International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Y.4563

(11/2021)

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

Internet of things and smart cities and communities – Services, applications, computation and data processing

---

Requirements and functional model to support data interoperability in Internet of things environments

Recommendation ITU-T Y.4563

ITU-T

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Enhancements to NGN	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Computing power networks	Y.2500–Y.2599
Packet-based Networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999
FUTURE NETWORKS	Y.3000–Y.3499
CLOUD COMPUTING	Y.3500–Y.3599
BIG DATA	Y.3600–Y.3799
QUANTUM KEY DISTRIBUTION NETWORKS	Y.3800–Y.3999
INTERNET OF THINGS AND SMART CITIES AND COMMUNITIES	
General	Y.4000–Y.4049
Definitions and terminologies	Y.4050–Y.4099
Requirements and use cases	Y.4100–Y.4249
Infrastructure, connectivity and networks	Y.4250–Y.4399
Frameworks, architectures and protocols	Y.4400–Y.4549
Services, applications, computation and data processing	**Y.4550–Y.4699**
Management, control and performance	Y.4700–Y.4799
Identification and security	Y.4800–Y.4899
Evaluation and assessment	Y.4900–Y.4999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T Y.4563

Requirements and functional model to support data interoperability in Internet of things environments

Summary

Recommendation ITU-T Y.4563 specifies the requirements and functional model to support data interoperability in Internet of things environments. Relevant requirements and technologies that support data interoperability have been proposed here.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.4563	2021-11-29	20	11.1002/1000/14819

Keywords

Data interoperability, Internet of things, ontology, semantic web.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Page

Recommendation ITU-T Y.4563

Requirements and functional model to support data interoperability in Internet of things environments

1 Scope

This Recommendation addresses the requirements and functional model for data interoperability in Internet of things (IoT) environments. The scope of this Recommendation covers several key requirements with respect to data interoperability in IoT environments and many important elements to fulfil these requirements. Specifically, this document covers the following:

– Overview of data interoperability in IoT environments;
– Requirements to support data interoperability;
– Functional model to support data interoperability;
– Functional components of the semantic mediation function;
– Functional components of the syntactic mediation function;
– Functional components of the object abstractions representation mediation function;
– Functional components of the interoperable data repositories.

NOTE 1 – For data interoperability with the web of object (WoO) based on [ITU-T Y.4452] see Annex A

NOTE 2 – For data interoperability approach of semantic and non-semantic data see Annex B.

NOTE 3 – For data interoperability example through semantic mediation see Appendix III.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.4401] Recommendation ITU-T Y.4401/Y.2068 (2015), *Functional framework and capabilities of the Internet of things*.

[ITU-T Y.4452] Recommendation ITU-T Y.4452 (2016), *Functional framework of web of objects*.

[ITU-T Y.4459] Recommendation ITU-T Y.4459 (2020), *Digital entity architecture framework for Internet of things interoperability*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following term defined elsewhere:

3.1.1 web of objects (WoO) [ITU-T Y.4452]: A concept and approach to incorporate virtual objects on the web and to facilitate the creation of IoT service components.

NOTE – It provides a service architectural model to support simple development, deployment and operation of the IoT services on the web (for WoO example see Appendix III).

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 common data model: A data model that allows the transformation of data into a single common data format from different formats that are collected from heterogeneous sources. For transformation into a common format or a data model, common terminologies, vocabularies, schemes need to be followed.

3.2.2 data interoperability: The ability of two or more systems or components to exchange data and to use the data that has been exchanged.

3.2.3 semantic data interoperability: The ability of two or more computer systems to automatically interpret data exchanged between them meaningfully and accurately in order to produce useful results as defined by the end users of both systems.

3.2.4 semantic data model: A conceptual model that includes the semantic information of instances. Semantic information of data defines the meaning of the data based on the context of the data.

3.2.5 syntactic data interoperability: The ability of two or more systems capable of communicating and exchanging data through specified data formats and communication protocols.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

API Application Programming Interface

CDM Common Data Model

CSV Comma Separated Values

CVO Composite Virtual Object

HTML HyperText Markup Language

HTTP Hypertext Transfer Protocol

IoT Internet of Things

JSON JavaScript Object Notation

LOV Linked Open Vocabularies

OCI Object Classification Item

OWL Web Ontology Language

RDF Resource Description Framework

REST Representational State Transfer

SKOS Simple Knowledge Organization System

SOR Semantic Ontology Registry

SSN Semantic Sensor Network

TD Thing Description

URI Uniform Resource Identifier

VO Virtual Object

WoO Web of Objects

WoT Web of Thing

XML Extensible Markup Language

5 Conventions

In this Recommendation:

The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this Recommendation is to be claimed.

The keywords "is recommended" indicate a requirement that is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

The keywords "is not recommended" indicate a requirement that is not recommended but which is not specifically prohibited. Thus, conformance with this specification can still be claimed even if this requirement is present.

The keywords "can optionally" indicate an optional requirement that is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 Overview of data interoperability in IoT environments

Internet of things (IoT) envisages billions of objects connected to the internet where numerous services will be deployed to support scalable and interoperable applications. However, to achieve this vision as indicated in IoT interoperability observation of [ITU-T Y.4459], the most challenging issue is handling interoperability of information and its services. Data interoperability is the ability of applications to exchange data between IoT environments and the use of this exchanged data in a meaningful way to support service provisioning. In general, data interoperability is concerned with the capability of communications between IoT environments that might have different forms including transfer, exchange, transformation and integration of data.

