Towards semantic interoperability standards based on ontologies

Hamza Baqa, Martin Bauer, Sonia Bilbao, Aitor Corchero, Laura Daniele, Iker Esnaola, Izaskun Fernández, Östen Frånberg, Raúl García Castro, Marc Girod-Genet, et al.

Towards Semantic Interoperability Standards based on Ontologies[1]

Semantic Interoperability White Paper

Editor

- Martin Bauer, NEC Laboratories Europe

Contributors

- Hamza Baqa, Easy Global Market
- Martin Bauer, NEC Laboratories Europe
- Sonia Bilbao, Tecnalia Corporación Tecnológica
- Aitor Corchero, Eurecat - Technology Centre of Catalonia
- Laura Daniele, TNO
- Iker Esnaola, IK4-TEKNIKER
- Izaskun Fernández, IK4-TEKNIKER
- Östen Frånberg,1A Konsult
- Raúl García-Castro, Universidad Politécnica de Madrid
- Marc Girod-Genet, Institut Mines-Télécom
- Patrick Guillemin, ETSI
- Amélie Gyrard, Kno.e.sis, Wright State University
- Charbel El Kaed, Google
- Antonio Kung, TRIALOG
- Jaeho Lee, University of Seoul
- Maxime Lefrançois, École des Mines de Saint-Étienne
- Wenbin Li, Orange
- Dave Raggett, W3C
- Michelle Wetterwald, NETELLANY / FBConsulting

Table of Contents

Background

This paper is co-authored by an informal group of experts from a broad range of backgrounds, all of whom are active in standards groups, consortia, alliances and/or research projects in the Internet of Things (IoT) space.

This paper has two objectives: 1) explain the need for semantic interoperability, 2) provide recommendations for semantic interoperability standards using ontologies.

The target audience for this paper are:

- IoT system product owners who need to understand how they can effectively ensure interoperability of their products.
- IoT system and standardization engineers without background in semantic technologies.

1 Introduction

The paper is structured as follows: Section 2 introduces semantic interoperability and its benefits; Section 3 provides industry requirements for semantic interoperability practice; Section 4 describes various initiatives for ontology-driven interoperability; Section 5 explains the various life cycles for ontology-driven interoperability; and finally, Section 6 provides recommendations on ontology-based semantic interoperability.

2 Semantic interoperability

Interoperability specification describes how two systems or components can engage into a working interaction e.g. two IoT devices. Semantic interoperability focuses on describing the semantics of such interaction.

A semantic interoperability process might focus on various description viewpoints (as shown in Figure 1: 1) information exchanged, 2) interactions, and 3) others

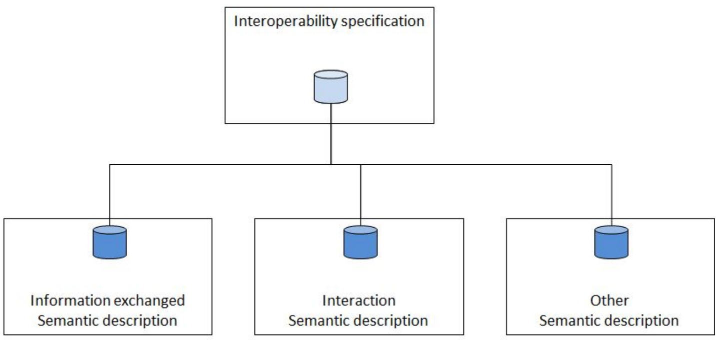


Figure 1. Semantic interoperability

For instance, the interoperability specification of the protocol between two IoT devices connected through a network may include:

- A semantic description to describe the device capabilities such as measuring temperature (other semantic description).
- A semantic description to describe the protocols such as wifi (or interactions)
- A semantic description to describe protocol data units such as celsius data unit (or information exchanged)..

2.1 Ontology-driven interoperability

Ontology-driven interoperability aims to produce the semantic descriptions in Figure 1.

An ontology describes concepts and relationships between concepts in a specific domain. For instance, in the case of a description of information exchanged, ontologies describe the concepts contained in the information exchanged as well as the relationship links between those concepts.

An ontology can be created using computer description languages such as RDF (Resource Description Framework), RDFS (Resource Description Framework) Schema) or OWL (Ontology Web Language). Languages can be serialized in several formats such as XML (eXtensible Markup Language). The semantic web stack classifies languages such as RDF, RDFS, and OWL (as shown in Figure 2).

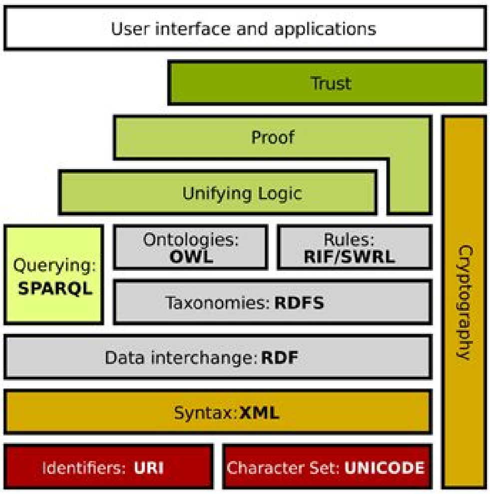


Figure 2. Semantic Web Cake[2] [1]

[2] Figure under CC0 license: https://commons.wikimedia.org/wiki/File:Semantic_web_stack.svg

2.2 Benefits of semantic interoperability

Applying semantic interoperability in the industry has several benefits:

- The quality of an interoperability specification is improved as a systematic process is applied for defining interoperability.
- The resulting specifications can be used as a reference when interpretation problems have to be solved. Instead of a textual specification, a formalised specification (e.g. ontologies) is available. It can be used by further tools for verification and validation.
- Maintenance and extension of the specification is more straightforward. While it is difficult to assess the impact of a modification in a textual specification of interoperability, it is easier to do so in a formalised specification.

3 Industry requirements for semantic interoperability practice

Producing a semantic interoperability specification that can be widely used in a market requires a specification practice that takes into account the following requirements: 1) co-creation and separation of concerns; 2) definition of the knowledge perimeter needed for a specification; 3) modular design principle following design pattern approaches; 4) evaluation of a specification, and 5) support of industry deployment concerns.

3.1 Co-creation and separation of concerns

Co-creation is a design approach that brings experts with different expertise and viewpoints together, for instance, a domain expert and a technology expert), in order to jointly produce a mutually valued outcome.

Separation of Concerns (SoC) is a design principle for separating an item to design into distinct elements, so that each element addresses a separate concern (Table 1 provides an example).

