

I40GO: A global ontology for industry 4.0

William Ochoa *, Javier Cuenca , Felix Larrinaga , Alain Pérez 

Electronics and Computer Engineering department, Mondragon Unibertsitatea, Mondragon, 20500, Gipuzkoa, Spain

ARTICLE INFO

Keywords:

Semantic ontology
Ontology reuse
Industry 4.0
Context awareness
Workflow management

ABSTRACT

Over the last two decades, semantic ontologies have been developed to represent manufacturing data across various domains. These ontologies constitute the knowledge base of manufacturing management systems, which primarily focus on optimizing the manufacturing process and improving its resilience. The ontologies developed in the Industry 4.0 domain are heterogeneous, hindering the interoperability of machines, devices, and applications composing manufacturing systems. Consequently, a demand arises for an ontology that provides common vocabularies to represent the data domains inherent to Industry 4.0. A global Industry 4.0 ontology must be easily reusable in different application contexts. This paper presents I40GO: a global ontology tailored to the Industry 4.0 domain. I40GO structures in layers and modules the knowledge represented in the Industry 4.0 most relevant ontologies. The MODDALS methodology is followed to classify knowledge into different layers. This methodology classifies ontology knowledge into common, variant, and application-specific layers following a similar approach to that of Software Product Lines (SPL). I40GO assists ontology engineers in developing domain-specific ontologies for manufacturing systems and enhances interoperability among applications. This work provides an overview of I40GO, emphasizing its development methodology and its modular and layered structure. Furthermore, it demonstrates the reuse of the I40GO ontology within an Industry 4.0 use case—an architecture for context-aware workflow management.

1. Introduction

Industry 4.0, often referred to as the fourth industrial revolution, is characterized by the integration of new Internet technologies into manufacturing processes, allowing the exchange and analysis of manufacturing data (Lasi et al., 2014a). Cyber-Physical Systems (CPSs) play a crucial role in the achievement of the Industry 4.0 vision. These systems are distinguished by their capability to communicate with other systems, facilitating data sharing across networks and enabling more distributed, autonomous and reliable systems (Latif et al., 2022). The integration of CPS in manufacturing leads to more intelligent, resilient, and self-adaptable processes, especially within the domain of “Smart” factories (Lee et al., 2015).

Intelligent Industry 4.0 applications should monitor and control manufacturing processes and foster cooperation among machines and humans to make decentralized decisions (Grangel-González et al., 2016). To accomplish this, these applications must efficiently collect, exchange, and extract knowledge from heterogeneous and complex data domains (Yan et al., 2017). These domains encompass product and service descriptions, manufacturing device functionality, manufacturing processes, device automation, communication networks, and energy

efficiency (Lasi et al., 2014b). Considering this, data interoperability, device integration, and data availability are crucial challenges in achieving the Industry 4.0 vision.

To address these challenges, Semantic Web vocabularies are employed, facilitating the establishment of relations among heterogeneous domains, even when the linking data are represented in different formats (Cardoso & Sheth, 2006). Consequently, the application of Semantic Web technologies to represent Industry 4.0 facilitates seamless data exchange, ensuring a shared understanding of manufacturing information between CPSs and legacy systems (Grangel-González et al., 2016).

In the context of Industry 4.0, current research evidences the development of semantic ontologies as a response to these challenges. Ontologies, characterized as formal vocabularies, serve to encapsulate a data domain through a structured arrangement of concepts and relations between them (Hebeler et al., 2009). The main elements of ontologies encompass classes (used to represent concepts), properties (employed to establish connections between classes), and axioms (used to constrain properties, thereby expressing facts about classes) (Lohmann et al., 2016). The use of ontology vocabularies facilitates the integration of data from diverse and heterogeneous domains, enabling access, exchange, and processing of knowledge across the different applications

* Corresponding author.

E-mail addresses: wsochoa@mondragon.edu (W. Ochoa), jcuenca@mondragon.edu (J. Cuenca), flarrinaga@mondragon.edu (F. Larrinaga), aperez@mondragon.edu (A. Pérez).

<https://doi.org/10.1016/j.eswa.2025.128792>

Received 19 October 2023; Received in revised form 22 June 2025; Accepted 24 June 2025

Available online 27 June 2025

0957-4174/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

(Cardoso & Sheth, 2006). Often, ontologies are used to assist intelligent agents in data analysis, knowledge extraction, and decision-making (Gruber, 2009).

The ontologies developed for the Industry 4.0 domain such as the RAMI 4.0 vocabulary (Grangel-González et al., 2016), the MASON ontology (Lemaignan et al., 2006), and the AutomationML ontology (Kovalenko et al., 2015) have emerged to represent manufacturing data from various domains. These ontologies were developed to serve as the knowledge repository for manufacturing management systems, as exemplified in Lu and Xu (2017), Zhang et al. (2017), and Sipsas et al. (2020). These systems focus on managing various aspects of the manufacturing process, ranging from optimization of product design to optimal scheduling of large-scale manufacturing resources, all in the pursuit of improving operational efficiency and resilience.

1.1. Motivation

The development of ontologies in the manufacturing field began before the inception of the Industry 4.0 vision (Pramanik et al., 2020). However, these early ontologies were custom-tailored for specific manufacturing systems. Yet, they share a common characteristic: representing knowledge pertaining to manufacturing data domains but employing divergent vocabularies. This heterogeneity hinders the exchange of information among various organizations and participants within the expansive Industry 4.0 vision.

Consequently, a need for the establishment of a *global ontology* tailored to the Industry 4.0 domain arises. This global ontology should serve as a unifying framework, providing a standardized representation of various domains of manufacturing data (Cuenca et al., 2020a). Global ontologies are points of reference when designing application-specific ontologies to be deployed across the various application contexts (Olivé, 2017). A global ontology would enable interoperability in the Industry 4.0 landscape, wherein diverse manufacturing applications, with specialized requirements, cooperate to improve the product manufacturing process (Grangel-González et al., 2016).

A global ontology for the domain of Industry 4.0 must balance reusability and usability. On the one hand, this ontology must be reusable to support a wide spectrum of applications within the domain of concern (Morbach et al., 2009). To achieve this, the ontology should encompass abstract knowledge that finds utility across various applications (Cuenca et al., 2020a). However, the effort of extending and modifying the ontology knowledge to adapt it to specific applications is demanding. Therefore, the ontology must also provide usability, mitigating the effort required to adapt the knowledge to the requirements of each application. To address this aspect, ontology knowledge must exhibit granular specificity (Morbach et al., 2009). It is essential to recognize that ontology reusability and usability inherently have opposite objectives, necessitating the design of an Industry 4.0 global ontology that deftly balances these concepts.

In this regard, layered ontologies emerge as the primary solution to balance reusability and usability, exemplified by ontologies such as the DABGEO ontology tailored to the energy domain (Cuenca et al., 2020a). Layered ontologies classify the knowledge into various abstraction layers. These layers effectively segregate the knowledge that has broad relevance in most applications, denoted as *common domain knowledge*, from the knowledge that serves the needs of specific applications, termed as *variant domain knowledge*. Furthermore, knowledge is divided into modules, each addressing a well-defined segment of the data domain. This structural arrangement empowers ontology engineers to selectively reuse pertinent knowledge within the various application contexts, minimizing efforts (Cuenca et al., 2020a).

1.2. Contribution

This paper presents and describes I40GO, an Industry 4.0 Global Ontology. I40GO is openly accessible and can be downloaded from its home

page.¹ The ontology is licensed under Creative Commons Attribution 4.0.² I40GO restructures and classifies the knowledge of available Industry 4.0 ontologies in different layers and modules.

The primary objective of I40GO is to act as a cohesive force, offering a consolidated knowledge base from existing Industry 4.0 ontologies. This consolidation facilitates smooth data integration, promotes compatibility across applications, and eases the development of ontologies tailored to specific subdomains within Industry 4.0.

Additionally, I40GO aligns with the FAIR principles (Findability, Accessibility, Interoperability, and Reusability) (Wilkinson et al., 2016). This alignment is demonstrated by including standardized metadata and providing comprehensive documentation for enhanced findability, the use of open-access repositories for accessibility, the adherence to semantic web standards for interoperability, and the adoption of a modular structure that promotes reusability. This alignment makes I40GO a valuable resource, facilitating the development of domain-specific ontologies and advancing knowledge sharing and collaboration within the Industry 4.0 domain. Furthermore, this paper showcases I40GO's practical application in an Industry 4.0 case study, demonstrating its reusability and highlighting its benefits to manufacturing systems.

This paper is structured as follows. In Section 2, I40GO is positioned with respect to current Industry 4.0 ontology-based solutions. Section 3 outlines the methodology for developing I40GO. Section 4 describes the content and structure of I40GO. Section 5 provides a case study in which I40GO was applied. Section 6 enumerates the limitations of this study. Section 7 summarizes the study conclusions, highlighting the main benefits of I40GO, as well as future work lines.

2. Related work

An analysis of ontology-based solutions developed in the Industry 4.0 domain is presented in this section. It also positions I40GO with respect to them. These solutions include 1) ontologies developed for Industry 4.0 solutions and 2) layered ontology architectures defined to develop reusable ontologies in the industry domain.

2.1. Industry 4.0 ontologies

Several ontologies serve as the knowledge base of manufacturing systems. The MASON ontology, introduced by Lemaignan et al. (2006), provides an upper ontology that establishes a common representation for manufacturing systems. It targets applications such as manufacturing cost estimation systems or multi-agent manufacturing systems. Lu and Xu (2017) presented the ManuService ontology, which serves as the knowledge base for a cloud manufacturing system, facilitating the mapping of customer requirements to manufacturer resources for efficient product manufacturing. Mesmer and Olewnik (2018) presented the PMPO ontology, designed to assist users with limited knowledge of manufacturing processes in discovering potential product manufacturers. Ehm et al. (2019) introduced the Digital Reference ontology, tailored for use in digital production, supply chain networks, and product life-cycle management. Similarly, Wawrzik et al. (2023) presented the GENIAL Basic Ontology, a common vocabulary that enables data exchange between microelectronic devices.

In addition to the ontologies used in manufacturing systems, since the "Industry 4.0" term was coined, several ontologies have been developed to semantically represent Industry 4.0 standards. Grangel-González et al. (2016) introduced the RAMI 4.0 vocabulary. The purpose of the RAMI 4.0 ontology is to serve as a semantic reference model used by intelligent devices to exchange data, enabling a self-organized and resilient product manufacturing process. Bader et al. (2020) presented I40KG, a knowledge graph for Industry 4.0-related standards.

¹ <http://www.purl.org/i4go>

² <https://creativecommons.org/licenses/by/4.0/>

Kovalenko et al. (2015) introduced the AutomationML ontology, which is based on the AutomationML standard of Industry 4.0 presented by Drath et al. (2008). The purpose of the AutomationML ontology is to support data analysis activities across the discipline/tool boundaries in production system engineering. It represents knowledge about the production systems participating in the manufacturing process, including functions and roles.

Finally, ontologies tailored for specific manufacturing contexts, such as chemical engineering and pharmaceutical plants, have also been developed. Morbach et al. (2009), Natarajan et al. (2012), Muñoz et al. (2010) have contributed to ontologies for manufacturing process management systems in the field of chemical engineering. These refer to processes that involve the conversion of chemicals, materials, and energy into products within chemical plants, e.g., manufacturing of pharmaceutical products. Morbach et al. (2009) introduced the OntoCAPE ontology, a formal ontology specified for computer-aided process engineering. Natarajan et al. (2012) presented the OntoSafe ontology, developed as a knowledge base for process supervision systems in chemical plants. Muñoz et al. (2010) introduced the BaPrOn ontology, designed to provide a robust and transparent framework for integrating batch-related information about chemical product manufacturing. Furthermore, Hailemariam and Venkatasubramanian (2010), Sesen et al. (2010) introduced ontologies for manufacturing management systems for pharmaceutical plants. Hailemariam and Venkatasubramanian (2010) presented the POPE ontology, specifically tailored for the pharmaceutical process domain within process engineering. Sesen et al. (2010) introduced the OntoReg ontology, aiming to support decision-making in regulatory compliance, enabling automated and intelligent validation of pharmaceutical products against existing regulatory requirements and standards.