Data interoperability can be achieved in different ways to provide seamless integration of services in heterogeneous IoT environments (for examples of data interoperability approaches see Appendix III). Several types of interoperability can be considered to fully realize data interoperability in IoT platforms. These include semantic mediation, syntactical mediation, and object abstraction representation mediation. In order to achieve a high-level object abstraction interoperability, semantic and syntactical interoperability is necessary.

Figure 6-1 illustrates a view of multiplatform data interoperability and the interaction of core components. Achieving interoperability from all these views results in organization level interoperability. The data representation in different formats and information models complicates its usability for IoT applications. This figure may include several new and legacy heterogeneous IoT systems such as buildings, transportation, and traffic management systems that make use of different devices and protocols to enable IoT services. There are three types of functions to realize data interoperability among IoT platforms, namely, the semantic mediation functions, syntactical mediation functions and object abstraction representation mediation functions. The details on these functions are described as follows and further elaborated in clause 9.

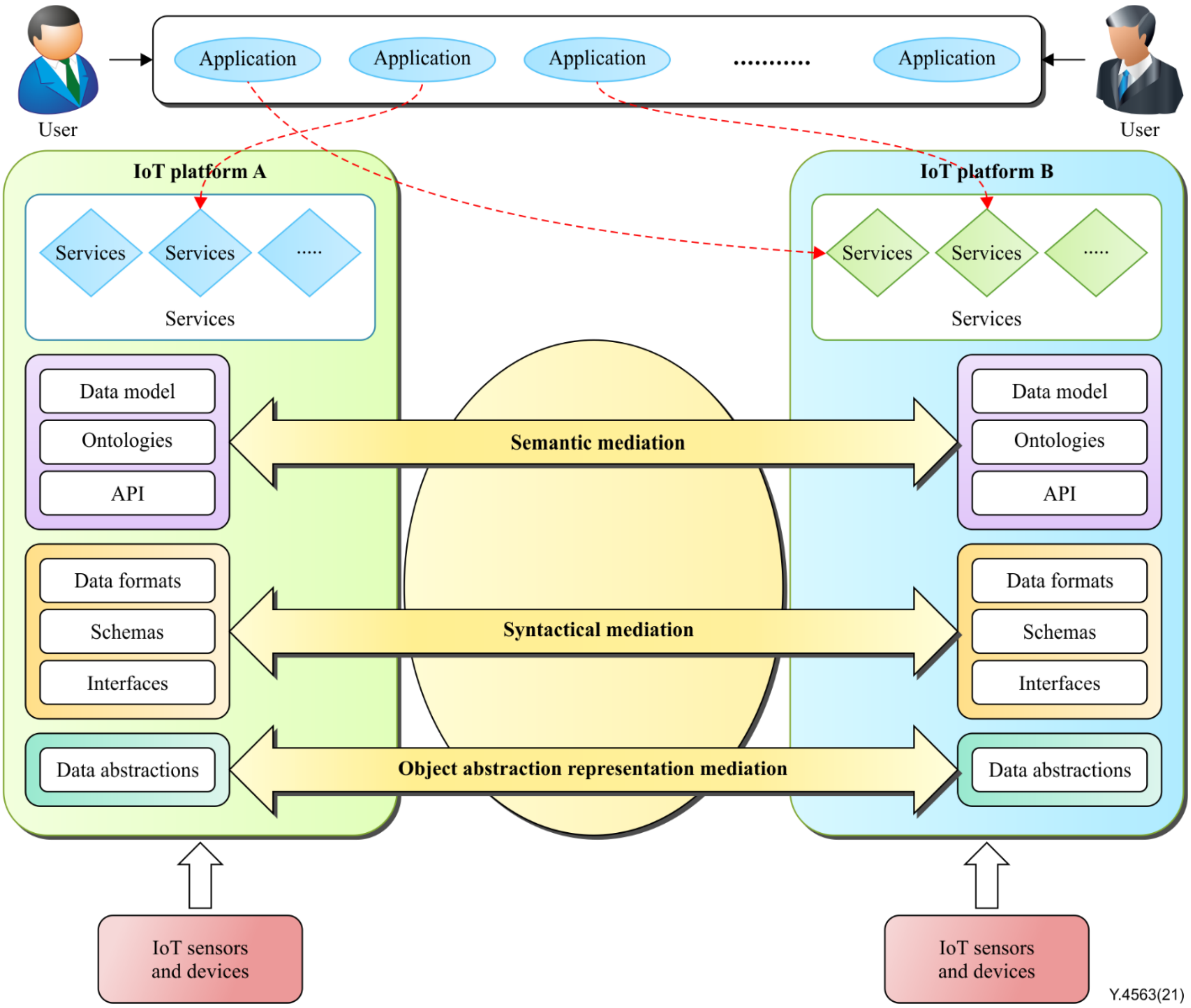

Figure 6-1 – A view of data interoperability in heterogeneous IoT environments

– Semantic mediation:

Semantic mediation is concerned with the meaning of data. Consensus on meaning is required while exchanging data across systems. Semantic mediation defines the true meaning of the contents that are generated by IoT devices and mutually agreed by a different system that uses these contents. The semantic interoperability will enable different stakeholders to access and understand data unambiguously.

