focuses on a house automation scenario with the objective of modeling home environments, their states and changes, to enabling interoperation mechanisms and intelligence for more complex actions. This ontology models both architectural elements and controllable ones. More recently, Rasmussen et al. [31] proposed the BOT

to support the exchange of information related to building life-cycles among the actors of the architecture, engineering, construction, owner, and operation industry and according to the building information modeling (BIM) methodology. The authors show how their ontology, combined with other existing ones (modeling sensors, observations, and IoT devices), can support existing applications to make them interoperable and shareable among interdisciplinary stakeholders.

In the context of the Industry 5.0, another important aspect is related to the organization of the industry itself, the roles of the human resources, their hierarchies and privileges, and so forth. When it comes to semantically modeling the concepts involved in an organization, a reference ontology is the Organization ontology (ORG) [12], aimed at modeling organizational structures and related information through the concepts of organizations, their actors, activities, and roles.

Through the availability of domain specific ontologies, researchers, and industries have started to adopt more and more knowledge graph-based solutions to realize flexible and homogeneously integrated systems [18]. This is true especially for the IoT domain, in which the intrinsic heterogeneity of devices and standards requires the construction of suitable integration solutions to allow for a flexible and fruitful exploitation of the produced data and services [21], [24], [36]. For instance, the work described in [24] focuses on an cyber–physical production system equipped with an industrial IoT (IIoT). The huge number of interconnected devices (sensors, actuators, and the edge computing devices) produce massive heterogeneous multidimensional data. Therefore, seeking an effective data representation strategy, the authors propose a multilayer knowledge graph, which includes not only data produced by IoT devices but also production and business processing data. Leveraging the so constructed knowledge graph, the authors, then, propose a cognition decision making solution for resource allocation to support manufacturing processes. Still related to the domain of interest for our proposal, many authors have also focused on the application of knowledge graphs to support industrial environments, in which heterogeneous data and multidisciplinary information are typically produced [4], [23]. Li et al. [23] focused on the exploitation of the knowledge graph technology to support the development of products and service innovation in the industry domain. To do so, they survey the related literature, up to early 2021, to identify promising solutions exploiting knowledge graphs in the industrial contexts. Despite many works have been focused on the adoption of semantic modeling and knowledge graphs to support data integration in both the IoT and industry domains, the advent of the Industry 5.0 and the IoE paradigms introduces new challenges that, to the best of our knowledge, none of the existing solutions consider and address. The shift toward a more complex scenario, in which humans, as main actors actively producing data through their body sensor networks, collaborate with smart and semi-autonomous devices, production chains, and business processes, requires the redesign and adaptation of existing data models. This is the objective of our proposal, which builds upon some of the existing solutions described in this section, and extends them to embrace such a new promising industrial ecosystem.

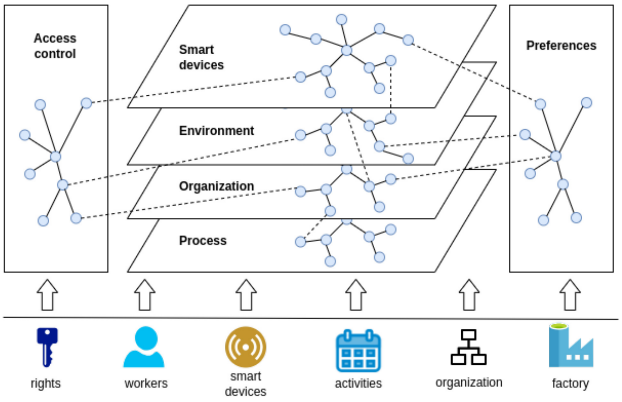

Fig. 1. Overview of the information modeled in the framework.

III. Reference Scenario

In this section, we discuss how information in the framework is categorized based on its typology and the object it represents. First, we recognize the main perspectives that are intertwined in an IoE environment, as also summarized in Fig. 1:

1) the *smart objects* perspective focuses on smart devices composed by one or more systems. In turn, each of them can be composed of a number of other devices, such as sensors and actuators. Their technical capabilities, operating ranges, configurations and locations are described;
2) the *environment* perspective, which focuses on the enterprise environment and its structure in subcomponents, e.g., buildings, floors, rooms;
3) the *organization* perspective, which includes the organizational structure in terms of suborganizations and units, and the reporting structure for the employees, where they are located and what roles are defined for them; and
4) the *process* perspective, which details processes in terms of activities in which employees are engaged.

Furthermore, two relevant perspectives can be identified that are orthogonal to the mentioned ones, namely:

1) *preferences* of employees related to environmental or smart devices' parameters and
2) *access control*, which focuses on what rights are associated with each organization role. Rights can be defined of three types with decreasing granularity: on environment, which apply on all systems located in the same place, on smart objects, which apply to all its subsystems, and on specific systems. Furthermore, rights can be transferred among agents (employees or smart objects) in the execution of an activity/process, through different types of relations, namely *delegation* or *collaboration*. In the former case, the existing rights of the delegated agent for the delegated activity are replaced with those of the delegating agent. Delegation can happen only if the delegated agent is not involved in another activity. On the other hand, in the latter case, the rights already

owned by the agent are combined with those of the collaborating one.

The data model of the framework is specifically aimed to: 1) provide a model of the IoE (including buildings, agents, roles, devices, smart objects, and processes); 2) provide a characterization of smart objects and devices in terms of their technical specifications and capabilities; 3) define rules and constraints for access management; and 4) store values produced by smart devices and employees, including recorded measurements, environmental parameters, actions, and activities performed by devices and employees. Aspects (1–3) correspond to the deployment, planning, and configuration of the IoE, while aspect 4 is concerned with capturing and analyzing the real-time execution and performance of tasks, including the storage of values generated by smart devices and employees.

In the context of Industry 5.0, data management necessitates a foundation built on flexible, interoperable, and easily maintainable structures that adhere to standards in model representation. While for aspect 4 we refer to standard technologies for data storage of IoT devices, this work adopts a knowledge graph model for aspects (1–3) in alignment with established best practices from related literature. On the top of it, a set of logical rules are defined in order to represent constraints, e.g., for access control, that cannot be directly represented in the graph model.

Knowledge graphs are today widely adopted at an enterprise level for their capability of providing a standardized way of representing and organizing data from multiple sources into a unified structure. As such, this model enhances data interoperability in scenarios characterized by diverse and heterogeneous information, by referring to shared vocabularies and a structure that can be understood and shared by different systems and applications. The use of graphs also simplifies data querying and analysis by facilitating the exploration of relationships between entities, hence reducing the time and effort required to find and integrate relevant information.