An overview of these ontologies is presented in Table 1. This table summarizes their scope, key features, and respective levels of interoperability and reusability. The “Scope and Key Features” column describes the intended application and knowledge domain targeted by each ontology. The columns “Interoperability” and “Reusability”, classify the ontologies based on their ability to enable data interoperability across applications and facilitate knowledge reuse in various contexts. Domain-specific ontologies generally offer lower levels of interoperability and

reusability due to their focus on a narrow subject matter (Gómez-Pérez et al., 2006; Uschold & Gruninger, 2004), while multi-domain ontologies and global or universal ontologies are designed to operate across a broader range of systems and contexts, providing higher levels of both interoperability and reusability (Choi et al., 2006; Song et al., 2017).

These terms are not tied to specific individuals or organizations that first defined them. Although, these terms are widely used in the field of ontology and knowledge engineering to describe different types of ontologies based on their scope and purpose. They have become standard terminology in the ontology community, and their definitions have evolved over time through collective usage and contributions from researchers and practitioners in the field.

- Domain-specific or Application-specific ontologies: Designed to represent knowledge within a specialized domain or field (Guarino, 1998; Navigli & Velardi, 2004). They focus on semantics that are unique to a particular domain. The practical applicability of these ontologies is high within specific systems and applications.
- Multi-domain ontologies: Aimed to represent knowledge from multiple related domains or subdomains within a field (Song et al., 2017). They often include concepts and relationships that bridge different subdomains, i.e. bio-informatics ontologies, which combine biology and informatics.
- Global or Universal ontologies: These are upper-level ontologies designed to provide a foundational representation of knowledge that spans diverse domains and disciplines (Abdoulleev, 2008; Choi et al., 2006). Typically serving as the basis for other ontologies.

The ontologies discussed in this section each address a portion of the Industry 4.0 domain, employing distinct vocabularies. Despite their shared focus on particular subdomains or knowledge areas, they lack a clear differentiation between common and application-specific knowledge.

I40GO was specifically developed to bridge these issues. It aims to serve as a general-purpose ontology, offering a unified representation of manufacturing data domains by integrating knowledge from the reviewed ontologies. In contrast to the analyzed ontologies, I40GO employs a layered ontology approach, which classifies knowledge into different abstraction layers. This structure separates broadly applicable

Table 1
Ontologies for manufacturing systems.

Ontology	Scope and Key Features	Interoperability		Reusability	
		Domain-specific	Multi-domain	Domain-specific	Multi-domain
Lemaignan et al. (2006) MASON	Knowledge base for manufacturing cost estimation systems and multi-agent manufacturing systems.	✓	×	×	✓
Lu and Xu (2017) ManuService	Knowledge base for cloud manufacturing systems aiming to map customer requirements to manufacturer resources.	✓	×	×	✓
Mesmer and Olewnik (2018) PMPO	Designed to help users with limited knowledge discover potential manufacturers for their products.	✓	×	×	✓
Ehm et al. (2019) Digital Reference	Targeted at digital production, supply chain networks, and product life-cycle management.	×	✓	×	✓
Wawrzik et al. (2023) GENIAL Basic	A common vocabulary facilitating data exchange between microelectronic devices.	✓	×	✓	×
Grangel-González et al. (2016) RAMI 4.0	Semantic reference model for intelligent devices to exchange data, enabling self-organized manufacturing.	×	✓	✓	×
Bader et al. (2020) I40KG	A knowledge graph for Industry 4.0-related standards.	✓	×	✓	×
Kovalenko et al. (2015) AutomationML	Supports data analysis across production system engineering disciplines.	×	✓	×	✓
Morbach et al. (2009) OntoCAPE	Developed for computer-aided process engineering in the chemical engineering field.	✓	×	×	✓
Natarajan et al. (2012) OntoSafe	The knowledge base for process supervision systems in chemical plants.	✓	×	×	✓
Muñoz et al. (2010) BaPrOn	Designed to integrate batch-related information about chemical product manufacturing.	✓	×	×	✓
Hailemariam and Venkatasubramanian (2010) POPE	An ontology tailored for the pharmaceutical process domain within process engineering.	✓	×	×	✓
Sesen et al. (2010) OntoReg	Aims to support decision-making in regulatory compliance for pharmaceutical products.	✓	×	×	✓

common domain knowledge from variant domain knowledge tailored to specific applications. This modularity facilitates easier reuse and adaptation of ontology components for various manufacturing applications by just selecting the specific knowledge required (Cuenca et al., 2020a). Depending on the application to be developed, developers can use and adapt the ontology modules containing abstract knowledge or those with more specific knowledge.

2.2. Layered ontology architectures

Several proposals have emerged in recent years for layered ontology architectures to create reusable and interoperable ontologies in the Industry 4.0 context.

The Industrial Ontologies Foundry (IOF) is an organizational unit managed by OAGi, a US-based non-profit standards organization (Karray et al., 2021). IOF aims to create reference ontologies for the industry and proposes a set of principles and guidelines to develop reusable and interoperable ontologies. One of these principles is to classify the ontologies into different abstraction levels (Karray et al., 2021):

- **General ontologies:** ontologies that represent general concepts that are domain-independent. For instance, the Basic Formal Ontology (BFO) is a general ontology developed to support data retrieval, analysis and integration across different domains (Otte et al., 2022).
- **Domain-upper ontologies:** ontologies representing common and relevant domain knowledge.
- **Domain-specific ontologies:** ontologies that represent more specific knowledge that is not common for all the applications of the domain concerned.
- **Application ontologies:** ontologies that represent domain knowledge used only in specific applications.

Beyond the Industrial Ontologies Foundry, another significant initiative focused on developing interoperable and standardized ontologies is the OntoCommons project.³ OntoCommons is an H2020 CSA project focused on developing interoperable and standardised ontologies to be used by manufacturing systems. OntoCommons proposes a system of interoperable ontologies classified into the following levels (Magas et al., 2024):

- **Top-level ontologies:** ontologies that include general concepts that are common to all application domains (i.e., ontologies for units of time and measures).
- **Middle-level ontologies:** ontologies that include general concepts of one or more specific domains (i.e., manufacturing, materials science).
- **Domain-level ontologies:** ontologies used by a specific domain of application (i.e., additive manufacturing).
- **Application-level ontologies:** ontologies reused by specific applications.

The domain knowledge of I40GO is classified according to the MODDALS methodology guidelines (Cuenca et al., 2020b). MODDALS is employed to design and develop layered ontologies that can be reused in various applications. Following this methodology, I40GO classifies the domain knowledge into the following levels of abstraction: 1) Common and relevant knowledge of the industry domain, 2) Variant-domain knowledge that is not common but recurrent across the domain and 3) knowledge that is reused by specific Industry 4.0 applications. Therefore, I40GO domain knowledge follows a classification similar to the architectures proposed by ontology standardisation initiatives for industry.

However, since I40GO is expected to be reused specifically within the Industry 4.0 domain, its current version does not include top-level ontologies that represent domain-independent knowledge.

3. Methodology

The I40GO development team included both ontology engineers and engineers with expertise in the Industry 4.0 domain.

The development process of I40GO was guided by the following requirements:

- The ontology should unify the knowledge of the existing Industry 4.0 ontologies.
- The ontology should classify the common and variant domain knowledge into different layers.

Bearing this in mind, I40GO was developed following the guidelines and tools proposed by the NeOn methodology (Suárez-Figueroa et al., 2012). NeOn defines a set of flexible scenarios for developing ontologies and ontology networks, facilitating ontology reuse. I40GO also follows the main steps of Linked Open Terms (LOT) methodology (Poveda-Villalón et al., 2022). LOT is based on NeOn, and is a lightweight methodology for developing ontologies and vocabularies focused on industrial projects.

In addition, the MODDALS methodology (Cuenca et al., 2020b) was applied to design I40GO and classify its domain knowledge. MODDALS adheres to well-established ontology engineering principles. Layered ontologies exhibit the same structure as Software Product Lines (SPLs), which are described as “software families that contain common reusable parts and variable parts that depend on specific customer needs” (Cuenca et al., 2019). Consequently, MODDALS leverages established SPL development techniques. The input to MODDALS comprises the existing ontologies in the relevant domain. Specifically, the ontologies analyzed in Section 2 served as a reference for designing and developing I40GO. That is, RAMI 4.0, MASON, Digital Reference, among others. The output is an ontology that incorporates the knowledge from existing ontologies, systematically classified into various layers.

The steps undertaken to develop I40GO are outlined in Fig. 1. In the subsequent subsections, we provide a summary of the steps taken to develop I40GO.

Step 1: Ontology requirement definition

I40GO’s purpose and scope were defined through collaboration between domain experts and ontology engineers. The main goal of I40GO is to provide a standardized and unified knowledge representation of the Industry 4.0 domain, which is currently represented by heterogeneous ontologies, with a focus on maintaining a moderate level of reuse effort. In addition, high-level requirements for the ontology were established, emphasising common knowledge representation and abstraction layering.

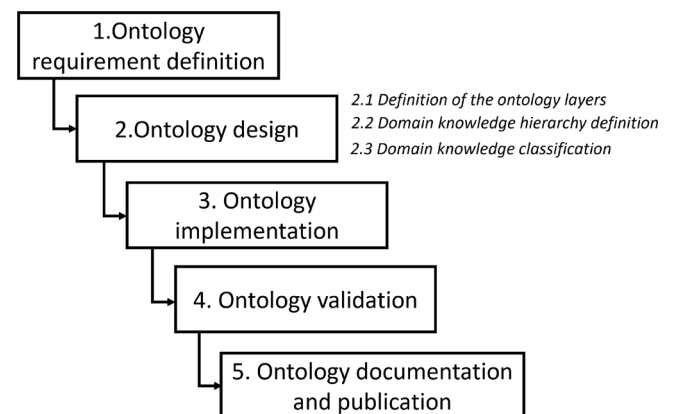


Fig. 1. I40GO development steps based on NeOn and MODDALS methodologies.

³ <https://ontocommons.eu/>

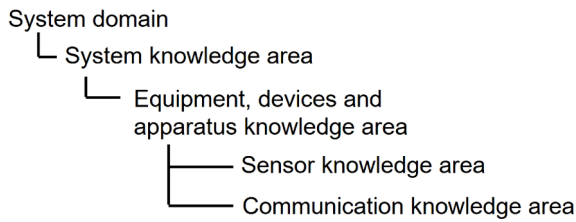


Fig. 2. Part of the knowledge hierarchy of I40GO.

Step 2: Ontology design

The layered structure of I40GO was designed following the steps outlined in the MODDALS methodology.

Step 2.1: Definition of the ontology layers

The I40GO ontology layers were defined in alignment with the layered ontology structure proposed by MODDALS. In particular, I40GO consists of three layers:

1. The *Common-domain* layer includes the domain knowledge employed by the majority of the Industry 4.0 ontologies taken as a reference to develop I40GO.
2. The *Variant-domain* layer comprises knowledge reused in a subset of Industry 4.0 ontologies.
3. The *Domain-task* layer encompasses domain knowledge reused in specific Industry 4.0 ontologies.

For a more detailed exploration of the concepts and knowledge contained within each layer, refer to [Section 4](#).

Step 2.2: Domain knowledge hierarchy definition

The domain knowledge within the I40GO ontology was structured as a knowledge hierarchy. In this hierarchy, the domains represented by the Industry 4.0 reference ontologies were divided into more specific knowledge pieces referred to as knowledge areas (KAs). KAs are a set of related ontology elements that have the potential to serve as modules, collectively representing focused domain topics (Cuenca et al., 2020b). Within the knowledge hierarchy, KAs are linked through parent-child relationships, where “child” KAs include and extend the knowledge from their “parent” KAs. Hence, KAs situated at lower hierarchical levels include and extend the knowledge from their higher-level counterparts. An illustrative part of the I40GO knowledge hierarchy is presented in [Fig. 2](#).

The identification of ontology domains and the extraction of KAs were accomplished through collaborative efforts involving domain experts and ontology engineers. This process entailed a manual analysis of the elements that constitute existing Industry 4.0 ontologies to identify the data domains they represent. For instance, within the Industry 4.0 ontologies, the “*System domain*” was recognized, which includes data about physical or abstract systems, mainly devices.