– Syntactical mediation:

Heterogeneous IoT devices generate data that are stored and used in different formats. Syntactical mediation is concerned with the data formats, syntax, and coding methods. Protocols used by IoT devices use standard syntax for the communication of data. These are expressed in diverse formats such as the Extensible markup language (XML), JavaScript object notation (JSON) or Hypertext markup language (HTML).

– Object abstraction representation mediation:

Object abstraction representation mediation provides the functional capabilities to support diverse object abstractions in terms of semantic and syntactic data in the representation, and metadata description and coding.

7 Requirements to support data interoperability

To support data interoperability in the IoT environment, the following requirements are required to be identified in the three dimensions of data interoperability i.e., semantic, syntactical and object abstraction representation interoperability.

7.1 Requirements for semantic data interoperability

To ensure semantic data interoperability the following aspects are required to be identified:

– Semantic data modelling: The semantic representation of data is required to express a common understanding across systems. A semantic representation model is necessary to provide the conceptual understanding of data as well the relationship among entities.

– Semantic integration and sharing: Mechanism for the linking of data based on semantic ontology models is required. The linking mechanism is required to be efficient to support dynamic integration and sharing of data.

– Semantic annotation of data: A semantic annotation mechanism is required to support the annotation of data coming from heterogeneous sources. Semantic annotation is required to be supported with a well-defined set of metadata to express the features of diverse IoT data.

– Semantic data management: The abstract semantic representations of IoT data are required to be managed through a management service. A suite of well-defined services is required to manage the data by allowing its access, retrieval, and storage operations.

– Semantic ontology alignment and mapping: It is required to provide an improved ontology alignment to support semantic interoperability. The ontology alignment techniques with enhanced accuracies can enable and improve interoperability across different systems.

– Semantic representation of knowledge: In IoT, providing rules with knowledge representation supports reasoning on the data which enhances its value. The information model defines the format to contain the information. It is required to be semantically rich and expressive enough to represent different forms of the objects being maintained. It is also required to scale well in evolving IoT technology. The information model should be flexible enough to represent semantic information. Ontologies in IoT provide an option to exchange the knowledge by giving the semantics required and enhancing the data in the information model.

– Semantic data transformation: Mechanism for the transformation of a semantic format to another is required. In the case of domain specific model, it can optionally provide transformation service among heterogeneous semantic data models.

7.2 Requirements for syntactical data interoperability

To enable data interoperability at a syntactical level among heterogeneous IoT systems the following requirements are required to be identified:

– The syntactical format identification, registration and management mechanisms;

– The syntactical format description models that provide expressivity in definitions;

– Well defined syntactical templates that can help generate response objects on the initial template instances;

– Syntactical translation mechanism to generate the transformation based on the provided templates;

– Syntactical formats registry is required to provide a repository of formats of diverse registered platforms;

– Well defined syntactical metadata schema and their mapping mechanisms;

– Verification methods for format translation and conversion process to validate the effectiveness of translation mechanism.

7.3 Requirements for object abstraction representation interoperability

The object abstraction representation interoperability is required to provide interoperable sharing of data in heterogeneous IoT environments. The following are required to characterize the object abstraction representation:

– Mechanism for creation and management of abstract data representations;
– Provision of semantics in the data representation model to maintain the same meaning across different data models;
– Uniform syntactic representation of data in standard formats;
– Description of metadata and their coding function to express diverse core data models;
– Provision of data and metadata profiles that can express the object abstract representations for different systems;
– Mechanism to generate object abstract representations profiles from different data.

8 Functional model to support data interoperability

The mediation function to support data interoperability in heterogeneous IoT environments consists of three dimensional functions and multiple featured repositories shown in Figure 8-1, and each component of the functional mediation model provides an individual functionality which is further described below:

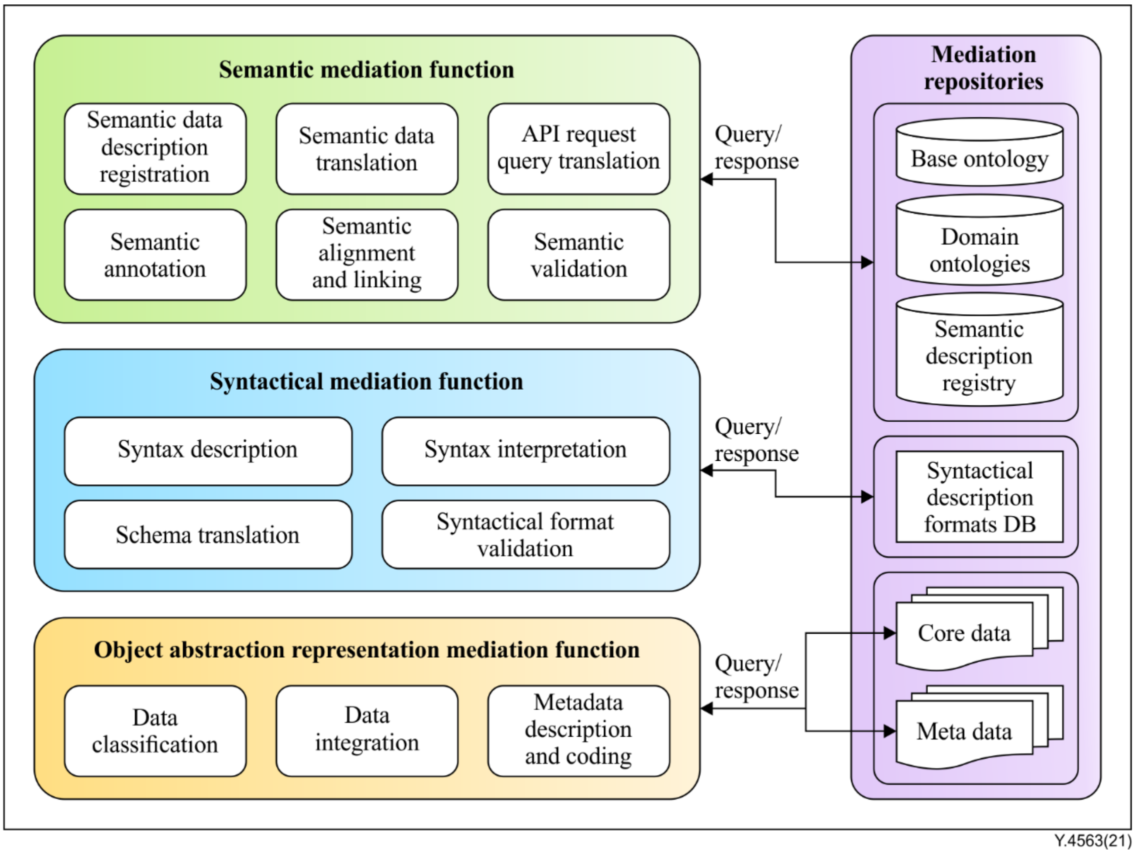

Figure 8-1 – Three dimensional functions and related repositories to support data interoperability

– Semantic mediation function: offers the mechanisms for semantic translation and linking of the data from different platforms (see clause 8.1);
– Syntactical mediation function: provides a syntax level mediation of different data formats, schemas and interfaces (see clause 8.2);
– Object abstraction representation mediation function: provides the mediation to support the heterogeneous objects data abstraction, classification and categorization (see clause 8.3).

8.1 Semantic mediation function

The semantic mediation function is responsible for supporting six sub-functions as shown below:

– Semantic data description registration: provides a registration mechanism, where platform level semantic data description formats are registered in the semantic description registry. The functional entity of semantic data description registration enables a useful record of semantic formats that can be used by semantic translation function;
– Semantic data translation: enables the translation of data formats to semantic formats that have been registered by the functional entity of semantic data description registration. The translation of data to semantic formats is achieved through the defined domain ontology;
– Application programming interface (API) request query translation: provides an interface to receive requests for the data to be translated to a particular registered format. The request for data cannot be directly provisioned as different platforms have diverse formats. The API request translation function provides a common way to translate the requests to specific API formats in order to provide seamless interoperability;
– Semantic annotation: enables the annotation of the data based on the standard ontology and semantic data model. It provides annotation using the selected annotation description language. It provisions data interoperability through specifying the particular function of IoT resources, their information and their operations which can be understood by other services;
– Semantic alignment and linking: provides an alignment strategy based on the defined alignment algorithm. The goal is to align the information model or the semantic schema with another schema to represent the same information but with different semantics. It enables to resolve semantic heterogeneity in systems where models have different meanings or ontologies;
– Semantic validation: is used to verify the semantic conversion of data with defined schema. It constitutes the mechanism to validate the semantic structure of the data with a validation test case defined on the basis of semantic ontology.

8.2 Syntactical mediation function

The syntactical mediation function supports four sub-functions to provide syntactical data interoperability as shown below:

– Syntax description: provides a registration of syntaxes for the platforms. By using this function, platform level syntax description formats are registered in the registry database. This function enables useful records of syntactical formats for the platforms which enable syntax level interoperability;
– Syntax interpretation: provides conversion among diverse data formats. In case of API requests to individual platforms, this functional component translates the queries specific to the platforms, for which data or services are requested;
– Schema translation: enables the translation mechanism at the schema level. Different platforms make use of different schemas to describe the data. This function provides the conversion to interoperate the schemas at the platform level;

– Syntactical validation: is used to verify the syntactical integrity of the translation. It constitutes the mechanism to validate the syntactical structure of the data based on the defined base schema.

8.3 Object abstraction representation mediation function

The object abstraction representation mediation function supports three sub-functions to provide interoperable object abstractions representation as shown below:

– Data classification: provides the functional capability to perform the classification and categorization of data. It enables the classified representation to be understandable through abstract representation model;

– Data integration: enables the functional capabilities to extract and integrate multiple IoT data. As data can be from different sources, it provides procedures to integrate the data in standard formats such as a common data model (CDM);

– Metadata description and coding: allows the metadata to be assigned to the data converted to the representation format. It enables to assign codes and other metadata descriptions to the core data in order to identify its sources.

8.4 Mediation repositories

Mediation repositories provide multiple featured repositories to support storage functions for the query and response for translation and caching in the provision of data interoperability. This includes four kinds of repositories:

– Base ontology: supports models to define core models for semantic interoperability of platforms;

– Domain ontology: supports models to define the domain specific models of the platforms;

– Semantic registry: supports registries in the description of semantic data;

– Syntactic description database (DB): supports the database to facilitate persistence of different data formats.