In particular, we refer to a graph model rooted in resource description framework (RDF) [5], a W3C standard for Web-based data modeling, storage, and interchange that provides a simple way to describe resources and their relationships, based on the notion of triple. Commonly used for (meta)data integration and knowledge representation, RDF is often extended by RDFS (RDF schema) [8], which enhances the vocabulary for describing data structures, including classes, properties, and subclass/super-class relationships, enabling precise and expressive resource descriptions. Additionally, OWL [26], an extension of RDFS, introduces formal semantics for added expressiveness in describing relationships among resources. In this work, RDFS and OWL are used for the definition of the SemIoE ontology, which serves as a schema for the platform knowledge graph and is introduced in the next section.

A. Example Scenario

In the following, we describe an example scenario that will be used throughout the remaining of this article to help the reader understanding our proposed solution. As sketched in Fig. 2, the considered scenario involves an industrial IoE. Here, two employees, namely "John Doe" and "Jane Smith," are both located at the site "Assembly Line 1" of a factory that produces automotive components, having the role of "Assembly Technician" with experience *level 1* and "Quality Control Specialist" with experience *level 2*, respectively. Both roles involve the right to read from a "CNC machine," which is a smart object located in the same site used to produce cylinder heads, engine blocks, and crankshafts that require high precision. In our example, the CNC machine provides access to its services and data through an underlying IoE network, thus allowing for real-time settings and adjustments. Moreover, a "cobot," a collaborative robot used to load raw materials and unload finished components. The cobot is equipped with AI capabilities and smart sensors to ensure an efficient interaction with human employees and reduce the risk of injuries during critical working activities. The factory environment also includes a smart "air purifier," to filter the particles of metal emitted by the CNC machine. The air purifier continuously monitors the quality of the air in the environment, adjusts air filtration configuration, and reports warnings in case of critical air quality conditions. Through the underlying IoE, the factory implements a comprehensive control system integrating data from the different smart devices available in the monitored environment. The system is designed to support the interaction among devices and employees during their working hours, as well as the managers to monitor the factory activities. To this objective, each smart device declares a set of capabilities that it can exploit to carry out tasks. For instance, the CNC machine can adjust its working settings based on the temperature information; therefore, it can also measure and provide temperature data of the surrounding environment. On the other hand, although its main goal is to maintain a safe air quality in the environment, the air purifier is also equipped with a sensor to measure the concentration of pollutants. This functionality is also declared as a secondary service offered by the device.

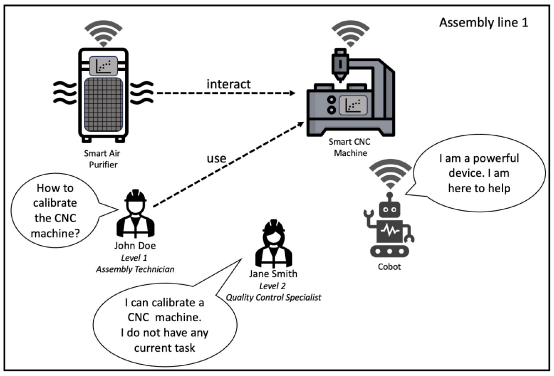

Fig. 2. Graphical representation of the considered example scenario.

Data and services are accessed and exploited through the considered IoE, which embraces the principles of autonomy, self-optimization and self-healing. Therefore, each involved entity can autonomously interact with other actors to support the activities of the industry they belong to. In particular, the self-healing mechanism is equipped with a collaborative

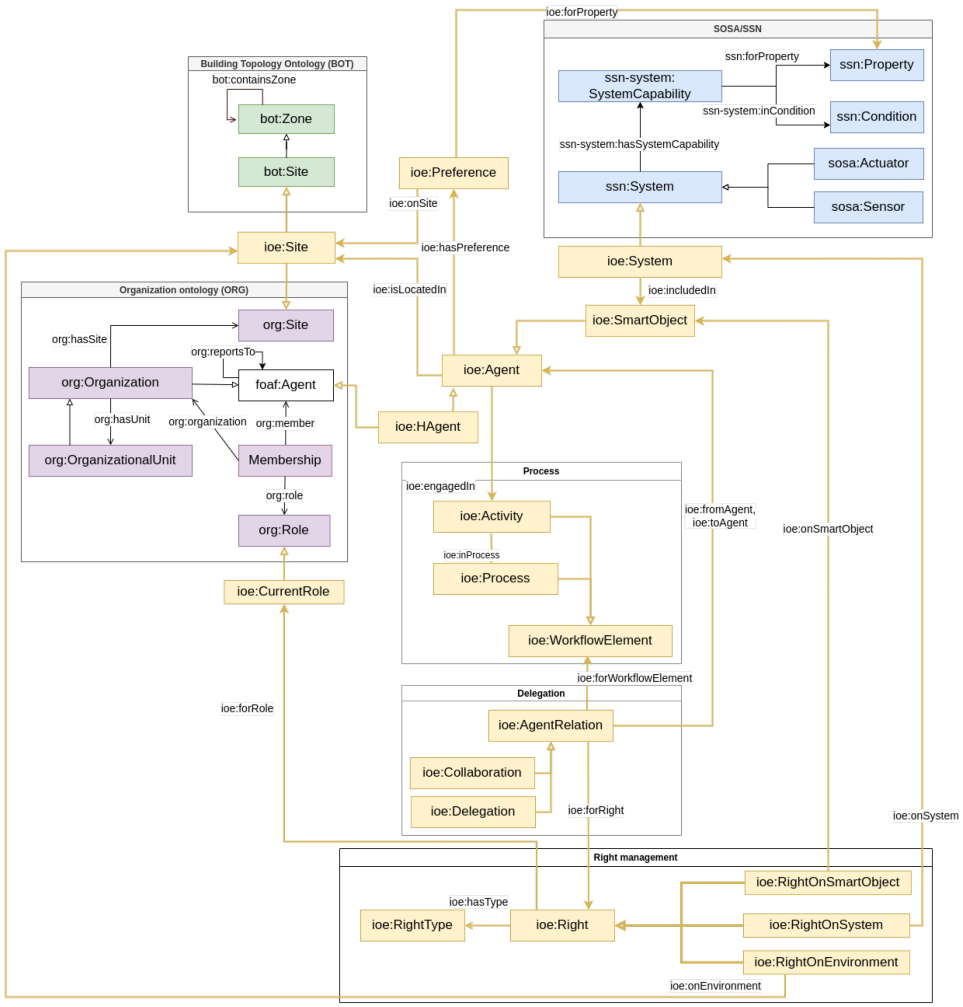

Fig. 3. Integrated schema of the SemIoE ontology.

anomaly detection strategy based on the approaches described in [1], [2], and [3]. According to this solution, smart devices of the network monitor each other by learning the expected behavior of their peers (behavioral fingerprinting) through lightweight deep-learning models based on gated recurrent unit (GRU) layers that as input the headers and the payload of the communication packets. Once such a typical behavior is known, a peer can verify the presence of anomalies by applying the underlying model to newly generated data. This is the case of the air purifier constantly interacting with the CNC machine to activate suitable air filtering when the smart milling machine is working on specific materials. Therefore, it is also in charge of identifying potential anomalies of the CNC machine using the aforementioned Behavioral Fingerprinting solution.