During the final activity of this step, the knowledge of each layer was classified into the various ontology modules, with one module allocated

for each KA. The modules were placed within the appropriate ontology layer as determined in Step 3, adhering to the knowledge hierarchy established in Step 2.

Once the ontology domains were defined, an exhaustive analysis of the class hierarchies and properties within existing ontologies enabled the extraction of KAs. For instance, in the illustrated example within [Fig. 2](#), the “*System domain*” incorporates the “*System KA*”, which contains knowledge concerning technical and network systems used in manufacturing management applications. Extending this further, the “*Equipment, devices, and apparatus KA*” builds upon knowledge within the “*System KA*”, encompassing information about devices or apparatus, whether they constitute the primary devices within an installation or all the devices required for a specific task. Moreover, the “*System domain*” features more specific KAs that further extend the knowledge about equipment, devices and apparatus, including *Sensor* or *Communication* KAs. These KAs include knowledge about sensors and communication devices respectively.

Step 2.3: Domain knowledge classification

The KAs defined in the previous step were systematically classified into their respective layers. To determine the extent to which each KA was represented by existing Industry 4.0 ontologies, a meticulous analysis was conducted. We considered a Knowledge Area to be represented by an ontology, if the ontology included relevant classes or properties.

The classification process entailed the application of the *application-knowledge matrix* technique, which facilitated the categorization of KAs into common and variant domains based on their representations across ontologies (Cuenca et al., 2020b). For each Industry 4.0 domain, a dedicated application-knowledge matrix was constructed. For instance, [Table 2](#) exemplifies an application-knowledge matrix encompassing a selection of KAs within the system data domain. The top row encompasses the Industry 4.0 ontologies, while the left column itemizes the KAs. To simplify the table, several ontologies were omitted.

The classification of KAs into common and variant domains was guided by their Common Variant ratio (CV ratio), as indicated in the right column of [Table 2](#). The CV ratio is defined as the ratio between the number of ontologies representing a KA and the total number of ontologies. The threshold value of the CV ratio was defined in compliance with MODDALS guidelines. MODDALS establishes that the CV ratio threshold must be defined following the domain experts’ criteria, depending on the number of analysed ontologies. Following this criteria, a threshold value of 60 % was employed to establish KA classification in I40GO ontology. KAs with a CV ratio equal to or greater than 60 % were placed in the *Common-domain layer*, while those with a CV ratio below 25 % were placed in the *Domain-task layer*. KAs falling within the remaining range were categorized under the *Variant-domain layer*. Notably, the knowledge classification derived from this process underwent validation by domain experts to rectify any potential misplacements within layers.

Step 3: Ontology implementation

At this step, the ontology modules were developed, with ontology engineers carefully selecting elements from existing Industry 4.0 ontologies for reuse. The knowledge from existing Industry 4.0 ontologies was

Table 2
Application-knowledge matrix of the system domain.

Knowledge Areas	Ontologies				CV ratio
	RAMI 4.0 ontology	AutomationML ontology	MASON ontology	Digital Reference ontology	
System	✓	✓	✓	✓	100 %
Equipment, devices and apparatus	✓	×	✓	✓	75 %
Sensor	✓	×	×	✓	50 %
Operation	✓	×	×	✓	50 %
Communication	×	×	×	✓	25 %

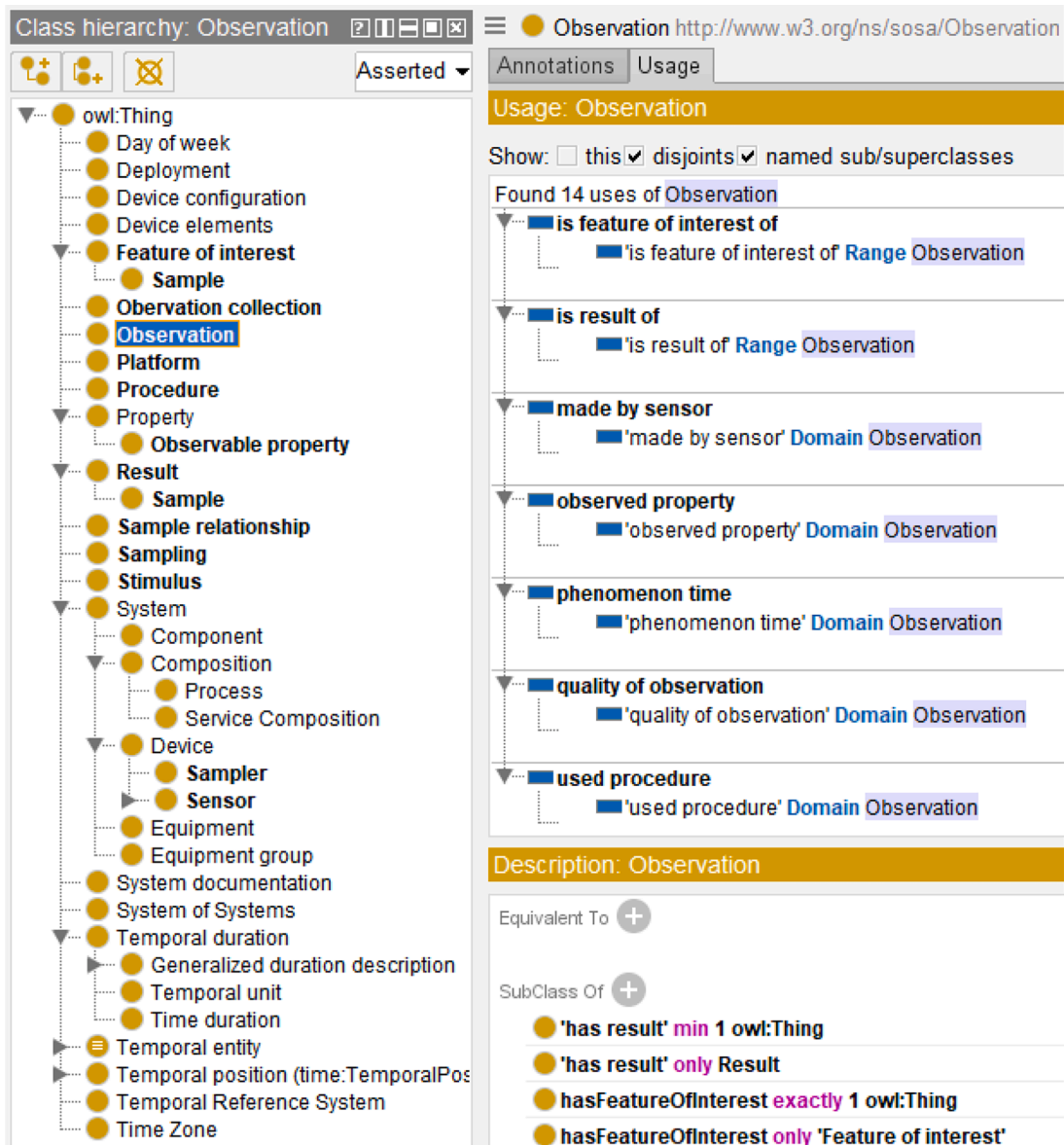


Fig. 3. Deployment of updated I40GO with SOSA classes and properties in Protégé.

reused and implemented into a layered structure, which was guided by the MODDALS methodology. The modules were implemented utilizing Protégé (Musen, 2015), a widely recognized ontology editor for building intelligent systems.

The creation of each module involved the incorporation of classes and properties from ontologies representing the corresponding KA. To determine the base ontology for each module, the one featuring the most comprehensive array of classes and properties relevant to the concerned KA was chosen. In addition, the consideration of ontologies adhering to existing standards played a role in this selection process. The knowledge encapsulated within the chosen ontology served as the base to develop the corresponding module, followed by the integration of classes and properties from other ontologies that had similar knowledge pertinent to the KA. For instance, in the case of the DigitalReference ontology, which uses the SSN/SOSA standard for sensor data representation, the sensor module was augmented with classes and properties derived from the SSN / SOSA ontology (e.g. observation and sample) and deployed in Protégé as illustrated in Fig. 3.

The acquired knowledge was extended with the classes and properties from the DigitalReference ontology. For instance, within DigitalRef-

erence, specific properties were available for defining parameters such as the maximum and minimum temperature of a Sensor. These properties were incorporated into the sensor ontology module.

Step 4: Ontology validation

The I40GO module validation process involved using well-known ontology validation tools. Syntax and structural assessments were performed using the OOPS! Pitfall Scanner (Poveda-Villalón et al., 2014), a tool designed to identify common pitfalls encountered during ontology development.

Step 5: Ontology documentation and publication

For user-friendly documentation, the Widoco⁴ ontology documentation tool was employed to generate readable documentation for I40GO. Furthermore, to enhance accessibility, the I40GO modules, along with

⁴ <https://github.com/dgarjio/Widoco>

their accompanying documentation, were published on an Apache server, ensuring online availability of the ontology.

In summary, the development of the I40GO ontology followed a structured methodology that combined the principles of the NeOn and MODDALS methodologies. This approach allowed the creation of a layered ontology that unifies knowledge from diverse Industry 4.0 ontologies while carefully distinguishing between common and variant domain knowledge. The knowledge hierarchy, established in the design phase, served as a reference point for module development. Leveraging existing ontologies and standards, the modules were meticulously crafted and rigorously validated. Human-oriented documentation was generated to facilitate understanding and usage. The resultant I40GO ontology not only achieves its goal of harmonizing knowledge representation across the Industry 4.0 landscape but also aligns with established validation and documentation standards, ensuring its accessibility and utility.

4. I40GO overview

The content and structure of I40GO (current version 1.0) are described in this section. I40GO comprises 29 modules implemented in OWL-2 DL, the standard ontology language (Consortium et al., 2021). Classes, object properties, and data properties were used to model domain concepts, relations, and attributes, respectively. In total, I40GO encompasses 1104 classes, 481 object properties, and 246 data properties, meticulously capturing the concepts and relations of the Industry 4.0 domains.

Given the large-scale nature of I40GO, a detailed description of every class and property is beyond the scope of this section. Therefore, this section offers a high-level overview of I40GO's structure and content. The knowledge description for each I40GO module can be found on the I40GO home page.⁵ Each module was properly published and documented, adhering to the best practices for ontology documentation.

4.1. I40GO knowledge

I40GO serves as a unifying force, providing a common representation of data domains identified within existing ontologies developed in the Industry 4.0 context. Specifically, it reuses the knowledge embedded in the ontologies reviewed in Section 2, including MASON, Digital Reference, GENIAL, RAMI 4.0, I40KG, AutomationML, and OntoCAPE. Furthermore, these ontologies themselves reuse knowledge from standard ontologies that are applicable across a wide range of application domains, such as:

- SSN/SOSA ontology (Janowicz et al., 2019): Describes sensors and actuator data, that is, sensor observations and samples.
- SOA ontology:⁶ Focused on Service-Oriented Architecture (SOA), this ontology provides vocabularies to service-oriented applications and systems.

The vocabularies used within I40GO serve as reusable building blocks for the development of application-specific ontologies for specific manufacturing applications. I40GO covers the following data domains within Industry 4.0:

- **System domain:** Within this domain, I40GO encompasses data concerning physical or abstract systems, with a primary focus on devices. This includes data pertaining to various types of devices, such as sensors, actuators, and communication devices, along with the coverage of software systems. The key classes within I40GO representing this domain encompass:
 - System: Utilized to capture the essence of a set of interrelated elements, including technical systems and network systems.

- Device: Encompassing any material element or assembly designed to perform specific functions.
- Equipment: Used to represent apparatus or sets of devices, these entities encompass the primary devices constituting an installation, or the full complement of devices essential for executing a particular task.
- Service: Designed to encapsulate data concerning the functionalities furnished by devices.
- **Manufacturing domain:** This domain revolves around the actors participating in Industry 4.0 manufacturing processes, ranging from products, suppliers, and customers. It also includes data about product suppliers and consumers, organizations, production plants, and products supplied to customers. The key classes of I40GO within this domain encompass:
 - Administration Shell: Utilized to capture assets in electronic form, following the RAMI 4.0 standards for seamless integration.
 - Industry 4.0 component: Encompassing systems that exchange data within the Industry 4.0 landscape.
 - Manufacturing concept: Enclosing data pertinent to the resources deployed during the manufacturing process, including tools and human resources.
 - Organization: Designed to represent the various organizations involved in manufacturing processes.