9 Functional components of the semantic mediation function

The functional components of the semantic mediation function identify the details of six sub-functional components to support the semantic data interoperability in IoT environments as follows:

– Semantic data description registration;

– Semantic data translation;

– API request query translation;

– Semantic annotation;

– Semantic alignment and linking;

– Semantic validation.

9.1 Semantic data description registration

The semantic data description registration function provides a registration capability, where platform level semantic data description formats are registered in the semantic description registry. This function enables a useful record of semantic formats that can be used by the semantic translation functions.

The functional components of semantic data description registration sub-function (as illustrated in the boxes of Figure 9-1) are as follows:

– Ontology discovery: provides the search and matching of ontology records in the semantic description registry;
– Ontology registration management: provides the functional capability to register and manage the semantic ontology models.

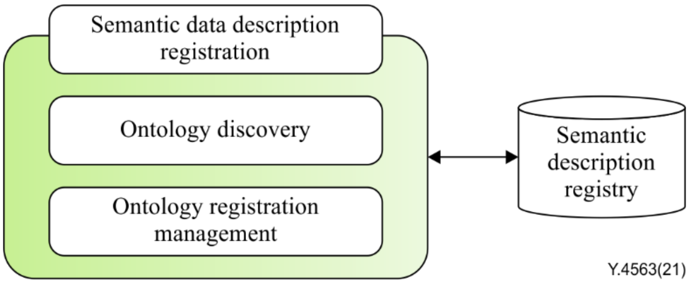

Figure 9-1 – Functional components in a semantic data description registration sub-function

9.2 Semantic data translation

The semantic data translation sub-function enables the translation of data formats to semantic formats that have been registered by the semantic data description registration sub-function. The translation of data to semantic formats is done through the defined domain ontology.

The functional components of a semantic data translation sub-function (as shown in Figure 9-2) are as follows:

– Base ontology translator: delivers the functional capability of translation of concepts from a domain ontology model to the base ontology model;
– Data model: provides the capability to express the semantic meaning of the exchanged data using the information objects.

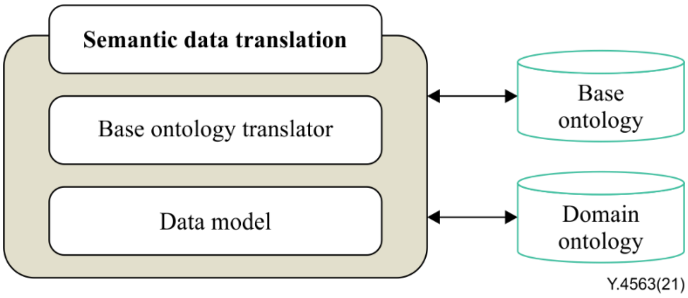

Figure 9-2 – Functional components in semantic data translation sub-function

In the description of the data model, semantic data contains semantic information regarding the data. The meaning of data differs from one context to another context. Semantic information of data defines the meaning of data based on the context of the data. Different applications use the same data differently based on the specification of the domain. A semantic data model is an abstract model that defines the meaning of a data by using the relationship between objects within the context. A semantic data model also describes how a virtual object (VO) is related to the real world object. A semantic model of a data ('Item') has been shown in Figure 9-3.

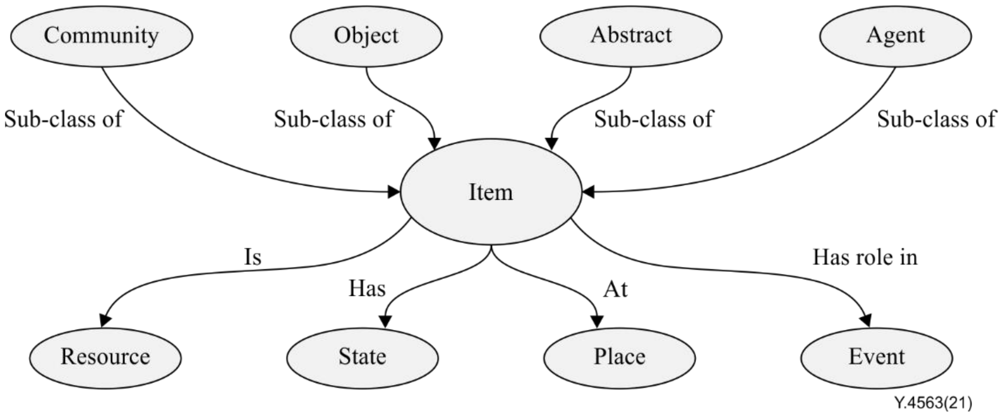

Figure 9-3 – A semantic model of a data ('Item')

A semantic data model has the following characteristics:

– Enables the interoperating in IoT environments to interpret the semantic meaning of the exchanged data using the information of the objects;

– The object relationship with other objects is expressed as a triple format;

– In a triple, the relationship between an object with another object is expressed using a binary relation;

– The triple includes subject-predicate-object relationship form. The first term in the triple is called subject; which is an object where a statement is expressed. The second term in the triple is a predicate which is the property that makes a relationship between the subject and the object. The third term in the triple is an object that represents a resource in the fact. Thus, the example of the fact becomes 'Richard has a smartphone';

– A fact must include a kind of relationship between a subject and multiple objects. The meaning of the relationship is also required to interpret the meaning of the fact;

– The kind of relationship needs to be standardized as well.