The employee John, instead, is assigned the task of configuring a new contour milling project for the CNC machine. Such a task would imply a calibration step that requires the support of a more experienced (level 2) employee. The role of John is a transferable one, so he can engage other users to give him support on his tasks. Jane, the other employee in this example, is a level 2 quality control specialist and, therefore, has enough experience for advanced configurations of CNC machines.

IV. Ontological Model for IoE

This section aims to define the schema of SemIoE, an OWL2 lightweight ontology designed to represent agents, smart objects, and other entities within an IoE ecosystem. The design of SemIoE is inspired to linked open terms (LOTs) [28], a modern ontology engineering methodology aimed at creating, maintaining, and using ontologies in a way that emphasizes Linked Data principles and openness. Starting from (i) a use case specification (see previous the Section), (ii) scope and functional ontology requirements have then been elicited. As for the (iii) implementation, external ontologies have been identified and extended (i.e., soft reuse by subclassing existing concepts). As such, the ontology reuses and integrates, according to the best practices in ontology engineering, a number of existing ontological models, as also summarized in Fig. 3. Further design patterns[1] have also been considered, among which N-ary relational pattern, quantity triad, naming, and annotation patterns. Finally, (iv) the ontology has been validated against the use case (see also

[1] http://ontologydesignpatterns.org

TABLE I
PREFIXES AND NAMESPACES FOR THE IMPORTED ONTOLOGIES

Ontology	Prefix	Namespace
Semantic Sensor Network	ssn ssn-system	http://www.w3.org/ns/ssn/ http://www.w3.org/ns/ssn/systems/
Sensor, Observation, Sample and Actuators	sosa	http://www.w3.org/ns/sosa/
Building Topology Ontology	bot	https://w3id.org/bot#
Org ontology	org	http://www.w3.org/ns/org#

Section VI), (v) published online, provided with a DOI, and (vi) documented.[2]

In the following, we first describe the imported ontologies in Section IV-A, then in Section IV-B we introduce the set of classes and relations capable to provide the needed connections among the different modules. We refer to classes and properties from the imported ontologies through the prefixes and corresponding namespaces reported in Table I. The namespace of SemIOE is the URI http://w3id.org/semioe#, while the prefix *ioe* is used for classes and relations. Please note that the URI is a persistent identifier which enables content negotiation. Adhering to best practices for Linked Data, it allows to serve the ontology specification in HTML or the serialization in various formats depending on the accepted MIME type of the HTTP request.

A. External Ontologies

Following widely adopted best practices on reusing ontological models, the development starts from the identification of relevant existing vocabularies/ontologies focusing on specific aspects of the IoE, which are detailed in the following.

1) Semantic Sensor Network and Sensor, Observation, Sample, and Actuator: the purpose of the SSN ontology [11] is to describe devices in terms of capabilities, measurement processes, observations, and deployments. In SSN, a *ssn:System* is an abstraction for a physical device which can contain other systems. A system is described in terms of a set of *ssn-system:SystemCapability*, (e.g., accuracy, drift, frequency, precision, and response time), which is a subclass of *ssn:Property* and describes its capabilities in various *ssn-system:Condition*s. On the other hand, the SOSA ontology [20] provides a lightweight core for SSN, which helps in broadening the target audience and application fields, e.g., by considering both *sosa:Sensor* and *sosa:Actuator* as subclasses of system in a coherent framework.

2) Building Topology Ontology (bot): developed by the W3C linked building data community group, the building topology ontology [31] is a minimal OWL DL ontology, aimed at defining relationships between the subcomponents of a generic building. The main class is the *bot:Zone* which is defined as a part of the physical/virtual world that is inherently both located in this world and has a 3-D spatial extent. Subclasses are of different types: a *bot:Site* is an area containing one or more *bot:Building*s, i.e., an independent unit of the built environment with a characteristic spatial structure. This last includes one or more *bot:Storey*, namely a level part of a building, which in turn contains *bot:Space*s, i.e., limited 3-D extents defined physically or notionally. Besides containment, other spatial relations are defined, such as adjacency and intersection. Several classes are aligned to upper level ontologies, such as DUL ontology and domain ontologies, among which BRICK, DogOnt, and ThinkHome.

3) Organization Ontology (org): published as a W3C recommendation, the organization ontology [12] is aimed to represent organizational structures. It is designed to allow domain-specific extensions to add classification of organizations and roles, as well as extensions to support neighboring information, such as organizational activities. The ontology includes classes and properties to support the representation of information typically reported in organizational charts: 1) the organizational structure, including the decomposition of an *org:Organization* into suborganizations and *org:OrganizationalUnit*s; 2) the reporting structure with *org:Membership*, *org:Role*, *org:Post*, and relations among people; 3) location information including *org:Site*, i.e., buildings where the organization (or a subpart thereof) is located; and 4) the organizational history (e.g., renaming of structures). Apart from these core concepts, the ontology can be extended to include more detailed information on organizational control structures and flows of accountability and empowerment.

B. Integrated Model

The ontological modules described above have been integrated through a bridge ontology which is graphically represented in Fig. 3. On the one hand, it allows to define the needed connections and alignments between relevant classes of different ontologies for the realization of an integrated model of the IoE environment. On the other hand, the bridge ontology defines further classes and properties to represent concepts that are relevant for the IoE environment (see Section III) and are not available in external ontologies. As such, the resulting ontology is equivalent to $\mathcal{O}_{\text{SemIoE}} \equiv \mathcal{O}_{\text{Bridge}} \sqcup \mathcal{M}_{\text{sosa/ssn}} \sqcup \mathcal{M}_{\text{bot}} \sqcup \mathcal{M}_{\text{org}}$, where $\mathcal{M}$ stands for the ontological mappings to the corresponding ontology.

The main class in the bridge ontology is *ioe:Agent*, which represents an agent in the IoE as a generalization of both a *ioe:HAgent*, i.e., a human agent, and a *ioe:SmartObject*, i.e., a smart device. Such a distinction allows to associate properties to each subclass separately. Both classes are intended to be extended with further properties for future usage.

The class *ioe:SmartObject* includes at least a *ioe:System*, which represents a device. According to our modeling approach, a smart object encompasses both a simple device, e.g., a sensor, and a more complex smart device, i.e., a smart watch. While in the former case the smart object would include a single system, in the second case it can include multiple systems (property *ioe:includedIn*). While technical capabilities of a system can be expressed through the class *ssn-system:SystemCapability*, higher level functionalities provided by the system can be represented by extending this concept.