4.2. I40GO structure

The structure of I40GO is exemplified through the *System domain*. Details on the representation of other domains are available on the I40GO home page. Fig. 4 provides a detailed overview of I40GO structure, with a specific focus on the *System domain*.

The modules of I40GO are classified into three abstraction layers, each encapsulating knowledge specific to its layer, as explained in the following subsections.

4.3. Common-domain layer

The *Common-domain layer* serves as a repository for domain knowledge that is ubiquitous across the majority of Industry 4.0 ontologies. For instance, the *System ontology* module includes information about systems. In the context of the Industry 4.0 Global Ontology, a system is considered a set of interrelated elements (e.g., technical systems, network systems) that, within a defined context, are viewed as a whole and separated from their environment. A system is generally defined with the view of achieving a given objective, for example by performing a definite function. The *System ontology* module describes system features like their components (e.g., devices), activities (e.g., real-time energy measurements of plant equipment), and purpose (e.g. analyzing production plant energy consumption). In a similar vein, the *Equipment, devices, and apparatus ontology* encloses knowledge pertaining to devices employed in diverse manufacturing processes (i.e., sensors, communication devices, software systems). The knowledge within this layer is fundamental, as it involves the basic concepts of most Industry 4.0 ontologies and finds applicability in many Industry 4.0 contexts. Therefore, the modules within this layer encompass abstract concepts and relations within the domain, although with a lesser degree of detail compared to modules within the subsequent layers (Morbach et al., 2009).

4.3.1. Variant-domain layer

The *Variant-domain layer* hosts variant domain knowledge that is recurrent across more than one Industry 4.0 ontology. The modules of this layer “*extend and include the knowledge of the common-domain layer, since they include more specific concepts, relations, and axioms*” (Cuenca et al., 2020a). For instance, the *Sensor ontology* module delves into the aspects of sensor devices and their operational characteristics, that is, measurement data. The designation of these data within

⁵ <http://www.purl.org/i4go>

⁶ <https://www.opengroup.org/soa/source-book/ontologyv2/p1.htm>

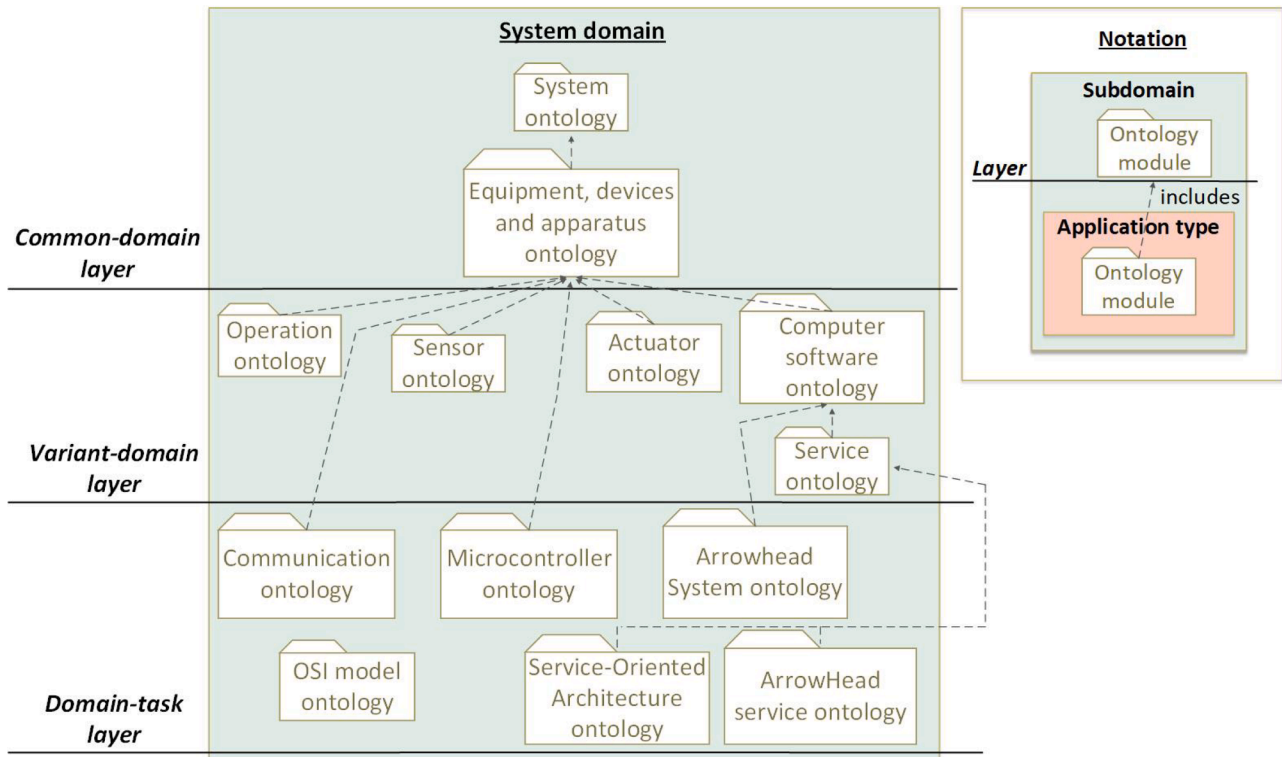


Fig. 4. Structure of the I40GO ontology (System domain).

the *Variant-domain layer* signifies a higher level of specificity. Similarly, the *Operation ontology* module captures knowledge pertaining to the operational activities of devices essential for installation and proper functioning. These operations encompass a wide range of activities, including monitoring, maintenance, and various work-related tasks.

Placing data (e.g. sensor data) in this layer does not diminish its significance within the domain. The *Common-domain layer* serves as the repository for the most general concepts throughout the domain, which are virtually ubiquitous across all ontologies. Conversely, the *Variant-domain layer* accommodates more specialized concepts that, while not universally applicable, remain important and find utility in various applications.

4.3.2. Domain-task layer

The *Domain-task layer*, on the other hand, is exclusively reserved for the variant knowledge that is reused solely within specific Industry 4.0 ontologies and applications. Modules within this layer extend the knowledge that resides in higher-level layers, integrating the vocabularies required by specific applications. For instance, the *Communication ontology* module represents knowledge about devices used for information exchange, adhering to predefined conventions. In a similar vein, the *Service-Oriented Architecture ontology* module encompasses concepts related to Service-oriented architecture (SOA). In the context of the I40GO ontology, SOA embodies a software design paradigm where application components provide services to the other components, through a communication protocol operating over a network.

With the overview of I40GO's structure and its layered design, a foundation is established for its application in various Industry 4.0 contexts. The subsequent section presents a practical case study showcasing I40GO's real-world utility within an architecture for context-aware workflow management. This case study highlights how I40GO integrates with manufacturing systems, enhancing interoperability, decision-making, and context-awareness in the Industry 4.0 landscape.

5. I40GO usage: An architecture for context-aware workflow management

In this section, I40GO plays a pivotal role within an architecture for context-aware workflow management developed by Mondragon Unibertsitatea. The architecture is described by setting a special focus on its context-aware component, where I40GO is reused.

The architecture responds to the growing need for innovation in the realm of Industry 4.0, where manufacturing processes face a spectrum of challenges, spanning from the coordination of diverse tasks across the production floor (Chen & Tu, 2009) to the dynamic adaptation of processes in real-time (Leitão & Restivo, 2006). To address these challenges, the architecture is constructed using industry standards for asset interoperability and workflow orchestration. Additionally, it places emphasis on context analysis, a critical aspect in identifying unforeseen situations and devising strategies to address them (Alexopoulos et al., 2016). This architectural framework leverages the capabilities of semantic web technologies to empower context-awareness, enabling real-time reconfiguration of manufacturing processes.

The architecture orchestrates machine functionalities by integrating the Asset Administration Shell as the RAMI 4.0 implementation, alongside business process modeling software for the design and execution of workflows. Although the initial version of this architecture was introduced in Ochoa et al. (2023), the current iteration incorporates a context-aware component designed to enhance the reactivity and flexibility of manufacturing operations during runtime. The various components of this architecture and their relationships are illustrated in Fig. 5. Furthermore, Table 3 briefly describes each component.

Semantic context analysis is handled by the Context Analyzer component, which employs semantic web technologies for describing machines, sensors, administration shells, and services. It stores and analyzes context data concerning the current machine states, enabling recommendations for device and service replacement within workflows. This component integrates elements from Recommendation Systems

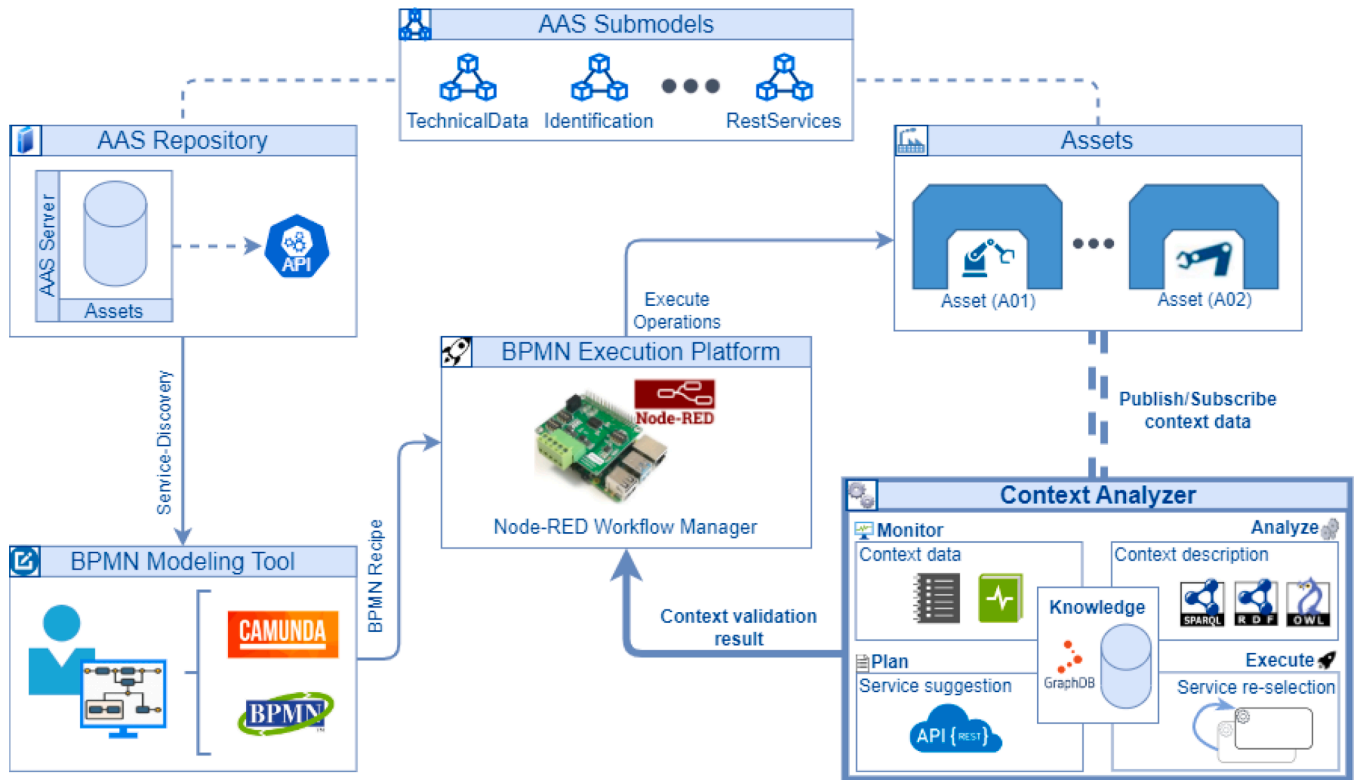


Fig. 5. Architecture for context-aware workflow management: An asset administration shell-based approach. Extended from Ochoa et al. (2023).