The practice of semantic interoperability, i.e. the creation of an interoperability specification requires two kinds of expertise: 1) domain experts bring knowledge on domain engineering, and 2) semantic interoperability experts bring knowledge on ontology engineering. Depending on the domain, other categories of experts are relevant such as security and privacy experts, or user-centric design experts, e.g. the eHealth vertical where systems have to be designed both taking into account security/privacy/trust (by design) and in co-conception with the patients, caregivers and the helpers (relatives). It is important to achieve a clear separation of concerns between domain experts and semantic interoperability experts. Without this separation of concerns, one can easily fall into a trap where a domain expert has to rely on a semantic interoperability expert to propose a specification, and where interoperability decisions are taken by the wrong expert (e.g. the domain expert changes the ontology). From a method and tools viewpoint, recommendations must be provided to enable separation of concerns. For instance, domain experts inspect and update a specification using a domain viewpoint, while the semantic interoperability expert's focus is on inspecting and updating a specification using ontology engineering.

Table 1. Separation of concerns SAREF example

Example	Description
Example of practice integrating separation of concern	An example of good separation of concerns is to organize co-creation sessions when both categories are present to make design decisions. This was achieved by the SAREF team when they organized a session for the European Large Scale Pilots during the IoT week in Bilbao in June 2018 to get input from domain experts that they could use to specify an ontology to model different domains (e.g., smart home, agriculture, energy) as depicted in Figure 7.

3.2 Defining the knowledge perimeter needed for a specification

It is important to clearly define the knowledge that is needed for a semantic interoperability specification. We call this the **knowledge perimeter**.

If the selected knowledge perimeter is too broad, then many concepts that are defined in the ontology might not be used. Worse, it could be counter effective. Moreover, when cross domain ontologies are used, it is important to select the subset of concepts and properties rather than the entire domain ontology.

If the selected knowledge perimeter is too small then needed concepts in the specification would be missing, which could result in an incomplete semantic specification.

Table 2. Example of practice for specification scope

Example	Description
Example of practice for specification scope	An interoperability specification is defined to enable cross domain interoperability. For instance, interoperability is needed between an energy management system and an electric vehicle charging system. The resulting ontology covers a common subset of the energy, mobility domain, and the vehicle charging system.

3.3 Modularization design principle

The modularization principle concerns the structuring of a wide concept into multiple and simpler sub-concepts that can be detailed independently[3]. These sub-concepts can therefore be described by self-contained knowledge sub-ontologies (modules) that are:

- Loosely coupled among themselves and can be designed, used and maintained in a stand-alone way, as well as processed with far less processing power requirements than complex ones. This is in particular mandatory for handling both: use cases involving embedded devices with low power/energy and resources constraints, edge computing and device-embedded analytics.
- Linked to other sub-ontologies with defined relationships. This preserves the full semantic richness of the model or ontology.
- Reusable.

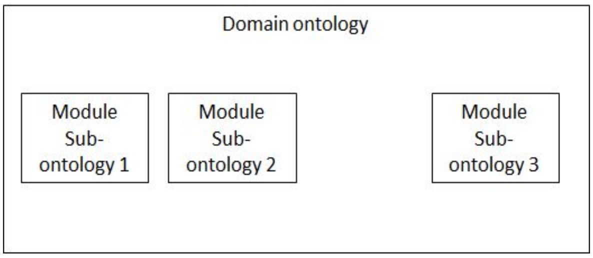


Figure 3. Modular specification

Guarino [2] proposes the following structure:

- top-level ontologies covering general concepts (e.g. space, time, matter, object, event, action) which are independent of a particular problem or domain;
- domain ontologies and task ontologies, covering concepts related to a generic domain (e.g. energy) or a generic task or activity (e.g. flexibility management); and
- application ontologies, covering a particular specialization of the above ontologies, often corresponding to the description of a specific capability (such as energy consumption measurement).

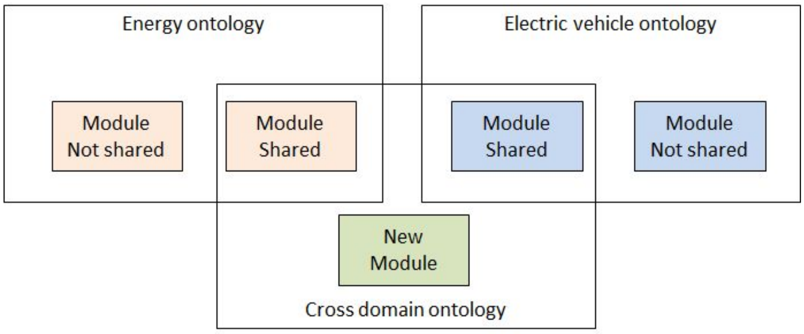
[3] This can be achieved by design pattern approaches

However, when using the modularization principle, one shall ensure that:

- integrity of a sub-ontology is maintained, i.e. if a sub-ontology depends on other ones, any sub-ontologies changes should preserve those dependency relations,
- The processing of the sub-ontologies union is not too complex.
- The reasoning and querying are still decidable for the modularized ontology, i.e. can still be performed within a finite time period.
- integrity of a sub-ontology is maintained, which means that if a sub-ontology depends on other ones, any changes should preserve those dependency links.

Modularization is easier to achieve if an organisation can use specification tools, like e.g. ModOnto [3] inspired for object oriented software engineering, to edit and structure of a specification into modules.

Table 3. Example of practice for ontology modularization

Example	Description
Example of practice for ontology modularization	In the previous cross-domain interoperability specification it is not useful to publish the entire energy domain ontology nor the entire electric vehicle ontology. A modular specification allows for the sharing of sub-ontologies at a sufficient level.  Figure 4. Example of modular specification

3.4 Evaluation of a specification

It is important to evaluate the “usefulness” of a specification. Specifications are defined for designing applications. One typical indicator is the level of consensus. A specification that has not reached consensus is likely not to be adopted. Semantic interoperability specifications that are not cocreated by domain and ontology experts can fall into this trap. Domain experts are required to constantly follow the specification process and agree on the content while semantic interoperability experts guarantee that the specification is sound.

Specifications need evaluations. It could rely on an indicator consisting of two TRLs (Technology Readiness Level) or a metrics used in the industry to measure whether a product is close to the market. A specification is deemed mature when both TRLs are high, TRL examples are: 1) a **domain specification TRL** which focuses on whether all domain needs are covered, and 2) an **ontology specification TRL**, which focuses on whether the specification is well-formed. Raad [4] provides a survey on ontology evaluation.

For instance, a tool assisting interoperability engineers to structure a specification into modules and to assess the TRL could be useful.