An agent is located at an *ioe:Site* of the organization. Furthermore, an agent can be engaged in an *ioe:Activity*. This

[2]Documentation was produced through LODE (https://essepuntato.it/lode/) and Widoco (https://github.com/dgarijo/Widoco/).

holds for both a human agent and a smart object, which in some cases may possess enough capabilities to perform activities in the organization. In the execution of an activity, multiple agents can be engaged, e.g., if a human agent is performing an activity using a CNC machine, both the former and the latter are engaged in the same activity.

The representation of the process perspective is realized by representing activities through the class *ioe:Activity*, which is part of a *ioe:Process*, and both are subclasses of *ioe:WorkflowElement*.[3]

The access control perspective includes the representation of rights through the class *ioe:Right*, which defines a right of a particular *ioe:RightType*, e.g., read, write, update, delete, and is associated with a *ioe:CurrentRole*. It includes as subclasses:

1) *ioe:RightOnSystem* is connected to the specific *ioe:System* which is the target of the right through property *ioe:onSystem*;
2) *ioe:RightOnSmartObject*, on the other hand, specifies a right which applies to a whole smart object (through property *ioe:onObject*), including all the (sub)systems included thereby; and
3) *ioe:RightOnEnvironment* defines, through property *ioe:onEnvironment*, a right on all smart objects which are *ioe:locatedIn* a given *ioe:Site*.

Finally, the representation of preferences is realized through the class *ioe:Preference*, which can be expressed by an *ioe:Agent*, either a human agent and a smart object. A preference is defined on a certain *ssn:Property* measured in a given *ioe:Site* and specifies (through the data property *ioe:hasPreferenceValue*) the preferred value for the property. For what concerns the relations among agents, the class *ioe:AgentRelation* defines a relation of *ioe:Collaboration* or *ioe:Delegation* between two *ioe:Agent*s (relations *ioe:fromAgent*, *ioe:toAgent*). The relation needs to specify the activity/process it is intended for, through property *ioe:forWorkflowElement*. As a result of the establishment of the relation, one or more new *ioe:Right*s are associated with the new *ioe:CurrentRole* of the target of the collaboration/delegation. The role specifies its temporal validity through starting and ending time (data properties *ioe:startTime* and *ioe:endTime*) and whether it can be transferred to other agents (data property *ioe:isTransferable*).

The ontology includes the alignments among classes of the bridge ontologies and classes of external modules that are summarized in Table II. As such, by inheriting properties from the parent classes, a *ioe:System* can be defined compositionally, through the property *ssn:hasSubSystem*, while a human agent can be assigned a *org:Role*. Furthermore, the containment relation among sites from the BOT (*bot:containsZone*) can be defined for *ioe:Site*s. The alignment also allows to recognize that an organizational unit, e.g., an office, is the same *ioe:Site* where an agent is located.

The ontology has been implemented through Protégé (version 5.6.3).[4] It includes 136 axioms, 20 classes, and ten object properties. Its consistency has been checked with the reasoner HermiT (version 1.4.3.456).[5] The serializations of the ontology in the Turtle [5] and RDF/XML formats are available at the URL http://w3id.org/semioe along with the documentation.

TABLE II
ALIGNMENTS TO EXTERNAL ONTOLOGIES

Local class	Property	External class
ioe:System	rdfs:subClassOf	ssn:System
ioe:Site	rdfs:subClassOf	bot:Site
ioe:Site	rdfs:subClassOf	org:Site
ioe:CurrentRole	rdfs:subClassOf	org:Role
ioe:HAgent	rdfs:subClassOf	foaf:Agent

```
:assembly_line_1 rdf:type ioe:Site .

:calibration rdf:type ioe:Activity .

:jane_smith rdf:type ioe:HAgent ;
            ioe:locatedIn :assembly_line_1 .

:john_doe rdf:type ioe:HAgent ;
          ioe:engagedIn :calibration ;
          ioe:locatedIn :assembly_line_1 .

:CNC_machine rdf:type ioe:SmartObject ;
             ioe:locatedIn :assembly_line_1 .

:Assembly_Technician rdf:type ioe:CurrentRole ;
                     ioe:isTransferable "true"^^xsd:boolean .

:Quality_Control_Specialist rdf:type ioe:CurrentRole.

:membership_1 rdf:type org:Membership ;
              org:member :john_doe ;
              org:role :Assembly_Technician.

:membership_2 rdf:type org:Membership ;
              org:member :jane_smith ;
              org:role :Quality_Control_Specialist.

:agent_relation_1 rdf:type ioe:Collaboration ;
                  ioe:forWorkflowElement :calibration ;
                  ioe:fromAgent :john_doe ;
                  ioe:toAgent :jane_smith .

:configure rdf:type ioe:RightType .

:read rdf:type ioe:RightType .

:right_read_CNC_machine rdf:type ioe:RightOnSmartObject ;
                    ioe:forRole :Assembly_Technician ,
                                :Quality_Control_Specialist ;
                    ioe:hasType :read ;
                    ioe:onSmartObject :CNC_machine .

:right_config_CNC_machine rdf:type ioe:RightOnSmartObject ;
                    ioe:forRole :Assembly_Technician ;
                    ioe:hasType :configure ;
                    ioe:onSmartObject :CNC_machine .
```

Listing 1. Turtle serialization using the SemIoE ontology.

C. Example

In this example we show the serialization in Turtle of a simple case scenario using classes and relations from the SemIoE ontology. The scenario is the one described in Section III-A and, in this case, John asks for the collaboration of Jane in the execution of an activity "calibration" he is performing on the CNC machine. The resulting Turtle serialization is reported in Listing 1.

[3]More expressive ontological representations of processes and workflows have been proposed in the literature (see [14] for a discussion).

[4]https://protege.stanford.edu/

[5]http://hermit-reasoner.org/

Algorithm 1 Access Control

Require: a (Agent), s (System), e (Site), t (RightType)
1: **if** locatedIn(s)!=locatedIn(a) **then return** False
2: **else**
3: r: Role = getRole(a)
4: RH: list<Right> = getRights(role)
5: **for each** $rh_i \in RH$ **do**
6: **if** RightOnSystem(rh_i) **then**
7: S = getSystemsFromRight(rh_i)
8: **if** $s \in S \wedge t$==getType(rh_i) **then return** True
9: **end if**
10: **end if**
11: **if** RightOnEnvironment(rh_i) **then**
12: env = getEnvironmentFromRight(rh_i)
13: **if** $env == e \wedge t$==getType(rh_i) **then return** True
14: **end if**
15: **end ifreturn** False
16: **end for**
17: **end if**

V. Services for the IoE

In this section, we define different types of service providing support to the execution of specific activities within a Industry 5.0 scenario. Services are built on top of the semantic layer of the platform, which includes the SemIoE ontology and the knowledge graph. Hereby, we focus on the formulation of advanced support services assuring the fulfillment of activities possibly involving, at least in some cases, multiple agents that cooperate for their execution based on their roles. The following services will be detailed below: 1) IoE access control; 2) IoE collaboration; 3) IoE secure delegation; and 4) IoE environment setting.