Table 3
Components of the architecture for context-aware workflow management.

Component	Description
Assets	Represents I4.0 digitized machines and sensors located on the production floor. These physical devices are digitized through AAS, turning them into I4.0 digital components (4.0, 2021). Assets are then grouped into administration shells and stored in the AAS Repository.
AAS repository	Contains the AAS Server and database that stores administration shell data that can be queried and maintained using an API (Application Programming Interface). Several AAS implementations exist as of this writing including Basyx (Basyx, 2022), NOVAAS (Di Orio et al., 2019), Admin-shell-io (AASX Server). However, Basyx is selected for this application case due to its robustness.
AAS submodels	Allows describing machines and their services through the use of <i>TechnicalData</i> , <i>Identification</i> , and <i>RestServices</i> submodels (Elektroindustrie, 2018). The latest characterizes properties of REST services such as URL, Name, Method, IsAsync, RequestBody, and Response.
BPMN modeling tool	Includes both, Camunda Modeler and a <i>AAS Web Service Discoverer</i> plugin that enables Camunda Modeler to discover services from the AAS Repository. By using these tools, users can design manufacturing business processes out of asset services using Business Process Model and Notation (BPMN), a standardized notation language for the design of workflow diagrams (OMG, 2022).
BPMN execution platform	Comprehends our <i>Node-RED Workflow Manager (Node-RED WM)</i> presented in Larrinaga et al. (2022). <i>Node-RED WM</i> interprets and runs workflow recipes written in BPMN. This component was thought out to allow service orchestration at the edge environment. Thus, it can be installed in embedded systems since it requires low resources.
Context analyzer	Enables dynamic workflow reconfiguration at runtime. <i>Context Analyzer</i> relies on semantic web technologies for the correct representation of context information. It includes <i>DeviceServiceOnt</i> , an application-specific ontology that reuses I40GO to represent machines, sensors, services, and quality.

(RS) to offer suggestions for such replacements (Zammali & Ben Yahia, 2021). RS utilizes intelligent algorithms to analyze preferences, behaviours, and the current state of involved entities as discussed by Jan-nach et al. (2010).

The Context Analyzer component includes an application-specific ontology named “DeviceServiceOnt”, which reuses I40GO. In this application case, the DeviceServiceOnt ontology is imported into a semantic repository⁷ (GraphDB⁸). This integration allows the Context Analyzer component to continuously monitor the context. During runtime, this component gathers data from devices and sensors, storing it within the semantic repository. Knowledge is stored as triplets that encompass information about these devices, their sensors, and their services. This architectural design follows the principles of MAPE-K (Monitor, Analyze,

Plan, Execute, and Knowledge), similar to the approach presented in Ghobaei-Arani and Shahidinejad (2022). Subsequent subsections elaborate on the development and utilization of DeviceServiceOnt.

5.1. DeviceServiceOnt development process (reusing I40GO)

The development of DeviceServiceOnt involved collaboration between ontology engineers and web engineers. This ontology was created leveraging modules and classes from I40GO and implemented using Protégé. The development process can be outlined in the following key phases.

Phase 1. Ontology requirement definition: The initial step involves defining functional ontology requirements. The ontology encompasses concepts such as Administration Shells, Devices, Services, and Quality, among others. To identify suitable classes for reuse in this application case, ontology engineers analyzed modules from the *Variant-domain*, *Common-domain*, and *Domain-task* layers. Table 4 summarizes the

⁷ <https://docs.camunda.io/docs/components/modeler/about-modeler/>

⁸ <https://graphdb.ontotext.com/>

Table 4
I4GO modules and classes reused by DeviceServiceOnt.

Layer	Module	Class	Description
Variant-domain	Sensor	Sensor	Constitutes sensors that belong to machines/devices.
Common-domain	Manufacture Equipment	Administration Shell Device	Represents machines/devices and their administration shells.
Domain-task	SOA	Service	Describes web services of machines/devices.

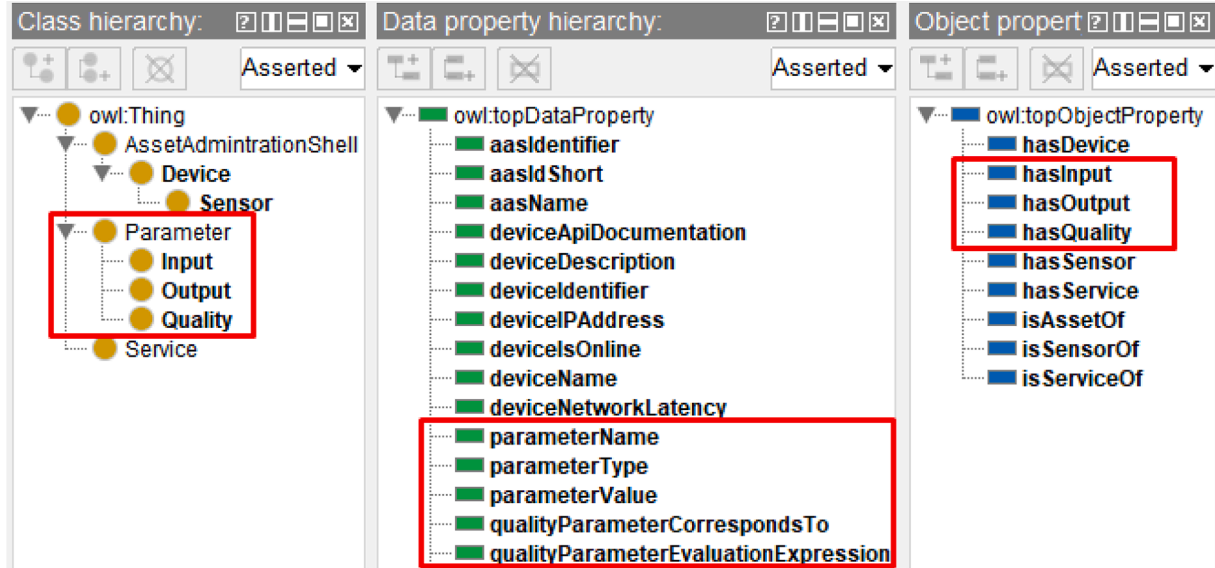


Fig. 6. New classes and properties added to DeviceServiceOnt ontology.

selected classes and modules from the sensor ontology,⁹ manufacture ontology,¹⁰ equipment ontology,¹¹ and soa ontology¹² for integration into the DeviceServiceOnt ontology. Thus, the integration process was carried out in Protégé by importing the modules and referencing the classes using the *owl:imports* statement.

Phase 2. Ontology reuse and re-engineering: In this phase, the reused I4GO modules are incorporated and adapted to align with the functional requirements of DeviceServiceOnt. Domain-specific classes, including Parameter, Input, Output, and Quality, are incorporated to match the needs of this application case, while existing classes such as AssetAdministrationShell, Device, Sensor, and Service remain unchanged. Fig. 6 illustrates the newly added classes and their properties highlighted in red. Furthermore, any unnecessary knowledge from the reused modules is pruned. This process results in the DeviceServiceOnt ontology, which is displayed in Fig. 7 using Protégé.

With the DeviceServiceOnt ontology thus developed and refined, the architecture was subsequently put to the test. Specifically, the architecture was tested on a manufacturing scenario to validate its context-awareness capabilities in re-selecting devices and services during runtime. The re-selection process uses the ontology and SPARQL for context analysis.

5.2. Context analysis process during runtime

Real-time information about machines and sensors on the production floor is continuously collected by the Context Analyzer component. During workflow execution, Node-RED WM communicates with Context-Analyzer to evaluate the current context and find devices or services

with the best quality values (also known as Quality of Service - QoS She et al., 2019) to perform a desired task. Node-RED WM is configured to always apply the recommendation for service/device replacement provided by the Context Analyzer, resulting in an automatic workflow re-configuration at runtime.

The evaluation process is described in Algorithm 1. Essentially, this component generates SPARQL queries based on device or service names and quality conditions in the form of conditional expressions. These queries are then dispatched to the semantic repository, and the result-set is returned to the Context Analyzer. In cases where a single instance meets all the quality conditions, the selection process concludes by returning that instance. However, when multiple instances satisfy the quality conditions, a sorting operation ensues. The determination of the best service/device relies on two evaluation strategies: (1) “the lower the quality value, the better” and (2) “the higher the quality value, the better”, contingent upon the conditional symbols (>, >=, <, <=) specified in the conditional expression. For instance, *HUMIDITY* <= 52 means that lower humidity is preferable, while *SuccessRate* > 90 implies a higher success rate is preferred. These evaluation strategies are enacted as sorting sub-queries, constructed based on priority and the conditional symbols within the conditional expression. These sorting sub-queries are appended to the main query and executed accordingly. Finally, the first instance returned in the resultset is deemed the best service/device.

5.3. Evaluation: Re-selecting robots in a manufacturing scenario

The evaluation scenario focuses on the runtime reconfiguration of manufacturing workflows. Reconfiguration involves selecting devices and services with the best quality values to perform a desired task.

The scenario consists of three Lego colour sorter robots (assets) that are operated through REST services and digitized using AAS. The digital representation of these assets is registered in the AAS Repository, which provides an API for querying and maintaining the assets. A

⁹ <http://www.purl.org/i4go/variant-domain/sensor>

¹⁰ <http://www.purl.org/i4go/common-domain/manufacture>

¹¹ <http://www.purl.org/i4go/common-domain/equipment>

¹² <http://www.purl.org/i4go/domain-task/soa>

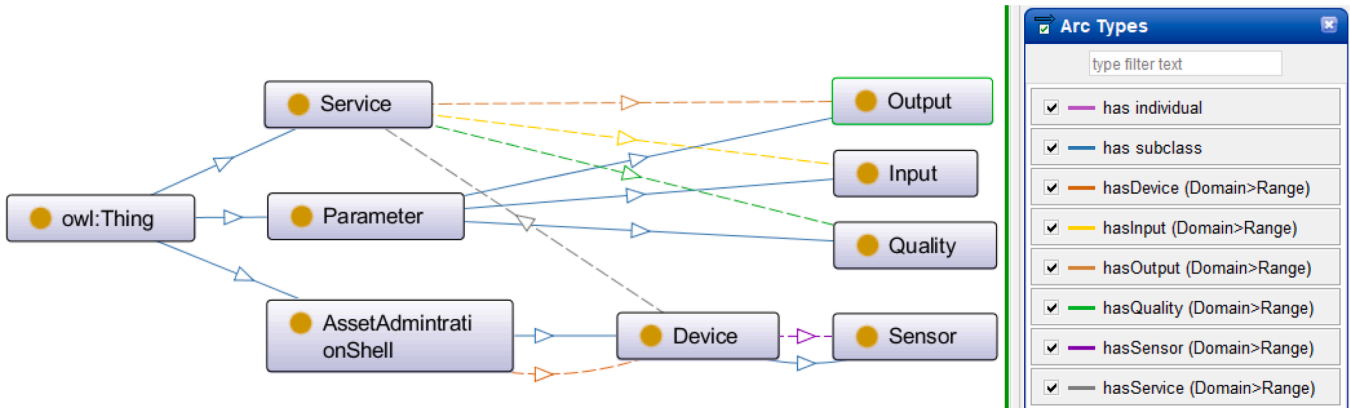


Fig. 7. DeviceServiceOnt ontology deployed in Protégé.

Algorithm 1 Context analyzer best service/device selection.

```

Require: Service Name and List of Quality Conditions
function SELECTBESTDEVICERVICE(ServiceName, QualityConditions)
  resultSet ← ∅
  while resultSet = ∅ and length(QualityConditions) ≥ 1 do
    query ← buildSparqlQuery(ServiceName, QualityConditions)
    sortingSubQuery ← empty
    for condition ∈ QualityConditions do
      symbol ← extractConditionalSymbol(condition)
      propertyName ← extractPropertyName(condition)
      sortingSubQuery ← sortingSubQuery + buildSortingSubquery(symbol,
        propertyName)
    end for
    query ← query + sortingSubQuery
    resultSet ← executeQuerySemanticRepository(query)
    if resultSet = ∅ then
      QualityConditions ← removeLatestCondition(QualityConditions)
    end if
  end while
  return getFirst(resultSet)
end function

```

manufacturing order is designed in BPMN format using the *BPMN Modeling Tool*, which allows importing assets' services through a service-discovery mechanism. The manufacturing order is uploaded to the *BPMN Execution Platform* component (Node-RED WM) and executed accordingly.