Table 4. Example of practice specification evaluation

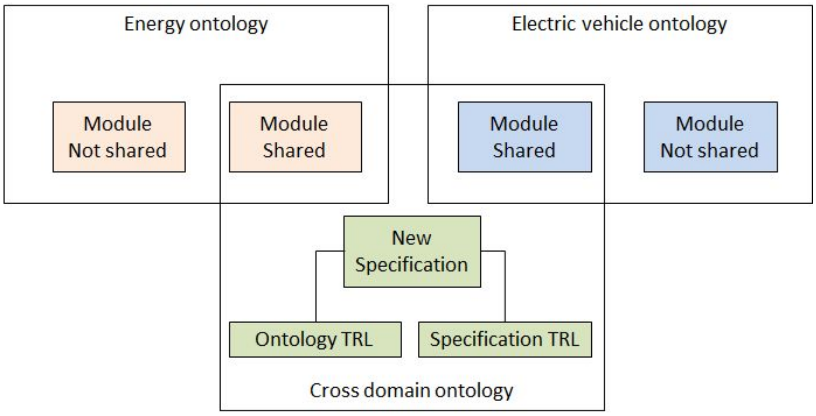
<table>
<tr><th>Example</th><th>Description</th></tr>
<tr><td>Example of practice for specification evaluation</td><td>In the previous example, the new cross domain specification starts with a low TRL for the ontology and for the specification. The TRL increases as the associated ontology is validated (ontology TRL) and the consensus is reached (specification TRL).


Figure 5. Example of specification evaluation</td></tr>
</table>

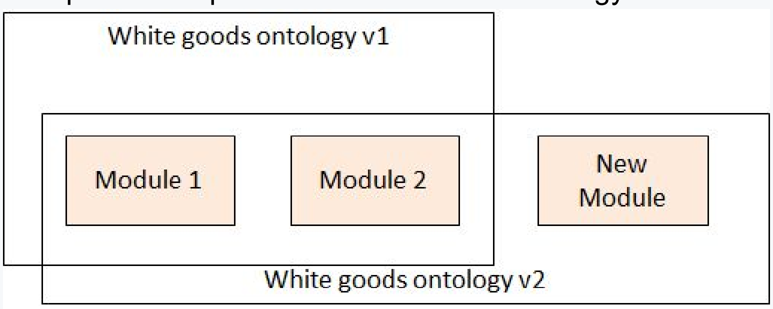
3.5 Deployment concerns

Deployment concerns in a specification of a semantic interoperability standard is important. Two main concerns are:

- **Provision for profiles and discovery**. Some specifications concern a domain market segment. For instance, device manufacturers want to add semantic specifications concerning features (e.g., providing web services to send data on the Web). Specifications might even be proprietary when device manufacturers agree on co-existing solutions solved by service discovery capabilities. Profiles are widely used in interoperability specifications (e.g., a washing machine) implements extra features for interoperability such as finer grain remotely control of the washing machine. Consequently semantic interoperability specifications should also support profiles; a profile can be a concept in the ontology.

- **Support for version management**. Semantic interoperability specifications evolve as a domain evolves to match the needs of different generations of products (e.g., a new generation of smartphone). Two types of version management are needed: 1) a specification change: the rules for compatibility must be anticipated, e.g. do two systems using different versions interoperate?, and 2) an ontology evolution [4]: is the specification changed? In the two cases, mechanisms to support such evolutions should be agreed upfront.

Table 5. Example of deployment requirements

Examples of deployment requirements	Description
Concerning profiles	**Managing ontologies from a profile viewpoint:** The profile concept is handled at the ontology level (either as part of the ontology, or as part of tools supporting the ontology). Browsing an ontology from a profile viewpoint is possible, i.e. only showing the concepts that are used by the profile. **Managing Intellectual Property Rights (IPR) while ensuring open interoperability specifications:** a semantic interoperability specification refers to ontology subsets which contain IPR, for instance, the use of an ontology describing a functional behavior that is patented.
Concerning version management	**Upward compatibility:** Here is an example scenario: a washing machine uses the SAREF V1 ontology. In a second generation of washing machine, an extended specification allows control of the washing machine by an Artificial Intelligence (AI) agent. The SAREF V1 ontology evolves to a SAREF V2 ontology. All new generation washing machines are upward compatible with SAREF V1 ontology.  Figure 6. Ontology evolution management

4 Initiatives for structured ontologies supported by standardization

4.1 Initiatives on ontologies supported by standardization

A number of ongoing standardization initiatives on semantic interoperability are described in Table 7 (initially referenced in [5] [6]).

Table 6. Standardization initiatives on semantic interoperability

Initiative	Description
W3C Semantic Sensor Network ontology	The Semantic Sensor Network (SSN) [7] ontology is an ontology for describing sensors and their observations, the involved procedures, the studied features of interest, the samples used to do so, and the observed properties, as well as actuators. SSN follows a horizontal and vertical modularization architecture by including a lightweight but self-contained core ontology called SOSA (Sensor, Observation, Sample, and Actuator) [8] for its elementary classes and properties [9]. With their different scope and different degrees of axiomatization, SSN and SOSA are able to support a wide range of applications and use cases, including satellite imagery, large-scale scientific monitoring, industrial and household infrastructures, social sensing, citizen science, observation-driven ontology engineering, and the Web of Things.
W3C Web of Things	The Web of Things (WoT) is an extension of the Internet of Things (IoT) to ease the access to data using the benefits of Web technologies [10,11]. Data is generated by things/devices and then exploited by more and more web-based applications to monitor healthcare or even control home automation devices. The W3C Web of Things (WoT) Interest Group is designing a vocabulary to describe interactions between objects through the Web, a potential implementation is the WoT ontology [12]. At the date of writing, the WoT ontology is not aligned with W3C SSN ontologies, but there is ongoing work on aligning them. A healthcare scenario has been designed "Remote health monitoring system" among several use cases.
oneM2M	oneM2M is an international standard for Machine-to-Machine (M2M) that has developed the oneM2M Base Ontology [13]. At the date of writing, the oneM2M Base Ontology is not aligned with W3C SSN, but it is aligned with SAREF core concepts.
SmartBAN (MyOntoSens)	MyOntoSens modular ontology, mainly based on SSN V1 and OGC standards, is an improvement of existing WSNs ontologies [14]. It has been standardized in 2015 for medical devices and

	BANs (Body Area Networks) as a Technical Specification (TS) within the SmartBAN Technical Committee of the ETSI standardization body [15]. This ontology is relevant to build health, wellbeing/wellness and personal safety applications based on smart devices.
SAREF	The Smart Applications Reference Ontology (SAREF) [16] is a standardized ontology for IoT devices and solutions published by ETSI in a series of Technical Specifications initially released in 2015 [17] and updated in 2017 [18]. Even if its initial objective was to build a reference ontology for appliances relevant for energy efficiency, SAREF is not limited to this scope and can serve as upper reference model to enable better integration of data from various vertical domains in the IoT. Hence, SAREF has been extended to different domains such as energy, environment, buildings, smart cities, agriculture, industry & manufacturing; and is currently being extended to the automotive, eHealth/ageing-well, wearables and water domains. SAREF has been designed re-using SSN and oneM2M according to [19]. ETSI has consolidated SAREF with new reference ontology patterns and is developing a new SAREF development workflow [20].
Schema.org	Schema.org. is a well-known schema catalog to structure data on Web pages to describe the location, person, etc. The IoT Schema.org extension [21] is planned; discussions are ongoing.