A. IoE Access Control

In our scenario, we assume that the IoE facilities of the industry are restricted to agents with a specific role. Therefore, an IoE Access Control service is necessary to prevent unauthorized users from accessing the data collected by the sensors. The SemIoE ontology models each environment of the factory as an *ioe:Site* composed by a set of subsites, e.g., a building or a room. In each site, processes are regulated by a series of *ioe:System*s (e.g., sensors) included in *ioe:SmartObject*s. A specific *ioe:HAgent*, when entering the site, will be allowed to have access to only the subset of *ioe:System*s that are compatible with his/her *org:Role*. This service is crucial for a smart factory because the recent scientific literature reports different types of attack that can leverage the manipulations of smart sensors to orchestrate distributed attacks on the entire smart objects [17], [35].

As reported in Algorithm 1, when an agent a enters an environment e to perform an activity on a system s with a specific right type t (e.g., read), the framework retrieves the information that characterizes the *role* of the agent and the required rights to access the system s. In particular, the framework checks if the agent a and the specific system s are in the same environment e (line 1). If confirmed, the framework retrieves the rights $RH = \{rh_1, \ldots, rh_n\}$ corresponding to the role r of the agent a (lines 3 and 4). It cycles through them (line 5) and returns *True* if the agent has the rights on the system s of type t (lines 6–8), or if the agent has a right on environment e of type t (lines 11–13). Otherwise, the framework returns *False* if these conditions are not verified.

Algorithm 2 IoE Collaboration

Require: a_1 (Agent), a_2 (Agent), st (Start Time), et (End Time)
1: r_1: Role = getRole(a_1)
2: r_2: Role = getRole(a_2)
3: **if** isTransferable(r_1) **then**
4: RH_{r_1}: list<Right> = getRights(r_1)
5: RH_{r_2}: list<Right> = getRights(r_2)
6: $RH_c = RH_{r_1} \bigcup RH_{r_2}$
7: r_c = createTempRole(RH_c, st, et)
8: $a_2.r_2$ = assignRole(r_c, startTime=st, endTime=et)
9: **end if**

Use Case Example: Let us consider again the scenario of Section III-A, when the agent John enters the site and accesses the framework to configure the new milling task on the CNC machine. The agent will be identified with a specific role characterized by a set of rights, and the rights associated with the role will then be verified for compliance with the requested task. With a positive outcome, the framework allows the agent to get control of the CNC machine and provides the permissions to access the precision cameras, proximity and pressure sensors to configure it for the new milling task.

B. IoE Collaboration

The second service enables the collaboration between multiple agents (*ioe:HAgent*) with roles (*org:Role*) of different level. In particular, the service considers the case in which two agents, say a_1 and a_2, with different roles, need to collaborate to complete an activity. The requirement is that the two agents must be co-located in the same *ioe:Site*. The service ensures that an agent can require the collaboration of another agent and can grant access to the same facilities in the IoE. In practice when a_1 has to complete an activity in a given time t, it can leverage the cooperation of a_2 by temporarily extending its privileges (*ioe:Right*) to grant access to the facilities of the IoE accessible by a_1.

In particular, as described in Algorithm 2, when the agent a_1 asks to collaborate with a_2, the framework has to check if the role r_1 of agent a_1 is transferable. To do so, the framework uses the function *isTransferable* (line 3). If the function returns *False* the process ends, while if it returns *True* the collaboration procedure starts. In this case, the framework retrieves from the roles r_1 and r_2 the list of rights RH_{r_1} and RH_{r_2} associated with them (lines 4 and 5). Then, the framework can create a new temporary role characterized by a list of rights obtained by the union between RH_{r_1} and RH_{r_2} and a period in which this new temporary role is valid (lines 6 and 7). The new temporary role is now ready to be assigned to the agent a_2 (line 8).

Use Case Example (Continued): Now, John has to proceed with the calibration of the CNC machine, which is a required step before being able to execute the milling task. This task requires the collaboration of a level 2 employee to avoid possible errors. Through the framework, he can request the collaboration of another agent, the human employee Jane, in this case. Then, the framework verifies if the rights of John are transferable. With a positive response, the framework can

Algorithm 3 IoE Secure Delegation

Require: a_1 (Agent), a_2 (Agent), ac (Activity), st (Start Time), et (End Time)
1: S_{ac}:list<System> = []
2: r_1: Role = getRole(a_1)
3: r_2: Role = getRole(a_2)
4: **if** isTransferable(r_1) **then**
5: RH_{r_2}: list<Right> = getRights(r_2)
6: RH_{r_1}: list<Right> = getRights(r_1)
7: O_{ac}:list<Object> = getObjectsEngagedInActivity(ac)
8: $RH_{O_{ac}}$: list<Right> = getRightsOnObjects(O_{ac})
9: **for each** $o_i \in O_{ac}$ **do**
10: $S_{ac} \leftarrow$ getSystemsIncludedIn(o_i)
11: **end for**
12: $RH_{OS_{ac}}$: list<Right> = getRightsOnSystems(S_{ac}) $\bigcup RH_{O_{ac}}$
13: RH_d: list<Right> $= RH_{OS_{ac}} \bigcap RH_{r_1}$
14: r_d = createTempRole(RH_d, st, et)
15: $a_2.r_2$ = assignRole(r_d, startTime=st, endTime=et)
16: **end if**

now generate a new temporary role for Jane with the required rights to fulfill the task assigned to John.

C. IoE Secure Delegation

Analogously to the previous service, a given *ioe:Agent*, say a_1, can leverage the availability of another *ioe:Agent*, say a_2, to complete a critical activity on a_1's behalf.

The idea behind this service is to allow the agents to share access to resources, devices or data, available to them, with an adequate security level.