9.3 API request query translation

This sub-function delivers an interface to receive requests for the data to be translated to a particular registered semantic format. Since different platforms have diverse formats, to support requests their translation needs to be provided. The API request query translation sub-function provides a common way to translate the requests to a specific API format, in order to provision seamless semantic interoperability.

The functional components of the API request query translation sub-function (as illustrated in the boxes in Figure 9-4) are as follows:

– API translator: provides the functional capability to translate the API request to a target request format;

– Query formation: provides the functional capability to generate new formatted queries;

– API formats: provides the API formats of the registered platforms.

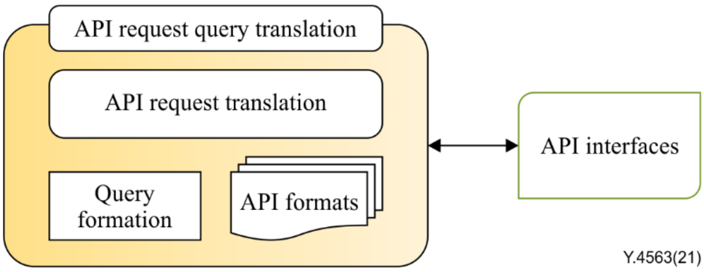

Figure 9-4 – Functional components of an API request query translation sub-function

9.4 Semantic annotation

This sub-function enables the annotation of the data based on the standard ontology and semantic data model. It provides annotation using the selected annotation description language. Semantic annotation allows the enrichment of content by linking machine processed information to the extracted concept of a semantic ontology. Semantic annotation allows attaching or adding additional information of the concepts of the ontology to the existing contents. The semantic annotation changes an isolated ontology into an ontology that can be interpreted, shared, and reused by other ontologies.

The capabilities of a semantic annotation sub-function are characterized by the following:

– The relationship between concepts and ontologies;

– Links information source to an ontology;

– Assigns semantic concepts and properties to the target data.

9.5 Semantic alignment and linking

The semantic alignment and linking sub-function enables the alignment and management of the source and the target semantic schemas.

The functional components of the semantic alignment and linking sub-function (as illustrated in the boxes of Figure 9-5) are as follows:

– Semantic ontology manager: includes the capability to manage the ontology aligner;

– Ontology aligner: provides the capability for semantic ontology alignment. It takes the source and target ontology models and returns the alignment results;

– Entity loader: provides the capability to load the entities from the source and target ontologies.

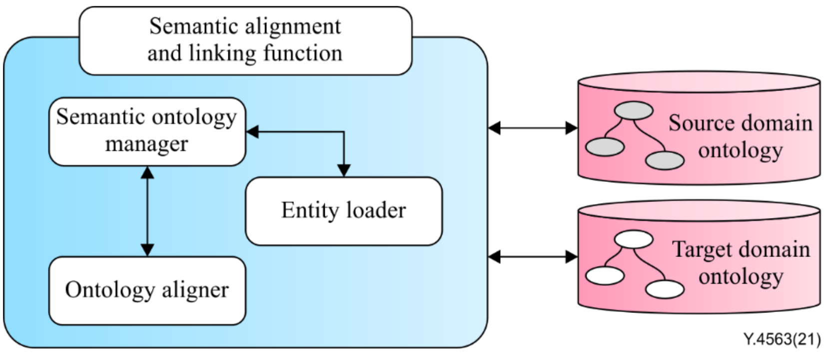

Figure 9-5 – Functional components of a semantic alignment and linking sub-function

Any IoT application that deals with several ontologies must provide a semantic mapping or a matching mechanism to ensure interoperability. When ontologies are matched, they are indeed aligned, which represent the correspondence among different ontologies. This kind of correspondence enables heterogeneous ontologies with different semantics to be interoperable. The identified correspondence among ontologies is a set of rules which transform from one to another ontology to enable the integration of the information. The requirement is to find out the correspondence among domains. This correspondence can be detected using the alignment of matching techniques.

An example of the alignment between two ontologies has been illustrated in the Figure 9-6. The names in the circles represent the concepts whereas the values on the arrows define the confidence among the concepts. The confidence is also referred as the value of similarity. The mapping of matching techniques generates this similarity value. Higher the value between two entities a more exact match is assumed.