4.2 System viewpoint of ontologies

While it is important to foster ontology developments, there is a need for convergence in order to avoid the following risks:

- The use of incompatible ontologies might actually prevent interoperability, thus creating a market fragmentation effect.
- There might be too many competing ontologies for the same domain creating a babel tower situation.

In order to prevent these issues, a system viewpoint should be taken, as exemplified by SAREF [17]. Figure 7 shows an architecture on how ontologies are structured: a base ontology (e.g., based on oneM2M) is above which a SAREF framework is positioned to host domain-specific ontologies.

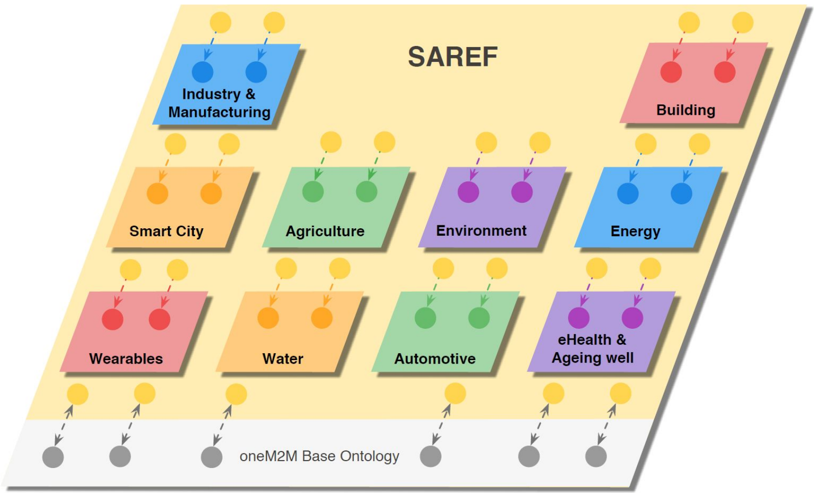


Figure 7. Example of system vision (from ETSI TR 103 411 [22])

5 Life cycles for ontology-driven interoperability

Supporting interoperability requires a system lifecycle viewpoint to ensure that proper requirements, design, implementation, validation and maintenance of interoperability features are integrated.

5.1 Interoperability-by-design

5.1.1 Introduction to system life cycles

ISO/IEC/IEEE 15288 (Systems and software engineering — System life cycle processes) [23] defines a system lifecycle as “an abstract functional model representing the conceptualization of a need for the system, its realization, utilization, evolution and disposal”. A system lifecycle is described as a set of processes, which can take place sequentially or in parallel, as shown in Figure 8 [24].[25]

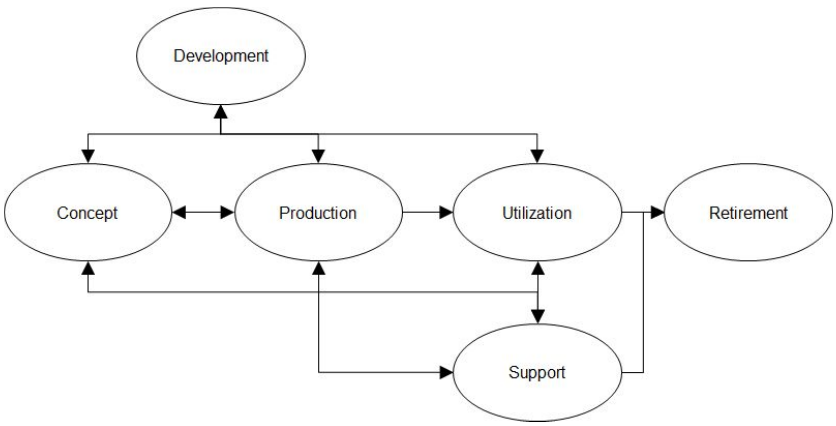


Figure 8. Example of System Life Cycle Processes

As shown in Figure 9, a process is described according to: its purpose; the outcome it creates, and its activities which themselves consist of tasks.

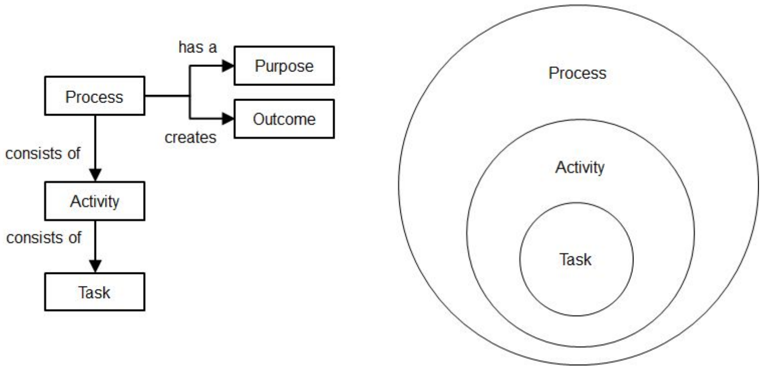


Figure 9. Processes, Activities and Tasks

The ISO/IEC/IEEE 15288 standard [23] describes thirty processes structured into four categories:

- Agreement processes which focus on activities related to supplier agreements,
- Organizational project-enabling processes which focus on activities related to improvement of the organization's business or undertaking,
- Technical management processes which focus on managing the resources and assets allocated to the engineering of a system, and
- Technical processes which focus on technical actions throughout the life cycle.

The sections below provide guidance on which system life cycle processes need to integrate interoperability activities.

5.1.2 Definition of interoperability-by-design

We define interoperability-by-design as the integration of the concept of interoperability in the design and lifecycle of systems, as shown in Figure 10.

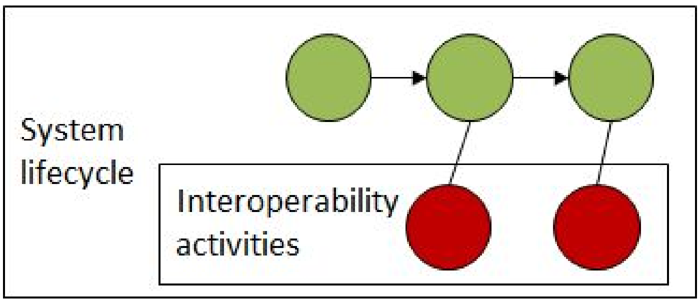


Figure 10. Interoperability-by-design

Relationship between interoperability-by-design process (i.e. integrating interoperability concerns in the development of a system) and an interoperability specification lifecycle is shown in Figure 11.