In practice, following Algorithm 3, due to the impossibility of executing an activity, agent a_1 can delegate the process to a second agent a_2 to carry out a given activity ac, in time. For this reason, the delegation d must be elapsed in a given period of time defined by a start time st and an end time et. In particular, the current roles r_1 and r_2 of the agents are retrieved using the function *getRole* (lines 2 and 3). Before proceeding with the process of delegation, the framework checks if the role r_1 is transferable to other agents (line 4). If the outcome is positive, is now possible to retrieve the smart objects O_{ac} involved in the execution of activity ac together with a_1, using *getObjectsEngagedInActivity* (line 7). Then, the rights associated with these objects can be derived with the function *getRightsOnObjects* (line 8). After that, the list of objects can be used to get the involved systems S_{ac} using *getSystemsIncludedIn* and all the associated rights with *getRightsOnSystems* (lines 9–11). At this point, the algorithm identifies the subset of rights RH_d, intersection between the rights RH_{r_1} associated with the role r_1 of the agent a_1 and the entire set of rights, namely $RH_{OS_{ac}}$ (line 12), on the objects O_{ac} and systems S_{ac} involved. The retrieved set of rights RH_d can now be used to generate the delegation rights (line 13). The obtained temporary rights are used to generate the new delegation role r_d (line 14), which is then assigned to a_2 with a limited life span limited by st and et (line 15).

Use Case Example (Continued): The reliability of the CNC machine is guaranteed by the constant anomaly detection performed by the air purifier in the scenario of Section III-A. Considering a scenario in which the CNC machine communicates its status (active, idle, heavy load, or maintenance mode) with the air purifier to reduce its activity during low production periods, thus saving energy and reducing operational costs.

Algorithm 4 IoE Environment Setting

Require: a_1 (Agent), e (Site)
1: P_e: list<Property> = []
2: $Pref_{a1}$: list<Preference> = getEnvPreferences(a_1)
3: O_e: list<SmartObject> = getObjectsLocatedIn(e)
4: **for each** $o_i \in O_e$ **do**
5: S_i: list<System> = getIncludedSystems(o_i)
6: $P_e \leftarrow$ getSystemsProperties(S_i)
7: **end for**
8: **for each** $p_i \in P_e$ **do**
9: p_i = applyPreferences(p_i, $Pref_{a1}$)
10: **end for**

This characteristic can be exploited by a maliciously controlled CNC machine that can perform cyber–physical attacks by communicating fake statuses to compromise the efficiency of the system. The air purifier, equipped with a fingerprinting model of the CNC machine, can efficiently detect the anomalous behavior by analyzing the content and the frequency of the packets received by the milling machine. However, this device may not be able to initialize and execute a behavioral model, as this task requires the training and inference of a deep learning model [1], [2], [3]. For this reason, it can ask through the framework if a more capable device is available for managing a behavioral fingerprinting model using its data. The framework identifies the cobot as a device showing high computational capability. Similarly to the collaboration service, the delegation service proceeds by generating a new role to assign to the identified powerful device that now will act on behalf of the air purifier to complete the target task.

D. IoE Environment Setting

This service focuses on the adaptation of the environmental parameters of a *ioe:Site*, e.g., a room, according to the preferences specified by an *ioe:Agent*. The service can access the information from the multiple *sosa:Sensor*s inside the site, and can leverage the actuators (*sosa:Actuator*) to adapt the environmental parameters of the room on the basis of the preferences of the *ioe:HAgent* that enters the site. This service is aimed at improving the productivity of an agent adjusting its working environment, such as, for example, the room temperature, the lightning level, and so forth, matching its preferences automatically. As presented in Algorithm 4, when an agent a_1 accesses a site e the framework retrieves the system's properties preferences Pref_{a1} specified by agent a_1 (line 2). Then, the framework retrieves the systems and their associated properties P_e in the site e (lines 3–6) and applies to them the preferences Pref_{a1} of the agent a_1 (line 9).

Use Case Example (Completed): After the calibration, the CNC Machine is ready to carry out the new milling task. At this point, when activated, it can alter the environment setting to require the increase of the power of the filtration system of the air purifier to absorb the produced particles and preserve the air quality of the room.

VI. Implementation and Discussion

A. Implementation

In this work, we refer to a basic IoT infrastructure comprising a collection of smart objects, with some permanently

```
SELECT ?p ?o
WHERE {<ins> a <entity>;
               ?p ?o}
```

Listing 2. SPARQL query for information on instance *ins* of type *entity*.

```
SELECT ?site
WHERE {<agent> a ioe:Agent;
                 ioe:locatedIn ?site}
```

Listing 3. SPARQL query for function *locatedIn()*.

```
SELECT ?role
WHERE {?m a org:Membership;
          org:member <agent>;
          org:role ?role}
```

Listing 4. SPARQL query for function *getRole()*.

```
SELECT ?right
WHERE {?right a ioe:Right;
              ioe:forRole <role>}
```

Listing 5. SPARQL query for function *getRights()*.

```
SELECT ?sys
WHERE {<right> a ioe:RightOnSystem;
                 ioe:onSystem ?sys}
```

Listing 6. SPARQL query for function *getSystemsFromRight()*.

```
SELECT ?agent
WHERE {?agent a ioe:HAgent.
       MINUS {?agent ioe:engagedIn ?act.
              ?atc a ioe:Activity.}}
```

Listing 7. SPARQL query to retrieve available agents.

deployed in the environment and others worn by employees. These devices possess the capability to monitor various parameters and execute actions. Generated data can be optionally stored in a cloud infrastructure, typically through standard protocols (e.g., MTTQ and HTTP) or directly retrieved from the device, depending on their peculiarities and capabilities. For some type of information, e.g., on personal health-related aspects, data is stored and consumed locally on the devices and cannot be accessed from outside. The knowledge graph of the IoE is defined on the top of the IoT infrastructure, enabling the integration of multiple, heterogeneous information by relying on classes and relations defined in the SemIoE ontology.

To support data storage and retrieval of information from the knowledge graph, a catalog of services provides functionalities at different levels, ranging from functions to perform low-granularity operations to advanced capabilities.

1) Storage Layer: It includes low-level CRUD functionalities operating on specific triples, namely the capability to add (delete) a given triple to (from) the graph. Such functions are provided by a triplestore, i.e., a DBMS specifically tailored to the storage and management of triples. Additionally, generic SPARQL [16] queries can be run on the graph, to support medium- and higher-level functions.

2) Entity Layer: It includes medium-level functionalities to manage graph entities, such as agents, smart devices, environments, roles, rights, or privileges. These functions typically operate on groups of triples and are categorized in two classes. The first is formed by functions aimed to extract all triples having the given instance as a subject. The following SPARQL query extracts all information centered on a given instance *ins* of class *entity*, in terms of properties and corresponding objects:

A second group of functions are devoted to extract specific pieces of information from the graph, to support more advanced functionalities. The following ones are, respectively, aimed to extract the *ioe:Site* in which an *ioe:Agent* is located (locatedIn), its *ioe:CurrentRole* (getRole), the rights associated to such a role (getRights) and the systems on which a given *ioe:RightOnSystem* is defined (getSystemsFromRight).

Further entity level functions can support specific purposes that are useful in the IoE context, e.g., retrieval of resources having a given status. For instance, the following query searches for all human agents in the organization that are available, i.e., that are not currently assigned to some activities.