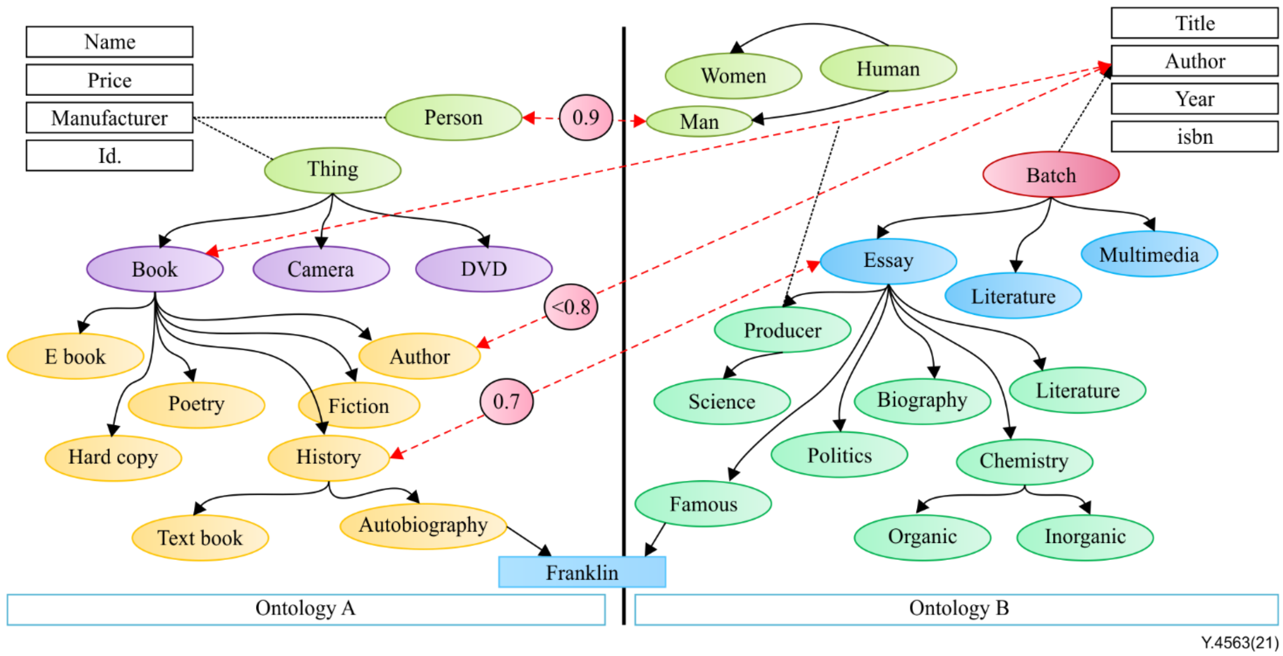

Figure 9-6 – Example of an ontology alignment

Moreover, several other aspects need to be considered for ontology alignment, these include:

– Efficiency consideration of ontology alignment methods;

– Evaluation of the ontology matching;

– The involvement of the user in the matching process.

9.6 Semantic validation

The sub-function is used to verify the semantic conversion of data with a defined schema. It constitutes the mechanism to validate the semantic structure of the data with a validation test case defined on the bases of semantic ontology.

The functional components of semantic validation sub-function (as shown in Figure 9-7) are as follows:

– Validator: provides the validation function on the provided input test case to validate the semantic alignment;

– Test execution: provides the execution facility to perform the validation for the alignment function.

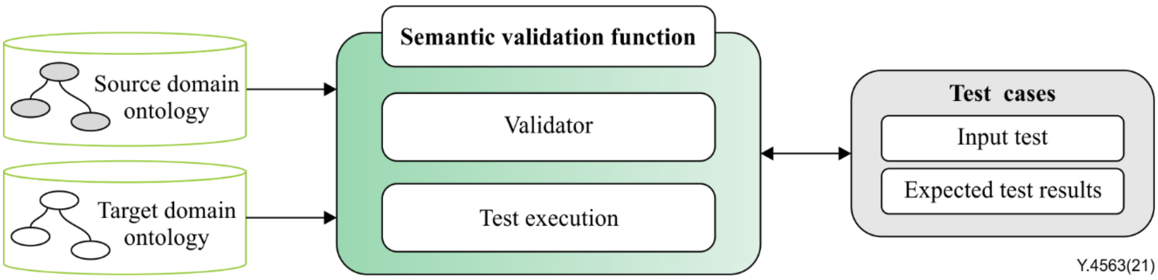

Figure 9-7 – Functional components of a semantic validation sub-function

10 Functional components of syntactic mediation functions

This clause provides the functional components of syntactical mediation functions in accordance to support syntactical data interoperability of IoT environments as follows.

– Syntax description;

– Syntax interpreter;

– Schema translation;

– Syntactical validation;

– Interaction of syntactical mediation function.

In heterogeneous IoT environments, syntactical data interoperability is considered as the interpretation of data structures and the formats that need to be exchanged among diverse IoT systems. Each resource in the IoT environment exposes some structure based on its underlying schema using a defined interface. For example, the representational state transfer (REST) API exposes an interface based on the REST methods.

The syntactic interoperability from a scenario where a sender encodes, serializes, and sends a message using some rules that are based on some grammar and the receiver deserializes and decodes the message using some rules based on similar grammar. The problem may arise if the rules used by the sender are different from the rules the receiver uses. This raises the syntactical interoperability problem in such a scenario.

Figure 10-1 shows the functional components of a syntactic mediation function to support syntactic data interoperability.

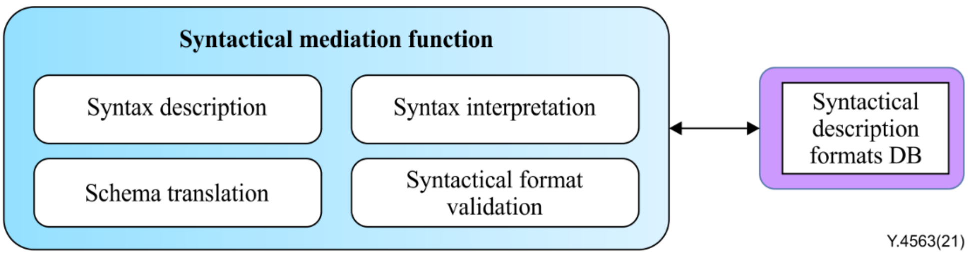

Figure 10-1 – Functional components of a syntactic mediation function

10.1 Syntax description

The syntax description sub-function provides a registration of syntaxes for the platforms. This sub-function provides a platform level syntax description format to be registered in the registry database.