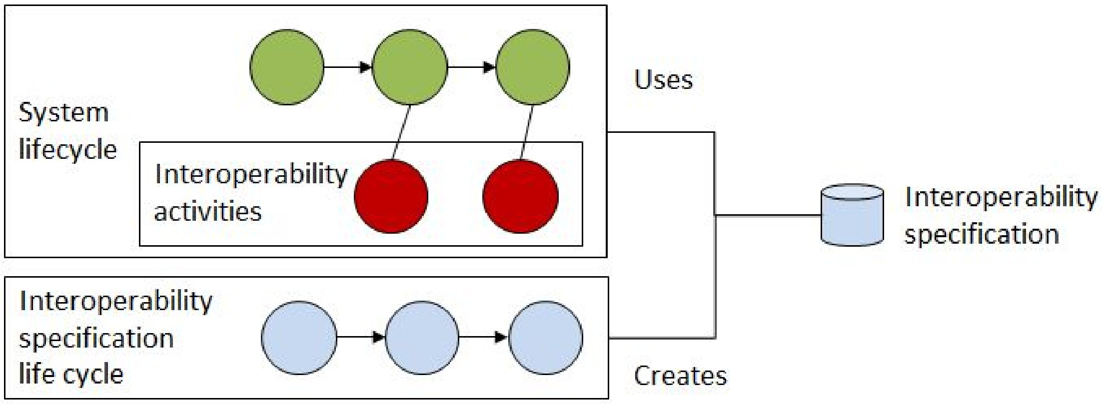


Figure 11. Interoperability-by-design vs Interoperability specification lifecycle

5.1.3 Interoperability activities system lifecycle

Activities/tasks related to interoperability by design that need to be integrated are shown in the table below which uses the ISO/IEC/IEEE 15288 processes and provides examples of activities that are related to interoperability.

Table 7. Lifecycle process and related interoperability activities

Typical lifecycle technical process (e.g. ISO/IEC/IEEE 15288)	**Interoperability activities**
Stakeholder needs and requirements definition	Interoperability needs and ontology requirements definition
System requirements definition process	Interoperability requirements
Architecture definition process	Interoperability point definition
Design definition process	No specific activity
System analysis process	Interoperability point specification
Implementation process	Interoperability point implementation
Integration process	No specific activity
Verification process	Interoperability test
Transition process	Interoperability plug test
Validation process	Validation test
Operation process	No specific activity
Maintenance process	Interoperability maintenance
Disposal process	No specific activity

5.1.4 Interoperability specification lifecycle

An interoperability specification follows its own lifecycle (a simple example is depicted in Figure 12 and explained in Table 8. Such lifecycles are well known.

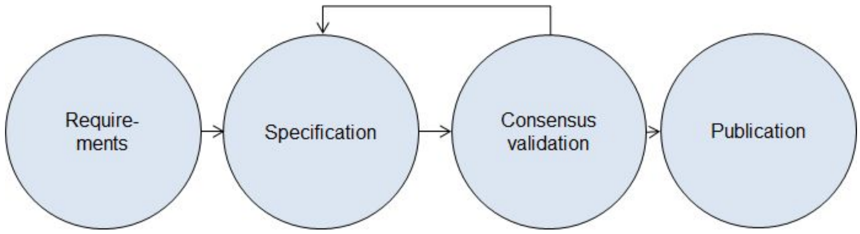


Figure 12. Example of interoperability specification lifecycle

Table 8. Interoperability specification lifecycle stages

Interoperability specification lifecycle stages	Description
Requirement	Define the requirements of the interoperability specification
Specification	Provide the specification
Consensus validation	Consensus reaching on the specification
Publication	Publish the interoperability specification

5.2 Ontology-driven semantic Interoperability

5.2.1 Life cycles involved

Ontology-driven semantic interoperability assumes that interoperability-by-design is based on the use of ontologies to describe the meaning of exchanged information. Figure 13 shows the relationship between the interoperability lifecycle and the ontology lifecycle. The following remarks can be made:

- The system lifecycles and the interoperability specification lifecycles are separated.
- The interoperability specification lifecycles and the ontology lifecycles are separated.

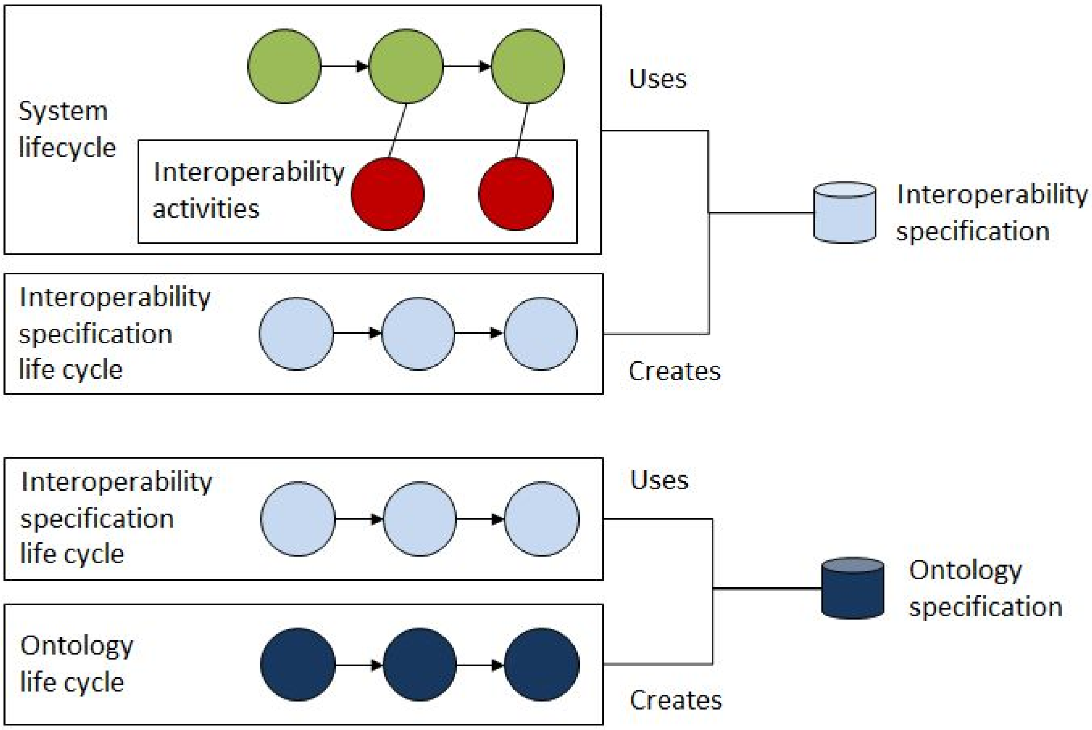


Figure 13. Ontology-driven Interoperability

5.2.2 Example for benefits of ontology-driven semantic interoperability

The benefit of ontology-driven ontology can be applied within Internet of Things applications, for instance:

- Domain specific capabilities are described (e.g., sensing information from a connected vehicle, or health sensing information from connected body sensors) annotated with domain specific ontologies;
- The annotated sensing information is extended with higher level concepts to provide an IoT application and platform viewpoint, using a service ontology model as suggested by the W3C [26] as shown in Figure 14. The result is that a sensor is viewed as a service (here a sensing service), which is described with unified, common and shared concepts:
    - A **service profile** which expresses the service capabilities,
    - A **service process** which specifies how the service works (including the service control and function calls),
    - The **service grounding**, which specifies how to access the service,

This approach is beneficial for cross-domain interoperability:

- a generic query service is available allowing the inspection of the device services (connected vehicle sensor, health body sensor or an environmental sensor)
- a unified discovery service can be used, and
- an overall application / platform level interoperability framework is available..