5.3.1. RDF representation of the Lego robot

The Lego robot used in this experiment has been semantically represented in RDF using GraphDB, as shown in Fig. 8. This representation showcases the classes and relations, including AssetAdministrationShell, Device, Sensor, Service, and Quality, defined previously. Among the services, notable examples include Move Left, Move Right, Throw Piece, and specialized services such as Classify Pieces. Among the sensors, there are instances such as Voltage, Vibration, and Accelerometer Sensors. These details are encoded in RDF triples and stored in GraphDB, offering a standardized and interoperable format for conveying the robot's essence.

5.3.2. Results

Following the acquisition of raw sensor data by the Context Analyzer component, a transformation process converts this data into semantically meaningful knowledge. This transformation relies on DeviceServiceOnt, which reuses modules and classes from I40GO. The resulting context data is semantically represented in the form of machine quality properties. This semantically enriched data is then stored in GraphDB and SPARQL queries can be executed.

Within the architectural framework, queries and the recommendation process are performed automatically. However, this exercise details

the possible situations in the selection process for the best device or service. These cases are influenced by the QoS-Aware service recommendation technique explained in Wang et al. (2010) and are:

Case 1. When a single device/service satisfies all QoS conditions, it is recommended for task execution.

Case 2. When multiple devices/services meet all QoS conditions, a sorting operation is employed to identify the device/service with the highest quality values. This entity is then recommended for task execution.

Case 3. When none of the devices/services fulfill all QoS conditions, but some partially meet these conditions, a sorting operation is executed. The device/service with the best quality values is recommended for task execution, accompanied by warnings.

Case 4. When none of the devices/services align with any QoS conditions, a sorting process identifies the device/service with the most favorable quality values. However, executing the task is not recommended, as it may not reach completion.

The aforementioned situations are tested using SPARQL queries, each providing insights into different decision-making situations. The SPARQL query for the first scenario is presented in Listing 1, and its outcomes are summarized in Table 5. The query results confirm that the selected device/service meets all specified quality conditions. As the chosen instance complies with all quality criteria, the workflow orchestrator can confidently employ it to execute the task, ensuring a high likelihood of task completion. The task completion probability is directly tied to the number of quality conditions satisfied. For example, if four conditions are met successfully, the task completion probability stands at 100%.

In the second scenario, as shown in Listing 2 and Table 6, the SPARQL query represents the case where numerous instances (devices/services) meet all quality conditions, resulting in a 100% task completion probability for all of them. To facilitate decision-making, the query incorporates a sorting statement ("ORDER BY"), yielding an ordered list of instances that considers both battery and temperature quality values. Ultimately, the first instance in this list is deemed the most suitable device/service, which the workflow orchestrator can utilize to execute the task.

Moving to the third scenario, Listing 3 represents the query when none of the devices/services fulfill all quality conditions. Consequently, the query must be iteratively adjusted to eliminate quality conditions one by one, starting with the lowest priority. This iterative process continues until at least one instance appears in the resultset (Table 7). In this context, the task completion probability becomes variable. For instance, if only two out of four conditions are met, the task completion probability is 50%, prompting the workflow orchestrator to proceed with task execution while being cautioned about potential uncertainties regarding completion.

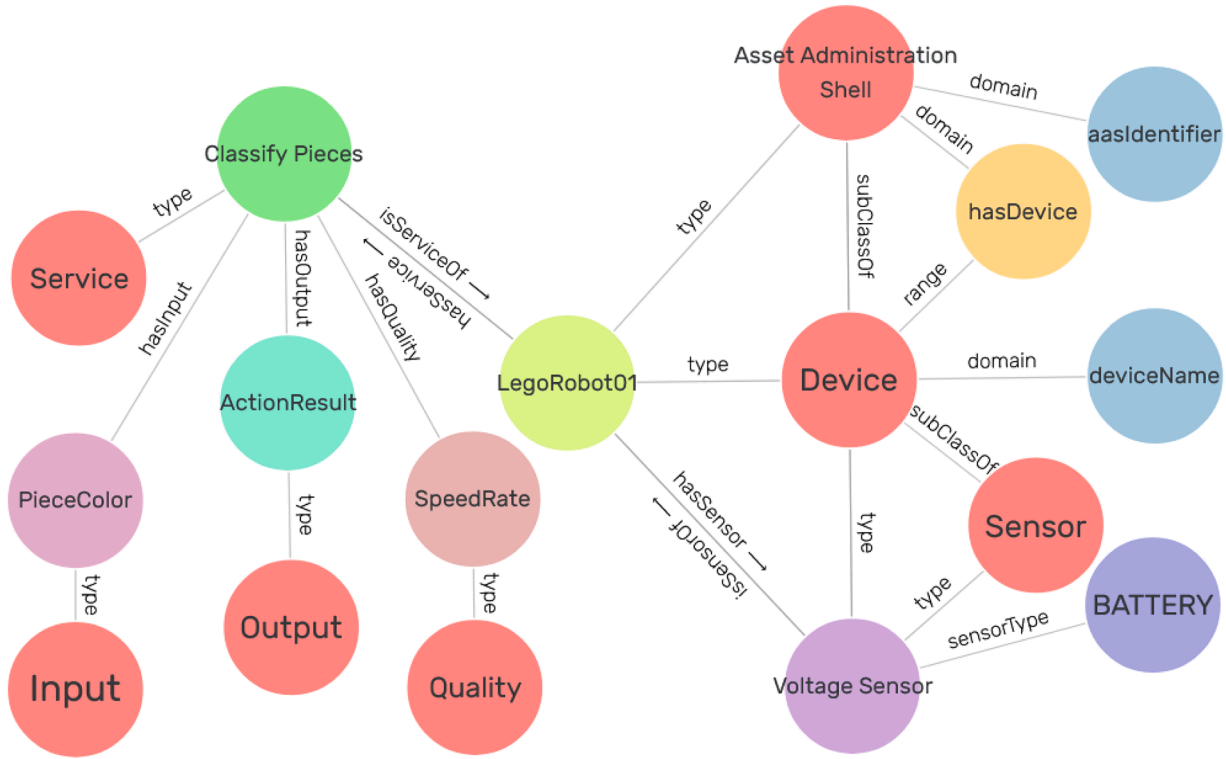


Fig. 8. RDF representation of the Lego robot in GraphDB.

Table 5
Case 1 SPARQL query results.

deviceName	serviceUrl	qosValAvgResponseTime	qosValHUMIDITY
Device_ColorSorter02	http://192.168.56.102:80/robot/move_left	6619	47

```

1 PREFIX ...
2 SELECT ?deviceName ?serviceUrl ?qosValAvgResponseTime ?qosValHUMIDITY
3 WHERE {
4   ?device dsOnt:aasIdentifier ?aasIdentifier .
5   ?device dsOnt:deviceName ?deviceName .
6   ?device dsOnt:hasService ?service .
7   ?service dsOnt:serviceName ?serviceName .
8   ?service dsOnt:serviceURL ?serviceUrl .
9   ?service dsOnt:hasQuality ?qAvgResponseTime .
10  ?qAvgResponseTime dsOnt:parameterName "AvgResponseTime" .
11  ?qAvgResponseTime dsOnt:parameterValue ?qosValAvgResponseTime .
12  ?service dsOnt:hasQuality ?qHUMIDITY .
13  ?qHUMIDITY dsOnt:parameterName "HUMIDITY" .
14  ?qHUMIDITY dsOnt:parameterValue ?qosValHUMIDITY .
15  filter (xsd:integer(?qosValAvgResponseTime) < 8000) .
16  filter (xsd:decimal(?qosValHUMIDITY) < 50) .
17  filter (?aasIdentifier = "AssetAdministrationShell---2") .
18  filter (?serviceName = "Service_MoveLeft") .
19 }
    
```

Listing 1. Case 1 SPARQL query

For the final scenario, Listing 4 represents the query when none of the devices/services align with the specified quality conditions, and the successive removal of quality conditions fails to yield any single instance. In such cases, all quality conditions are discarded, and an “ORDER BY” statement is added, resulting in an ordered list of instances (Table 8). Given that these instances do not satisfy any quality conditions, the workflow orchestrator must be aware that utilizing them for task execution does not guarantee its completion, rather it implies a high likelihood of non-completion.

Overall, the testing of these scenarios reaffirms the potential of the context-aware workflow management architecture, while showcasing its semantic web integration for achieving context-awareness.

5.4. Benefits and limitations of integrating I40GO

Incorporating the I40GO ontology within the architecture yields various advantages that significantly enhance the functionality and

```

1 PREFIX ...
2 SELECT ?deviceName ?serviceUrl ?qosValBATTERY ?qosValTEMPERATURE
3 WHERE {
4   ?device dsOnt:aasIdentifier ?aasIdentifier .
5   ?device dsOnt:deviceName ?deviceName .
6   ?device dsOnt:hasService ?service .
7   ?service dsOnt:serviceName ?serviceName .
8   ?service dsOnt:serviceURL ?serviceUrl .
9   ?service dsOnt:hasQuality ?qBATTERY .
10  ?qBATTERY dsOnt:parameterName "BATTERY" .
11  ?qBATTERY dsOnt:parameterValue ?qosValBATTERY .
12  ?service dsOnt:hasQuality ?qTEMPERATURE .
13  ?qTEMPERATURE dsOnt:parameterName "TEMPERATURE" .
14  ?qTEMPERATURE dsOnt:parameterValue ?qosValTEMPERATURE .
15  filter (xsd:decimal(?qosValBATTERY) >= 25) .
16  filter (xsd:decimal(?qosValTEMPERATURE) < 400) .
17  filter (?serviceName = "Service_ThrowPiece") .
18 } ORDER BY DESC(?qosValBATTERY && ?qosValTEMPERATURE)

```

Listing 2. Case 2 SPARQL query

```

1 PREFIX ...
2 SELECT ?deviceName ?serviceUrl ?qosValBATTERY ?qosValTEMPERATURE
3 WHERE {
4   ?device dsOnt:aasIdentifier ?aasIdentifier .
5   ?device dsOnt:deviceName ?deviceName .
6   ?device dsOnt:hasService ?service .
7   ?service dsOnt:serviceName ?serviceName .
8   ?service dsOnt:serviceURL ?serviceUrl .
9   ?service dsOnt:hasQuality ?qBATTERY .
10  ?qBATTERY dsOnt:parameterName "BATTERY" .
11  ?qBATTERY dsOnt:parameterValue ?qosValBATTERY .
12  ?service dsOnt:hasQuality ?qTEMPERATURE .
13  ?qTEMPERATURE dsOnt:parameterName "TEMPERATURE" .
14  ?qTEMPERATURE dsOnt:parameterValue ?qosValTEMPERATURE .
15  filter (xsd:decimal(?qosValBATTERY) >= 25) .
16  filter (?serviceName = "Service_ThrowPiece") .
17 } ORDER BY DESC(?qosValBATTERY && ?qosValTEMPERATURE)

```

Listing 3. Case 3 SPARQL query

```

1 PREFIX ...
2 SELECT ?deviceName ?serviceUrl ?qosValBATTERY ?qosValTEMPERATURE WHERE {
3   ?device dsOnt:aasIdentifier ?aasIdentifier .
4   ?device dsOnt:deviceName ?deviceName .
5   ?service dsOnt:hasService ?service .
6   ?service dsOnt:serviceURL ?serviceUrl .
7   ?service dsOnt:hasQuality ?qBATTERY .
8   ?qBATTERY dsOnt:parameterName "BATTERY" .
9   ?qBATTERY dsOnt:parameterValue ?qosValBATTERY .
10  ?service dsOnt:hasQuality ?qTEMPERATURE .
11  ?qTEMPERATURE dsOnt:parameterName "TEMPERATURE" .
12  ?qTEMPERATURE dsOnt:parameterValue ?qosValTEMPERATURE .
13  filter (?serviceName = "Service_ThrowPiece") .
14  ORDER BY DESC(?qosValBATTERY && ?qosValTEMPERATURE)
15 }

```

Listing 4. Case 4 SPARQL query

efficiency of Industry 4.0 systems. The key benefits brought about by the integration of I40GO are:

- Knowledge reusability: Ontology engineers can readily reuse I40GO components, accelerating ontology development for various manufacturing applications. Ontology engineers selected and reused only the modules necessary for the use case, thus reducing the ontology reuse process (Cuenca et al., 2020a). Then, ontology engineering activities (adding new classes, and ontology pruning) were conducted to align the selected modules with the use case functional requirements. Therefore, reusing I40GO did not require significant effort, as engineers were able to select and modify only the knowledge needed to develop DeviceServiceOnt.
- Enhanced interoperability: I40GO seamlessly connects diverse data domains, promoting efficient communication and interoperability among manufacturing components.
- Enhanced context-awareness: The ontology empowers the context-aware component within the architecture to dynamically adapt to changing manufacturing environments, maintaining the semantic repository without changes due to the reused knowledge of I40GO.