3) Support Layer: It includes high-level functions to implement the advanced management services for the IoE, as described in Section V.

Each group relies on the functionalities provided by the lower level. To give an example, the advanced service *IoE Access Control* relies on the entity-level services *getRole* and *getRights*, respectively, to extract the role for a given agent and the rights for a given role. Likewise, the service *getRole*, as shown, requires the capability to query triples related to the specified role. The functionalities provided by each layer are implemented as APIs in the framework, on which the logic of higher-level functions are constructed.

To assess the practical usability of the model, we conducted tests across various scenarios using synthetic knowledge graphs of varying sizes. These scenarios ranged from smaller cases (100 human agents and 1000 smart objects, each containing one to ten systems) to larger cases (350 human agents and 20 000 smart objects), with knowledge graph sizes spanning from 150k to 2.9M triples. In all tested scenarios, the execution time for the reported queries consistently remained between 10 and 30 ms.[6]

B. Discussion

SemIoE is designed as a flexible lightweight ontology, with a minimal ontological commitment. As such, the ontology can be extended to accommodate changed requirements through dynamic extension of concepts, e.g., with the aim to add further subtypologies of smart objects, or further properties of an agent. As shown, it is intrinsically adapted to support operations at different scales, since it is meant to represent metadata related to smart objects, people and processes in an IoE scenario, while data produced by devices at the IoE

[6]On GraphDB 10.2.2 running on Rocky Linux 9.4, Intel Xeon Gold 6252N CPU at 2.30 GHz, 4 cores, and 16 GB RAM.

level can be stored in respective storage systems and their management, including scalability issues, is at framework level. As a matter of fact, because our approach is intended for metadata representation, also the security aspects concerning access control and secure handling of data, typically related to the CIA (confidentiality, integrity, and availability) triad, are demanded to the local endpoint of the involved data sources.

As discussed, the ontological layer provides support to integration of metadata of heterogeneous devices and systems in a uniform manner. This provides the basis for a variety of practical use cases in an Industry 5.0 scenario, ranging from supporting data access, data stream monitoring, and complex automation, to defining advanced and adaptive mechanisms for managing authorization. On the one hand, an ontology-based data access approach can be designed to bridge the gap between semantics of data and its format, allowing for more flexible and expressive querying capabilities while maintaining compatibility with existing database systems. In case data is fed in the system as data streams, a similar approach can be used to subscribe to semantically enriched topics, e.g., all data produced by sensors of a given type in a specific site of the organization, in order to filter only contextually relevant information. This paves the way to designing complex applications, including immersive tools for human–computer interaction, which can be built by mesh-up of information from multiple sources, and can provide an integrated view which is context-based, relevant for fulfillment of tasks at hand and compliant with authorization policies.

VII. Conclusion

In this article, we have focused on a novel reference scenario, in which the IoE technology is adopted as a main driver to foster the transition toward the Industry 5.0 paradigm. Our investigation started from the analysis of the complexity of the considered scenario, for which the integration of the data produced by the entities of such an industrial IoE appears critical. Following the reuse and abstraction design principles, we, hence, proposed a novel ontology, called SemIoE, which is built by extending and complementing consolidated state-of-the-art models. The SemIoE is a lightweight ontology formalizing the entities and the dynamics involved in an IoE-based industry, and designed as a foundational model that can be extended to cope with any industrial context. In particular, its classes and relationships have been identified by considering the core concepts of a modern IoE-based industrial environment. Therefore, these fundamental representations can be extended using design principles of reusing and abstraction, adapting the data model to the specific data sources available in a target industrial domain (for instance, by defining subtypologies of smart objects, or further properties of an agent). Additionally, our core ontology is compatible with existing ontology-based platforms, such as the Palantir Foundry.[7] Additionally, to support the activities carried out in our reference Industry 5.0 scenario, we defined and developed a set of IoE services built on top of a knowledge graph leveraging SemIoE ontology. The proposal described in this article represents a first contribution toward the establishment of new-generation industries embracing the Industry 5.0 paradigm and leveraging the IoE technology.

In the future, we plan to extend our solution by considering a more refined concept of service provisioning in the IoE, according to which each involved entity can act as both a user and a provider of a set of services. Moreover, the data produced by specific services may be subject to different security and privacy constraints, thus imposing the necessity to model classes of security and privacy requirements that could extend our ontology. In this sense, a notion of "scope" of a smart object will be considered, e.g., as in [9], involving aspects, such as trust, impact on other objects and security, focusing on exchanges of information among devices. This would help in recognizing malfunctioning devices early and avoid propagation of erroneous information, or, in critical scenarios where a high quality level is required, to filter devices according to the scope for which their measurements are suitable. Finally, we plan to integrate our solution with a human interaction module that will engage with the platform services and the knowledge graph, contributing to the creation of a user-centric immersive working environment.

[7] https://www.palantir.com/docs/foundry/platform-overview/overview/

References

[1] A. Aramini, M. Arazzi, T. Facchinetti, L. S. Q. N. Ngankem, and A. Nocera, "An enhanced behavioral fingerprinting approach for the Internet of Things," in *Proc. IEEE 18th Int. Conf. Fact. Commun. Syst. (WFCS)*, 2022, pp. 1–8.

[2] M. Arazzi, S. Nicolazzo, and A. Nocera, "A fully privacy-preserving solution for anomaly detection in IoT using federated learning and homomorphic encryption," *Inf. Syst. Front.*, pp. 1–24, Nov. 2023.

[3] M. Arazzi, S. Nicolazzo, and A. Nocera, "A novel IoT trust model leveraging fully distributed behavioral fingerprinting and secure delegation," *Pervasive Mobile Comput.*, vol. 99, Apr. 2024, Art. no. 101889.

[4] S. R. Bader, I. Grangel-Gonzalez, P. Nanjappa, M. Vidal, and M. Maleshkova, "A knowledge graph for industry 4.0," in *Proc. 17th Int. Conf. Semant. Web (ESWC)*, 2020, pp. 465–480.

[5] D. Beckett and B. McBride, "RDF/XML syntax specification (revised)," *W3C Recomm.*, vol. 10, no. 2.3, pp. 1–56, 2004.

[6] M. Bermudez-Edo, T. Elsaleh, P. Barnaghi, and K. Taylor, "IoT-Lite: A lightweight semantic model for the Internet of Things," in *Proc. INTL IEEE Conf. Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People, Smart World Congr.*, 2016, pp. 90–97.

[7] D. Bonino and F. Corno, "DogOnt-ontology modeling for intelligent domotic environments," in *Proc. 7th Int. Semant. Web Conf.*, 2008, pp. 790–803.

[8] D. Brickley, R. V. Guha, and B. McBride, "Rdf schema 1.1," *W3C Recomm.*, vol. 25, p. 10, Feb. 2014.