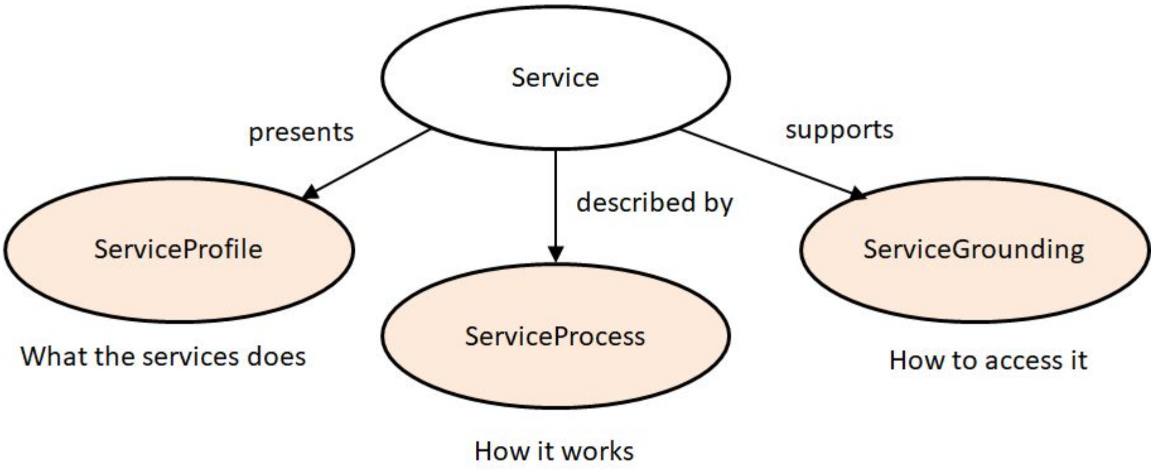


Figure 14: High level example of a service ontology model (OWL-S) [26]

5.2.3 Ontology engineering

Ontology development typically follows a lifecycle, as shown in Figure 15 and explained in Table 9.

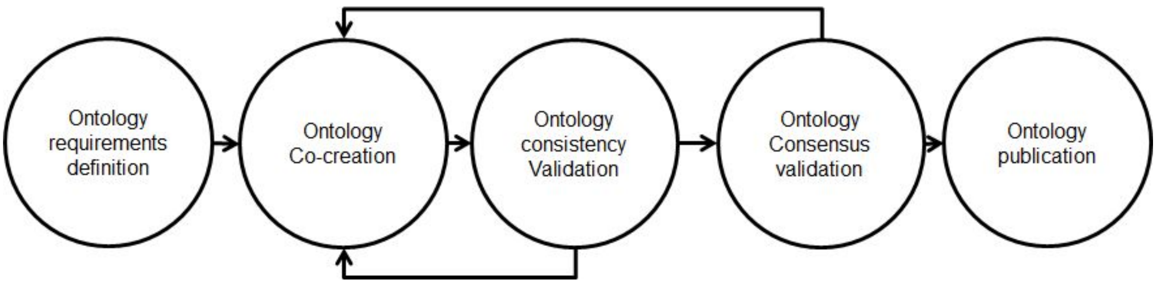


Figure 15. Ontology lifecycle model example

Table 9. Ontology lifecycle process

Ontology lifecycle process	**Description**
Ontology requirements definition	Define the requirements of the ontology to create
Ontology co-creation	Co-create the ontology. This process must at least include a domain specific expert and on ontology expert
Ontology consistency validation	Validation that an ontology is well-formed

Ontology consensus validation	Consensus reaching on the created ontology
Ontology publication	Publish the ontology

A number of ontology lifecycle models have been proposed such as the OTK methodology [27], the Neon project collection of lifecycles [28] or the 101 methodology [29].

Table 10 below shows the stages of OTK.

Table 10. Ontology lifecycle stages

Ontology lifecycle stages	**Description**
Feasibility study	Identify stakeholders and use cases, identify tools.
Ontology kickoff	Capture requirements Analyse knowledge sources Develop baseline ontology
Refinement	Extract knowledge Formalise
Evaluation	Technology focused evaluation User focused evaluation Ontology focused evaluation
Application and evolution	Apply ontology Manage evolution and maintenance

The Neon project lists the following models:

- Waterfall models such as
  - the four-phase model (initiation, design, implementation, maintenance),
  - the five-phase model (initiation, reuse, design, implementation, maintenance),
  - the five-phase+merging phase model (initiation, reuse, merging, design, implementation, maintenance),
  - the six-phase model (initiation, reuse, re-engineering, design, implementation, maintenance), and
  - the six-phase+merging phase model (initiation, reuse, merging, re-engineering, design, implementation, maintenance),
- Iterative-incremental ontology network lifecycle models, where there are iterations and where each iteration follows a waterfall model.

The NeOn Methodology is a scenario-based methodology supporting different aspects of the ontology development process: from the reuse of existing resources, to the dynamic evolution of ontologies in distributed environments where knowledge is introduced by different people at different stages. Furthermore, the proposed scenarios are decomposed into different activities which can be combined for the achievement of the expected goal.

There are nine scenarios defined in the NeOn Methodology:

- Scenario 1: From specification to implementation
- Scenario 2: Reusing and re-engineering non-ontological resources (NORs)
- Scenario 3: Reusing ontological resources
- Scenario 4: Reusing and re-engineering ontological resources
- Scenario 5: Reusing and merging ontological resources
- Scenario 6: Reusing, merging and re-engineering ontological resources
- Scenario 7: Reusing ontology design patterns (ODPs)
- Scenario 8: Restructuring ontological resources
- Scenario 9: Localizing ontological resources

5.2.4 Ontology validation methodsSemantic-based

Several methods are available for the validation of an ontology: 1) Syntactic-based validation, 2) Semantic-based validation, and 3) Evolution-based validation.

Syntactic-based validation mainly consists in detecting potential pitfalls that could lead to modelling errors. It includes the use of undefined properties and classes, poorly formed namespaces, problematic prefixes, literal syntax.

Semantic-based validation uses rules which are built in the ontology languages and rules users provided to detect logical issues in ontologies (ex: contradictory inferred result). Examples of the first type are when two objects in an OWL ontology are said to be different from each other (owl:differentFrom), the ontology can't say that they are the same thing (owl:sameAs).