Table 6

Case 2 SPARQL query results.

deviceName	serviceUrl	qosValBATTERY	qosValTEMPERATURE
Device_ColorSorter01	http://192.168.56.101:80/robot/throw_piece	96	47
Device_ColorSorter03	http://192.168.56.103:80/robot/throw_piece	44	85

Table 7

Case 3 SPARQL query results.

deviceName	serviceUrl	qosValBATTERY	qosValTEMPERATURE
Device_ColorSorter01	http://192.168.56.101:80/robot/throw_piece	96	47
Device_ColorSorter03	http://192.168.56.103:80/robot/throw_piece	44	85

Table 8

Case 4 SPARQL query results.

deviceName	serviceUrl	qosValBATTERY	qosValTEMPERATURE
Device_ColorSorter01	http://192.168.56.101:80/robot/throw_piece	19	47
Device_ColorSorter03	http://192.168.56.103:80/robot/throw_piece	14	85

While the integration of I40GO offers numerous advantages, it may introduce a moderate learning curve for users unfamiliar with the ontology's structure and semantics. Additionally, ensuring consistent updates and maintenance of I40GO to align with evolving Industry 4.0 standards and requirements can pose a challenge. However, these limitations are outweighed by the substantial benefits of improved interoperability, context-awareness, and knowledge reusability.

6. Limitations disclosure

This section explains the main limitations of the conducted study, which are grouped into three key areas: the I40GO ontology, the methodology applied, and the use case scenario.

I40GO

The I40GO ontology is restricted to the knowledge encapsulated within Industry 4.0 ontologies available at the time of this study. Consequently, it does not incorporate concepts from ontologies or standards published during or after the research period. Moreover, I40GO does not provide mappings to link equivalent or overlapping concepts across the reused Industry 4.0 ontologies, which could hinder interoperability in more complex integration scenarios.

Methodology

The development of I40GO using the MODDALS methodology introduces the following limitations:

- The MODDALS guidelines are designed for the creation of the initial version of I40GO. As new ontologies and standards emerge in the Industry 4.0 domain, the current structure of I40GO may require modifications. However, MODDALS does not provide explicit guidelines for managing such changes or for developing future iterations of the ontology.
- The classification and organization of knowledge within I40GO are based on the available Industry 4.0 ontologies at the time of the study. The quality and validity of these reference ontologies were not independently assessed. The methodology assumes that the selected ontologies are developed collaboratively with domain experts, which may not always be the case.

Use case

The use case scenario presented in this study is limited to a controlled experimental setup. While it demonstrates the potential application of I40GO, it does not account for all real-world complexities, such

as heterogeneous data sources, incomplete or noisy data, and dynamic changes in device or service availability. Additionally, the SPARQL queries used in the scenarios assume ideal conditions where all necessary data is available and correctly linked, which may not always reflect practical industrial environments.

7. Conclusions and future work

This article presents I40GO, a global ontology for the Industry 4.0 domain. I40GO provides a unified representation of data domains within Industry 4.0, enabling reusability and supporting the development of domain-specific ontologies for various manufacturing applications. It reuses and organizes knowledge from existing Industry 4.0 ontologies into abstraction layers, effectively separating common knowledge-widely applicable across multiple applications-from variant knowledge, tailored to specific applications. This classification is based on the number of ontologies that represent each concept or set of concepts.

The ontology has been successfully reused within a context-aware architecture for workflow management. The ontology structural organization facilitated the selection of pertinent knowledge for the development of the ontology for the orchestration system. Furthermore, the modularity design of I40GO facilitated the understanding and adaptability of the reused ontology modules.

In terms of future work, several lines of work are defined to improve I40GO. We summarize them in the following subsections.

Extension of I40GO with new ontologies and standards

During and after the conducted research new ontologies have been developed in the Industry 4.0 domain. These ontologies will be taken into account to extend I40GO in future versions. Many of these ontologies are published in the Industry Portal.¹³ Some recent and relevant ontologies include the following: the IOF Core ontology represents common-domain knowledge to be used across different industrial applications (Drobnjakovic et al., 2022). This ontology is extended by other ontologies that include the knowledge of specific application domains within the industry. For instance, the IOF Maintenance Reference Ontology includes the concepts used by applications focused on maintenance and asset failure management (Hodkiewicz et al., 2024). The OMPD Static Procedure Ontology includes knowledge about the failures and maintenance tasks of manufacturing processes (Woods et al., 2023).

¹³ <https://industryportal.enit.fr>

In addition, the concepts of industry standards such as ISO 15,926 (Teijgeler, 2024) (which integrates life-cycle data of process plants) will be taken into account to extend I4OGO.

Definition of guidelines for the ontology extension and maintenance

The I4OGO ontology serves as a foundational tool for understanding and representing concepts within the Industry 4.0 domain. To ensure its continued relevance and effectiveness, it is essential to implement mechanisms for maintenance and evolution. As future work this article proposes the definition of a framework that incorporates best practices from existing methodologies, such as LOT (Poveda-Villalón et al., 2022), NeOn (Suárez-Figueroa et al., 2012) and METHONTOLOGY (Fernández-López et al., 1997), to address the challenges associated with updating and adapting I4OGO over time. The maintenance process should consider two phases.

Phase 1. Identification of new requirements, standards, or ontologies: The first part is the identification of new requirements, standards, or ontologies that include new knowledge or changes. A systematic review process should be conducted to periodically identify new ontologies and standards, such as the ones mentioned in the previous subsection. In addition, pilot projects that apply I4OGO in real-world scenarios will be conducted. Both new ontologies and the experience of pilot projects can provide valuable insights into how the ontology performs in practice to identify new knowledge that should be added to I4OGO.

Phase 2. Update of the ontology: The second part consists of updating the ontology to include new knowledge. The MODDALS methodology (applied to develop I4OGO), does not cover the ontology maintenance phase. Therefore, guidelines to maintain and update layered ontologies (including I4OGO) should be defined. New ontologies should be analysed to identify the knowledge they represent. After analyzing the knowledge of new ontologies, some knowledge that was variant may be considered common and vice versa. Thus, the update process should consider the reallocation of ontology modules to layers if common-variant thresholds change. In addition, changes in external ontologies linked with I4OGO should be analyzed and managed to maintain the links to these ontologies. After identifying these changes, the rest of the ontology development phases (ontology implementation, validation and publication) should be conducted.

Creation of mappings with other ontologies

I4OGO provides a common representation of industry 4.0 knowledge included by existing ontologies. To enable interoperability between applications reusing I4OGO knowledge and those using existing Industry 4.0 ontologies, mappings between I4OGO and these ontologies should be implemented. In addition, the ontology maintenance should take into account changes in external ontologies linked with I4OGO. The links between I4OGO and these ontologies should be reviewed periodically to check if the links are affected by changes in external ontologies.

CRedit authorship contribution statement

William Ochoa: Conceptualization, Software, Writing – Original Draft; **Javier Cuenca:** Methodology, Conceptualization, Software, Writing – Original Draft; **Felix Larrinaga:** Supervision, Validation, Conceptualization; **Alain Perez:** Software, Supervision, Validation.

Data availability

links to ontologies are provided. As well as links to source code of the demonstrative implementation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research work has been funded by the Department of Education, Universities and Research of the Basque Government under the SIIRSE project (KK-2022/00007) and the Ikerketa Taldeak program (IT-IT1519-22).

References

- Abdoulaliev, A. (2008). Reality, universal ontology and knowledge systems: Toward the intelligent world: Toward the intelligent world. IGI Global.
- Alexopoulos, K., Makris, S., Xanthakis, V., Sipsas, K., & Chryssolouris, G. (2016). A concept for context-aware computing in manufacturing: The white goods case. *International Journal of Computer Integrated Manufacturing*, 29(8), 839–849.
- Bader, S. R., Grangel-Gonzalez, I., Nanjappa, P., Vidal, M.-E., & Maleshkova, M. (2020). A knowledge graph for industry 4.0. In *European semantic web conference* (pp. 465–480). Springer.
- Basyx (2022). Basyx. <https://wiki.eclipse.org/BaSyx>.
- Cardoso, J., & Sheth, A. (2006). The semantic web and its applications. In *Semantic web services, processes and applications* (pp. 3–33). Springer.
- Chen, R.-S., & Tu, M. A. (2009). Development of an agent-based system for manufacturing control and coordination with ontology and RFID technology. *Expert Systems with Applications*, 36(4), 7581–7593. <https://doi.org/10.1016/j.eswa.2008.09.068>
- Choi, N., Song, I. Y., & Han, H. (2006). A survey on ontology mapping. *SIGMOD Record*, 35, 34–41. <https://doi.org/10.1145/1168092.1168097>
- Consortium, W. W. W. et al. (2021). OWL 2 web ontology language document overview. <https://www.w3.org/TR/owl2-overview/>.
- Cuenca, J., Larrinaga, F., & Curry, E. (2019). Experiences on applying SPL engineering techniques to design a (re) usable ontology in the energy domain. In *SEKE* (pp. 606–777).
- Cuenca, J., Larrinaga, F., & Curry, E. (2020a). DABGEO: A reusable and usable global energy ontology for the energy domain. *Journal of Web Semantics*, 61–62, 100550. <https://doi.org/10.1016/j.websem.2020.100550>
- Cuenca, J., Larrinaga, F., & Curry, E. (2020b). MODDALS Methodology for designing layered ontology structures. *Applied Ontology*, 15(2), 185–217. <https://doi.org/10.3233/AO-200225>
- Di Orio, G., Maló, P., & Barata, J. (2019). NOVAAS: A reference implementation of industrie 4.0 asset administration shell with best-of-breed practices from it engineering. In *IECON 2019-45th Annual conference of the IEEE industrial electronics society* (pp. 5505–5512). IEEE (vol. 1).
- Drath, R., Luder, A., Peschke, J., & Hundt, L. (2008). AutomationML - The glue for seamless automation engineering. In *2008 IEEE International conference on emerging technologies and factory automation* (pp. 616–623). https://doi.org/10.1007/978-3-319-22867-9_3
- Drobnjakovic, M., Kulvatunyou, B., Ameri, F., Will, C., Smith, B., & Jones, A. (2022). The industrial ontologies foundry (IOF) core ontology. In *FOMI 2022: 12th International workshop on formal ontologies meet industry, September 12–15, 2022, Tarbes, France* (pp. 1–13).
- Ehm, H., Ramzy, N., Moder, P., Summerer, C., Fetz, S., & Neau, C. (2019). Digital reference - A semantic web for semiconductor manufacturing and supply chains containing semiconductors. In *2019 Winter simulation conference (WSC)* (pp. 2409–2418). <https://doi.org/10.1109/WSC40007.2019.9004831>
- Elektroindustrie, Z. D. (2018). The reference architectural model RAMI 4.0 and the industrie 4.0 component. <https://www.zvei.org/en/press-media/publications/the-reference-architectural-model-industrie-40-rami-40>.
- Fernández-López, M., Gómez-Pérez, A., & Juristo Juzgado, N. (1997). Methontology: From ontological art towards ontological engineering. In *Proceedings of the ontological engineering AAAI-97 spring symposium series* (p.). American Association for Artificial Intelligence. Ontology Engineering Group - OEG <https://oa.upm.es/5484/>.
- Ghobaei-Arani, M., & Shahidinejad, A. (2022). A cost-efficient IoT service placement approach using whale optimization algorithm in fog computing environment. *Expert Systems with Applications*, 200, 117012. <https://doi.org/10.1016/j.eswa.2022.117012>
- Gómez-Pérez, A., Fernández-López, M., & Corcho, O. (2006). *Ontological engineering: With examples from the areas of knowledge management, e-commerce and the semantic web*. Springer Science & Business Media.
- Grangel-González, I., Halilaj, L., Coskun, G., Auer, S., Collarana, D., & Hoffmeister, M. (2016). Towards a semantic administrative shell for industry 4.0 components. In *2016 IEEE Tenth international conference on semantic computing (ICSC)* (pp. 230–237). <https://doi.org/10.1109/ICSC.2016.58>
- Gruber, T. (2009). Ontology. In *Encyclopedia of database systems*, pp. 1963–1965). Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-39940-9_1318
- Guarino, N. (1998). Semantic matching: Formal ontological distinctions for information organization, extraction, and integration 1299. https://doi.org/10.1007/3-540-63438-X_8