[9] F. Cauteruccio et al., "An approach to compute the scope of a social object in a multi-IoT scenario," *Pervasive Mobile Comput.*, vol. 67, Sep. 2020, Art. no. 101223.

[10] M. Breque, L. De Nul, and A. Petridis, *Industry 5.0—Towards a Sustainable, Human-Centric and Resilient European Industry*. Eur. Commiss., Dir.-General Res., Innov., Brussels, Belgium, Publ. Office Eur. Union, Luxembourg, Luxembourg, 2021.

[11] M. Compton et al., "The SSN ontology of the W3C semantic sensor network incubator group," *J. Web Semant.*, vol. 17, pp. 25–32, Dec. 2012.

[12] *The Organization Ontology*. World Wide Web Consort. Stand. Organ., Cambridge, MA, USA, 2014.

[13] K. A. Demir, G. Döven, and B. Sezen, "Industry 5.0 and human-robot co-working," *Procedia Comput. Sci.*, vol. 158, pp. 688–695, Jan. 2019.

[14] C. Diamantini, A. Mircoli, D. Potena, and E. Storti, "Process-aware IIoT knowledge graph: A semantic model for industrial IoT integration and analytics," *Future Gener. Comput. Syst.*, vol. 139, pp. 224–238, Feb. 2023.

[15] M. Eid, R. Liscano, and A. El Saddik, "A novel ontology for sensor networks data," in *Proc. IEEE Int. Conf. Comput. Intell. Meas. Syst. Appl.*, 2006, pp. 75–79.
[16] W3C SPARQL Working Group, *Sparql 1.1 Overview*. World Wide Web Consort. Stand. Organ., Cambridge, MA, USA, 2013.
[17] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," *ACM Comput. Surv.*, vol. 42, no. 1, pp. 1–31, 2009.
[18] A. Hogan et al., *Knowledge Graphs*, (Synthesis Lectures on Data, Semantics, and Knowledge), vol. 12, Cham, Switzerland: Springer, 2021, pp. 1–257.
[19] M. Horridge, H. Knublauch, A. Rector, R. Stevens, and C. Wroe, *A Practical Guide to Building OWL Ontologies Using the Protégé-OWL Plugin and Co-Ode Tools Edition 1.0*, Univ. of Manchester, Manchester, U.K., 2004.
[20] K. Janowicz, A. Haller, S. J. D. Cox, D. Le Phuoc, and M. Lefrançois, "SOSA: A lightweight ontology for sensors, observations, samples, and actuators," *J. Web Semant.*, vol. 56, pp. 1–10, May 2019.
[21] D. Le-Phuoc, H. N. M. Quoc, H. N. Quoc, T. T. Nhat, and M. Hauswirth, "The graph of things: A step towards the live knowledge graph of connected things," *J. Web Semant.*, vols. 37–38, pp. 25–35, Mar. 2016.
[22] J. Leng et al., "Industry 5.0: Prospect and retrospect," *J. Manuf. Syst.*, vol. 65, pp. 279–295, Oct. 2022.
[23] X. Li, M. Lyu, Z. Wang, C. Chen, and P. Zheng, "Exploiting knowledge graphs in industrial products and services: A survey of key aspects, challenges, and future perspectives," *Comput. Ind.*, vol. 129, Aug. 2021, Art. no. 103449.
[24] M. Liu, X. Li, J. Li, Y. Liu, B. Zhou, and J. Bao, "A knowledge graph-based data representation approach for IIoT-enabled cognitive manufacturing," *Adv. Eng. Inform.*, vol. 51, Jan. 2022, Art. no. 101515.
[25] P. K. R. Maddikunta et al., "Industry 5.0: A survey on enabling technologies and potential applications," *J. Ind. Inf. Integr.*, vol. 26, Mar. 2022, Art. no. 100257.
[26] D. L. McGuinness and F. Van Harmelen, "Owl Web ontology language overview," *W3C Recomm.*, vol. 10, no. 10, p. 2004, 2004.
[27] A. Pease, I. Niles, and J. Li, "The suggested upper merged ontology: A large ontology for the semantic Web and its applications," in *Proc. Work. Notes AAAI Workshop Ontologies Semant. Web*, 2002, pp. 7–10.
[28] M. Poveda-Villalón, A. Fernández-Izquierdo, M. Fernández-López, and R. García-Castro, "Lot: An industrial oriented ontology engineering framework," *Eng. Appl. Artif. Intell.*, vol. 111, p. 104755, 2022.
[29] R. Raffik, R. R. Sathya, V. Vaishali, S. Balavedhaa, and N. J. Lakshmi, "Industry 5.0: Enhancing human-robot collaboration through collaborative robots–a review," in *Proc. 2nd Int. Conf. Adv. Elect., Electron., Commun., Comput. Autom. (ICAECA)*, 2023, pp. 1–6.
[30] H. Rahman and M. I. Hussain, "A comprehensive survey on semantic interoperability for Internet of Things: State-of-the-art and research challenges," *Trans. Emerg. Telecommun. Technol.*, vol. 31, no. 12, 2020, Art. no. e3902.
[31] M. H. Rasmussen, M. Lefrançois, G. F. Schneider, and P. Pauwels, "BOT: The building topology ontology of the W3C linked building data group," *Semant. Web*, vol. 12, no. 1, pp. 143–161, 2021.
[32] R. M. S. Priya et al., "Load balancing of energy cloud using wind driven and firefly algorithms in Internet of everything," *J. Parallel Distrib. Comput.*, vol. 142, pp. 16–26, Aug. 2020.
[33] D. J. Russomanno, C. R. Kothari, and O. A. Thomas, "Building a sensor ontology: A practical approach leveraging ISO and OGC models.," in *Proc. IC-AI*, 2005, pp. 637–643.
[34] C. Schlenoff, T. Hong, C. Liu, R. Eastman, and S. Foufou, "A literature review of sensor ontologies for manufacturing applications," in *Proc. IEEE Int. Symp. Robot. Sens. Environ. (ROSE)*, 2013, pp. 96–101.
[35] R. Vishwakarma and A. K. Jain, "A survey of DDoS attacking techniques and defence mechanisms in the IoT network," *Telecommun. Syst.*, vol. 73, no. 1, pp. 3–25, 2020.
[36] C. Xie, B. Yu, Z. Zeng, Y. Yang, and Q. Liu, "Multilayer Internet-of-Things middleware based on knowledge graph," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2635–2648, Feb. 2021.
[37] X. Xu, Y. Lu, B. Vogel-Heuser, and L. Wang, "Industry 4.0 and industry 5.0—Inception, conception and perception," *J. Manuf. Syst.*, vol. 61, pp. 530–535, Oct. 2021.

Open Access funding provided by 'Università Politecnica delle Marche' within the CRUI CARE Agreement