Finally, **evolution-based validation** consists in observing the evolution of the ontology usage, over its usage lifecycle. The original ontology schema is a posteriori compared to all the instances of that ontology that have been used and or introduced (i.e. amended) during a given period of time. The retained evaluation criteria can be:

- Ontology domain changes, i.e. any new knowledge that could have been added to the domain formalized by the original ontology,
- Ontology usage perspectives changes, in a given domain, impacting the ontology conceptualization,

- Ontology specification changes (ontology stability metric), i.e. number of new concepts or attributes introduced in the original ontology.

Ontology usage, after its publication, is also monitored, and access to ontology classes (i.e. concepts) and attributes can be counted. This provides metrics for pointing out the concepts and attributes most often used, as well as the never used concepts and attributes that will most probably have to finally be removed from the ontology since a priori useless. Evolution-based ontology validation is suitable to address the objectives of the ontology lifecycle presented in the next section.

5.2.5 Ontology-driven semantic interoperability lifecycle

Semantic interoperability can be driven by ontologies as shown in Figure 16.

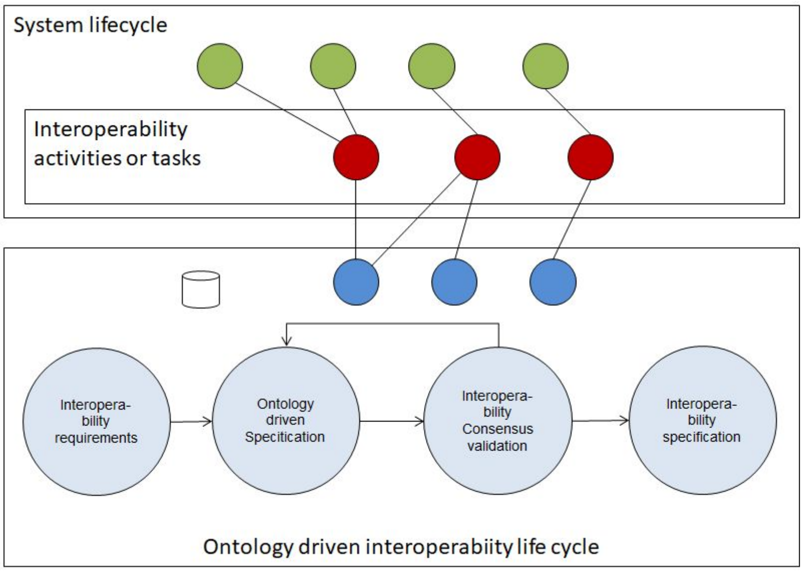


Figure 16. Ontology-driven semantic interoperability lifecycles

Table 11. Ontology-driven interoperability specification lifecycle process

Ontology-driven Interoperability specification lifecycle process	Description	
Interoperability specification requirements	Semantic interoperability ontology requirements definition	Define the type of knowledge that needs to be captured in the ontology (domain, cross domain and transversal, e.g. health, transport and security) Define the operational requirements (e.g. compatibility) Identify an ontology version management scheme
Ontology driven specification	Semantic interoperability ontology	Define the ontologies to be used, the part that is encapsulated, the part that is exposed and adapted

	structure co-creation	Seek consensus for standardisation
	Semantic interoperability ontology co-construction	Finalise or build the ontology that describes the interoperability point Seek consensus for standardisation
	Semantic interoperability ontology test and validation	Validate that the ontology is well formed and semantically consistent Validate that the exposed ontology is what is expected
Interoperability consensus validation	Semantic interoperability ontology commissioning and deployment	Acceptance by the ecosystem (e.g. community that will use the ontology) that the ontology is at suitable maturity level Integrate in the ontology version management
Publication	Semantic interoperability ontology maintenance	Update and enhance the exposed ontology Validate the updated ontology
	Semantic interoperability ontology decommissioning	Update the ontology version management

The ETSI document [22] describes in detail the ontology development process as shown in Figure 17.

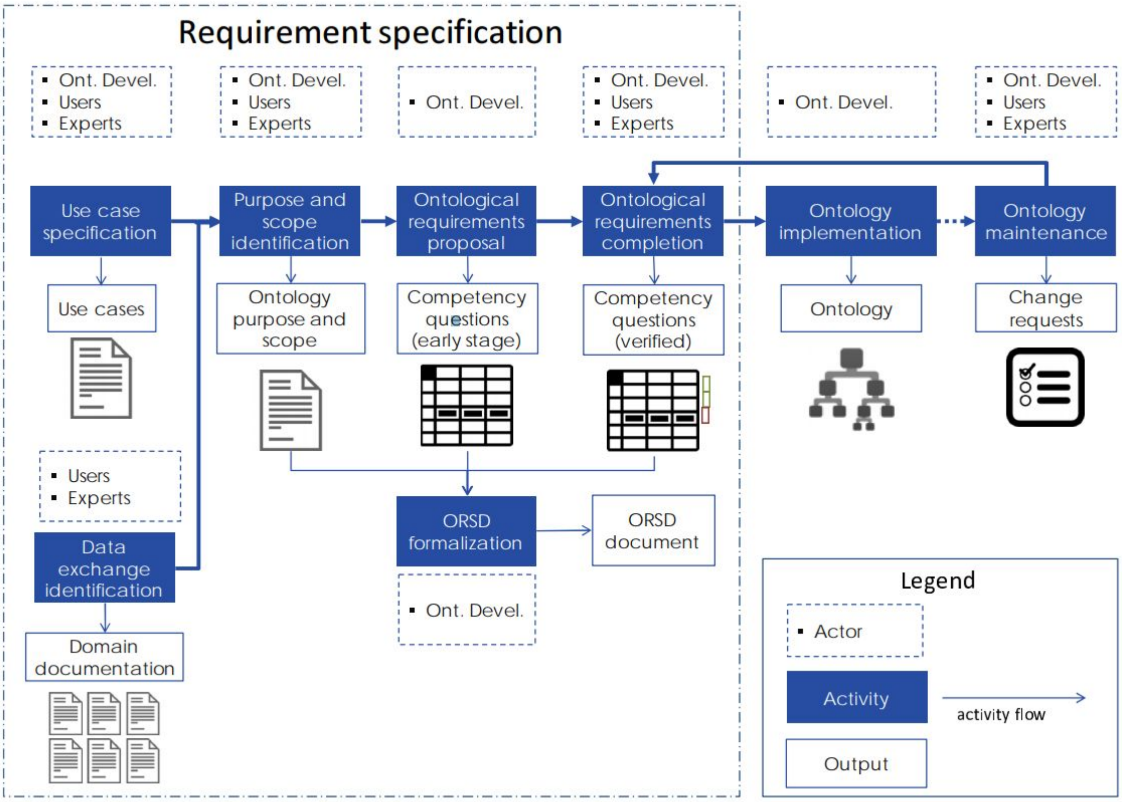


Figure 17. Ontology development process (from [22])

6 Recommendations for ontology-driven semantic interoperability standards

The following recommendations for ontology-driven semantic interoperability standards are:

- Providing guidance to ensure a standardised practice of ontology-driven interoperability. The overall guidance would be provided by ISO/IEC 21823-3 [30] which is under development, and it would be complemented by other types of guidance (e.g., on co-creation, modular design).