- Hailemariam, L., & Venkatasubramanian, V. (2010). Purdue ontology for pharmaceutical engineering: Part I. conceptual framework. *Journal of Pharmaceutical Innovation*, 5, 88–99. <https://doi.org/10.1007/s12247-010-9081-3>
- Hebeler, J., Fisher, M., Blace, R., Perez-Lopez, A., & Dean, M. (2009). Semantic web programming. Indianapolis IN: Wiley. <https://www.safaribooksonline.com/library/view/semantic-web-programming/9780470418017/>.
- Hodkiewicz, M., Woods, C., Selway, M., & Stumptner, M. (2024). Industrial ontology foundry (IOF) maintenance reference ontology. <https://doi.org/10.26182/chzp-vs60>
- Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). Recommender systems: An introduction. Cambridge University Press.
- Janowicz, K., Haller, A., Cox, S. J. D., Le Phuoc, D., & Lefrançois, M. (2019). SOSA: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics*, 56, 1–10. <https://doi.org/10.1016/j.websem.2018.06.003>
- Karray, M., Otte, N., Rai, R., Ameri, F., Kulvatunyou, B., Smith, B., Kiritzis, D., Will, C., & Arista, R., et al. (2021). The industrial ontologies foundry (IOF) perspectives. In *Proceedings: Industrial ontology foundry (IOF) achieving data interoperability workshop, international conference on interoperability for enterprise systems and applications, Tarbes, France, March 17–24, 2020*.
- Kovalenko, O., Wimmer, M., Sabou, M., Lüder, A., Ekaputra, F. J., & Biffl, S. (2015). Modeling automationML: Semantic web technologies vs. model-driven engineering. In *2015 IEEE 20th Conference on emerging technologies factory automation (ETFA)* (pp. 1–4). <https://doi.org/10.1109/ETFA.2015.7301643>
- Larrinaga, F., Ochoa, W., Perez, A., Cuenca, J., Legaristi, J., & Illarramendi, M. (2022). Node-RED workflow manager for edge service orchestration. In *NOMS 2022-2022 IEEE/IFIP network operations and management symposium* (pp. 1–6). IEEE.
- Lasi, H., Fetteke, P., Kemper, H. G., Feld, T., & Hoffmann, M. (2014a). Industry 4.0. *Business and Information Systems Engineering*, 6, 239–242. <https://doi.org/10.1007/s12599-014-0334-4>
- Lasi, H., Fetteke, P., Kemper, H.-G., Feld, T., & Hoffmann, M. (2014b). Industry 4.0. *Business & Information Systems Engineering*, 6(4), 239–242.
- Latif, S. A., Wen, F. B. X., Iwendi, C., Wang, L.-I. F., Mohsin, S. M., Han, Z., & Band, S. S. (2022). AI-empowered, blockchain and SDN integrated security architecture for IoT network of cyber physical systems. *Computer Communications*, 181, 274–283. <https://doi.org/10.1016/j.comcom.2021.09.029>
- Lee, J., Bagheri, B., & Kao, H.-A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3, 18–23. <https://doi.org/10.1016/j.mfglet.2014.12.001>
- Leitão, P., & Restivo, F. (2006). ADACOR: A holonic architecture for agile and adaptive manufacturing control. *Computers in Industry*, 57(2), 121–130. <https://doi.org/10.1016/j.compind.2005.05.005>
- Lemaignan, S., Siadat, A., Dantan, J.-Y., & Semenenko, A. (2006). MASON: A proposal for an ontology of manufacturing domain. In *IEEE workshop on distributed intelligent systems: Collective intelligence and its applications (DIS'06)* (pp. 195–200). <https://doi.org/10.1109/DIS.2006.48>
- Lohmann, S., Negru, S., Haag, F., & Ertl, T. (2016). Visualizing ontologies with VOWL. *Semantic Web*, 7(4), 399–419.
- Lu, Y., & Xu, X. (2017). A semantic web-based framework for service composition in a cloud manufacturing environment. *Journal of Manufacturing Systems*, 42, 69–81. <https://doi.org/10.1016/j.jmsy.2016.11.004>
- Magas, M., Kiritzis, D., Poveda-Villalón, M., Yang, L., Björling, S.-E., & Rudenā, A. (2024). Ecosystem integration: The use of ontologies in integrating knowledge across manufacturing value networks. *Frontiers in Manufacturing Technology*, 4. <https://doi.org/10.3389/fmtec.2024.1331197>
- Mesmer, L., & Olewnik, A. (2018). Enabling supplier discovery through a part-focused manufacturing process ontology. *International Journal of Computer Integrated Manufacturing*, 31(1), 87–100. <https://doi.org/10.1080/0951192X.2017.1357837>
- Morbach, J., Wiesner, A., & Marquardt, W., et al. (2009). OntoCAPE-A (re)usable ontology for computer-aided process engineering. *Computers & Chemical Engineering*, 33(10), 1546–1556. Selected Papers from the 18th European Symposium on Computer Aided Process Engineering (ESCAPE-18) <https://doi.org/10.1016/j.compchemeng.2009.01.019>
- Musen, M. A. (2015). The protégé project: A look back and a look forward. *AI Matters*, 1(4), 4–12.
- Muñoz, E., Espuña, A., & Puigjaner, L. (2010). Towards an ontological infrastructure for chemical batch process management. *Computers & Chemical Engineering*, 34(5), 668–682. Selected Paper of Symposium ESCAPE 19, June 14–17, 2009, Krakow, Poland. <https://doi.org/10.1016/j.compchemeng.2009.12.009>
- Natarajan, S., Ghosh, K., & Srinivasan, R. (2012). An ontology for distributed process supervision of large-scale chemical plants. *Computers & Chemical Engineering*, 46, 124–140. <https://doi.org/10.1016/j.compchemeng.2012.06.009>
- Navigli, R., & Velardi, P. (2004). Learning domain ontologies from document warehouses and dedicated web sites. *Computational Linguistics*, 30(2), 151–179. <https://doi.org/10.1162/089120104323093276>
- Ochoa, W., Larrinaga, F., & Pérez, A. (2023). Architecture for managing AAS-based business processes. *Procedia Computer Science*, 217, 217–226. 4th International Conference on Industry 4.0 and Smart Manufacturing. <https://doi.org/10.1016/j.procs.2022.12.217>
- Olivé, A. (2017). The universal ontology: A vision for conceptual modeling and the semantic web. In *Conceptual modeling: 36th International conference, ER 2017, Valencia, Spain, November 6–9, 2017, proceedings 36* (pp. 1–17). Springer.
- OMG (2022). Business process model and notation (BPMN). <https://www.omg.org/spec/BPMN/2.0/About-BPMN/>.
- Otte, J. N., Beverley, J., & Ruttenberg, A. (2022). BFO: Basic formal ontology. *Applied Ontology*, 17(1), 17–43.
- 4.0, P. I. (2021). Details of the asset administration shell Part 1. https://www.platform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.html.
- Poveda-Villalón, M., Gómez-Pérez, A., & Suárez-Figueroa, M. C. (2014). OOPS!(ontology pitfall scanner!): An on-line tool for ontology evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 10(2), 7–34.
- Poveda-Villalón, M., Fernández-Izquierdo, A., Fernández-López, M., & García-Castro, R. (2022). LOT: An industrial oriented ontology engineering framework. *Engineering Applications of Artificial Intelligence*, 111, 104755. <https://doi.org/10.1016/j.engappai.2022.104755>
- Pramanik, P. K. D., Mukherjee, B., Pal, S., Upadhyaya, B. K., & Dutta, S. (2020). Ubiquitous manufacturing in the age of industry 4.0: A state-of-the-art primer. Springer International Publishing, (pp. 73–112). https://doi.org/10.1007/978-3-030-14544-6_5
- Sesen, M. B., Suresh, P., Banares-Alcantara, R., & Venkatasubramanian, V. (2010). An ontological framework for automated regulatory compliance in pharmaceutical manufacturing. *Computers & Chemical Engineering*, 34(7), 1155–1169. Process Modeling and Control in Drug Development and Manufacturing <https://doi.org/10.1016/j.compchemeng.2009.09.004>
- She, Q., Wei, X., Nie, G., & Chen, D. (2019). QoS-aware cloud service composition: A systematic mapping study from the perspective of computational intelligence. *Expert Systems with Applications*, 138, 112804. <https://doi.org/10.1016/j.eswa.2019.07.021>
- Sipsas, K., Zalonis, A., Broechler, R., Baumgaertel, H., & Ehm, H. (2020). A platform application for exploiting the digital reference ontology. In *2020 IEEE Conference on industrial cyberphysical systems (ICPS)* (pp. 557–562). (vol. 1). <https://doi.org/10.1109/ICPS48405.2020.9274692>
- Song, S., Zhang, X., & Qin, G. (2017). Multi-domain ontology mapping based on semantics. *Cluster Computing*, 20, 3379–3391.
- Suárez-Figueroa, M. C., Gómez-Pérez, A., & Fernández-López, M. (2012). The neon methodology for ontology engineering. In *Ontology Engineering in a Networked World*, pp. 9–34. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-24794-1_2
- Tejgeler, H. (2024). ISO 15926 - The lingua Franca of global interoperability. <https://15926.blog/ISO15926-consolidating-paper-20240714.pdf>.
- Uschold, M., & Gruninger, M. (2004). Ontologies and semantics for seamless connectivity. *ACM SIGMOD Record*, 33(4), 58–64.
- Wang, P., Chao, K.-M., & Lo, C.-C. (2010). On optimal decision for QoS-aware composite service selection. *Expert Systems with Applications*, 37(1), 440–449. <https://doi.org/10.1016/j.eswa.2009.05.070>
- Wawrzik, F., Rafique, K. A., Rahman, F., & Grimm, C. (2023). Ontology learning applications of knowledge base construction for microelectronic systems information. *Information*, 14(3). <https://doi.org/10.3390/info14030176>
- Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E. et al. (2016). The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, 3(1), 1–9.
- Woods, C., French, T., Hodkiewicz, M., & Bikaun, T. (2023). An ontology for maintenance procedure documentation. *Applied Ontology*, 18(2), 169–206.
- Yan, J., Meng, Y., Lu, L., & Li, L. (2017). Industrial big data in an industry 4.0 environment: Challenges, schemes, and applications for predictive maintenance. *IEEE Access*, 5, 23484–23491.
- Zammali, S., & Ben Yahia, S. (2021). How to select and weight context dimensions conditions for context-aware recommendation? *Expert Systems with Applications*, 182, 115176. <https://doi.org/10.1016/j.eswa.2021.115176>
- Zhang, Y., Zhang, G., Liu, Y., & Hu, D. (2017). Research on services encapsulation and virtualization access model of machine for cloud manufacturing. *Journal of Intelligent Manufacturing*, 28(5), 1109–1123.