- Providing guidance on the creation and maintenance of reference ontologies. This includes assistance on ontology engineering and lifecycle management. This also involves the set up of a community of ontology practitioners to share and collect practices and tools.

- Developing ontology standards, including general ontologies and domain ontologies.

References

1. Semantic Web Layer Cake Tweak, Explained. [cited 20 Oct 2019]. Available: https://medium.com/openlink-software-blog/semantic-web-layer-cake-tweak-explained-6ba5c6ac3fab

2. Guarino N. Semantic matching: Formal ontological distinctions for information organization, extraction, and integration. Information Extraction A Multidisciplinary Approach to an Emerging Information Technology. 1997. pp. 139–170. doi:10.1007/3-540-63438-x_8

3. Bezerra C, Freitas F, Euzenat J, Zimmermann A. ModOnto: A tool for modularizing ontologies. Proceedings of the 3rd workshop on ontologies and their applications (Wonto). 2009.

4. Raad J, Cruz C. A Survey on Ontology Evaluation Methods. Proceedings of the International Conference on Knowledge Engineering and Ontology Development. 2015. Available: https://hal.archives-ouvertes.fr/hal-01274199/document

5. Gyrard A, Gaur M, Padhee S, Sheth A, Juganaru-Mathieu M. Knowledge Extraction for the Web of Things (KE4WoT). Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW ’18. 2018. doi:10.1145/3184558.3192305

6. Knowledge Extraction for the Web of Things (KE4WoT) Challenge 2018. [cited 20 Oct 2019]. Available: http://wiki.knoesis.org/index.php/KE4WoTChallengeWWW2018

7. OGC / W3C Semantic Sensor Network Incubator Group. Semantic Sensor Network Ontology (SSN). Available: https://www.w3.org/TR/vocab-ssn/

8. Sensor Observation, Sample and Actuator (SOSA) Ontology. [cited 20 Oct 2019]. Available: http://www.w3.org/ns/sosa/

9. Haller A, Janowicz K, Cox SJD, Lefrançois M, Taylor K, Le Phuoc D, et al. The modular SSN ontology: A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation. Semantic Web. 2018. pp. 9–32. doi:10.3233/sw-180320

10. Web of Things Architecture. Available: https://www.w3.org/TR/wot-architecture/

11. Web of Things - Thing Description. Available: https://www.w3.org/TR/wot-thing-description/

12. Web of Things (WoT) Thing Description Ontology. In: W3C Editor’s Draft 20 October 2019 [Internet]. [cited 20 Oct 2019]. Available: https://w3c.github.io/wot-thing-description/ontology/td.html

13. Ontologies used for oneM2M. In: oneM2M [Internet]. [cited 20 Oct 2019]. Available: http://www.onem2m.org/technical/onem2m-ontologies

14. Nachabe L, Girod-Genet M, El Hassan B. Unified Data Model for Wireless Sensor Network. IEEE Sensors Journal. 2015. pp. 3657–3667.

doi:10.1109/jsen.2015.2393951

15. ETSI. Smart Body Area Network (SmartBAN); Unified data representation formats, semantic and open data model. 10-2015. Report No.: TS 103 378 (V1.1.1). Available: https://www.etsi.org/deliver/etsi_ts/103300_103399/103378/01.01.01_60/ts_103378v010101p.pdf

16. Smart Applications REFerence Ontology (SAREF). [cited 20 Oct 2019]. Available: https://saref.etsi.org/saref#

17. Daniele L, den Hartog F, Roes J. Created in Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology. Formal Ontologies Meet Industry FOMI 2015. Cham: Springer; 2015. pp. 100–112.

18. ETSI. SmartM2M Smart Appliances; Reference Ontology and oneM2M Mapping. 03-2017. Report No.: TS 103 264 V2.1.1. Available: https://www.etsi.org/deliver/etsi_ts/103200_103299/103264/02.01.01_60/ts_103264v020101p.pdf

19. Moreira JL, Daniele LM, Pires LF, van Sinderen MJ, Wasielewska K, Szmeja P, et al. Towards iot platforms' integration: Semantic translations between W3C SSN and ETSI SAREF. Proceedings of SIS-IoT: Semantic Interoperability and Standardization in the IoT Workshop 2017.

20. SAREF Framework and Development Workflow. [cited 20 Oct 2019]. Available: https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?WKI_ID=57865

21. schema.org. [cited 20 Oct 2019]. Available: https://iot.schema.org/

22. ETSI. ETSI Technical Report: SmartM2M; Smart Appliances; SAREF extension investigation. 2017 Feb. Report No.: TR 103 411 V1.1.1. Available: https://www.etsi.org/deliver/etsi_tr/103400_103499/103411/01.01.01_60/tr_103411v010101p.pdf

23. ISO/IEC/IEEE. Systems and software engineering -- System life cycle processes. 2015. Report No.: 15288. Available: https://ieeexplore.ieee.org/document/4475828

24. International Organization for Standardization (Ginebra). ISO/IEC TR 24748-1 Technical Report: Systems and Software Engineering : Life Cycle Management. Guide for life cycle management. Guide de gestion du cycle de vie. 2010.

25. Lake JG. Report on Development of ISO Standard 15288 System Life Cycle Processes. INCOSE International Symposium. 1997. pp. 435–442. doi:10.1002/j.2334-5837.1997.tb02203.x

26. OWL-S: Semantic Markup for Web Services. 22 Nov 2004 [cited 20 Oct 2019]. Available: https://www.w3.org/Submission/OWL-S/

27. On-To-Knowledge (OTK) Methodology. [cited 20 Oct 2019]. Available: https://pdfs.semanticscholar.org/5021/a50d86b2afc5bccebd3105e79362bd4f2580.pdf

28. Neon methodology lifecycles. [cited 20 Oct 2019]. Available:

http://neon-project.org/web-content/media/book-chapters/Chapter-04.pdf

29. Noy NF, McGuinness DL. Ontology Development 101: A Guide to Creating Your First Ontology. Available: https://protege.stanford.edu/publications/ontology_development/ontology101.pdf

30. ISO/IEC/IEEE. Internet of Things (IoT) - Interoperability for IoT Systems - Part 3: Semantic interoperability. Standard under development. Report No.: 21823-3.

Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No.732240 (SynchroniCity) and No. 688467 (VICINITY); from ETSI under Specialist Task Forces 534, 556, and 566.
This work is partially funded by Hazards SEES NSF Award EAR 1520870, and KHealth NIH 1 R01 HD087132-01. The opinions expressed are those of the authors and do not reflect those of the sponsors.