The functional components of the syntax description sub-function are as follows:

– Syntax registration interface: provides an interface to record the platform syntax profiles in the syntactical description formats DB;

– Template discovery: provides the functional capability to discover existing syntax templates;
– Syntactical metadata management: provides the functional capability to manage metadata related to the syntactical models.

Syntax description function can utilize a common data model to represent the diverse formats in a unified way. A common data model (CDM) is a set of data elements that can be mapped among different interoperating systems (see the approach shown in Appendix IV).

10.2 Syntax interpreter

The syntax interpreter sub-function provides conversion and translation among diverse data formats. In the case of API requests to individual platforms, this functional component translates the syntax specific to the platforms, for which data or services are requested.

The functional components of syntax interpreter sub-function are as follows:

– API syntax converter: provides the capability to translate the API syntax description to a standard format;
– Profile manager: delivers the capability to the store and retrieve syntax profiles.

10.3 Schema translation sub-function

This schema translation sub-function enables the translation mechanism at the schema level. Different platforms make use of different schemas to describe the data syntax. The function provides the conversion to interoperate the schemas at the platform level.

The functional components of a schema translation sub-function are as follows:

– Core schema translator: provides the functional capability to translate the core data schema with respect to syntactical conversion in a CDM;
– Metadata schema translator: provides the functional capability to convert metadata formats to describe core data into common metadata formats.

The metadata is described as the information related to data such as data types, description, etc. A metadata is an application oriented predefined model that describes the data, data types, data files, the logics and constraints that are required to analyse the data for application specific decision making. Due to the increase in the amount of data, the metadata model describes the data as well as links them with another model as a knowledge base.

10.4 Syntactical validation sub-function

The syntax validation sub-function verifies the syntactical integrity of the translation. It constitutes the mechanism to validate the syntactical structure of the data based on the defined core schema.

The functional components of a semantic validation sub-function are as follows:

– Syntax format validator: delivers the syntax level validation function on the provided input syntax profile to validate the translation;
– Syntactical testing container: provides the run time facility to execute the validation for the syntactical translation function.

10.5 Interaction of a syntactical mediation sub-function

The component-based interaction procedure among the syntactic mediation functional units has been shown in the Figure 10-2. To provide syntactical interoperability a syntax translation function generates the conversion schema through the syntax interpreter service. The main goal of the syntactical mediation sub-function is to generate the syntactical alignment based on the syntax that is chosen for the subject alignment.

The syntax description and management services throughout the syntax translation perform three major tasks that include the following:

– Discovery of syntax template from the template repository;

– Syntax registration once a conversion has been performed successfully;

– Syntactical metadata management to support additional syntax level checking.

To provide the verification of a syntactical conversion, syntax format validation function tests the generated translation schema against a pre-validated test case in order to ensure the integrity of the mediation process.

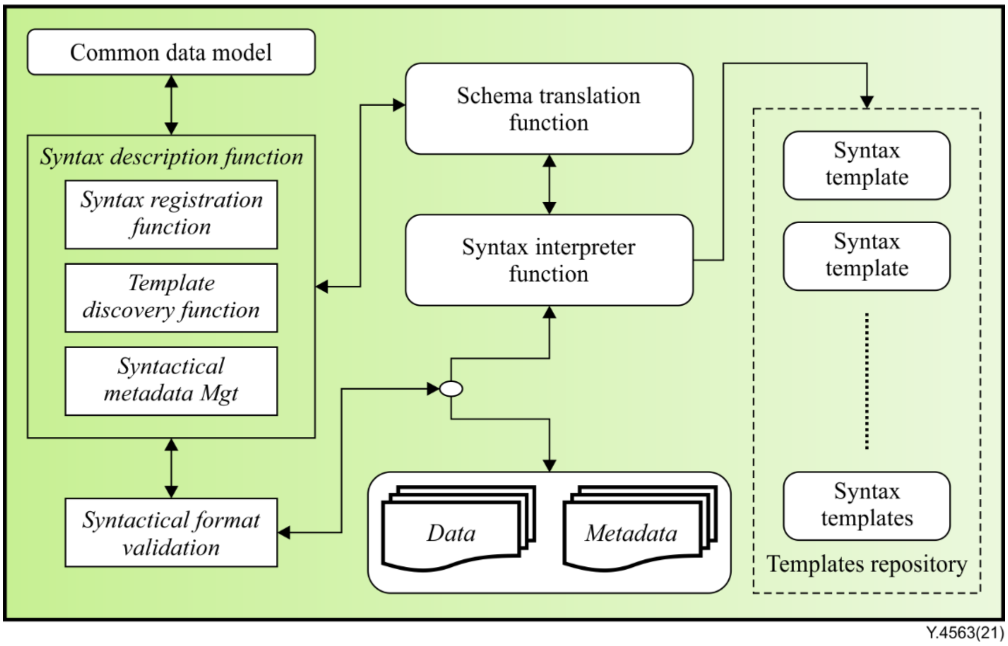

Figure 10-2 – Syntactical mediation interaction: Component based interaction procedure among the syntactic mediation functional units

NOTE – For syntax level interoperability formats in other applications, see Appendix II.

11 Functional components of an object abstraction representation mediation

This clause provides the functional components of object abstraction representation functions in accordance with the organizational data interoperability. Figure 11-1 illustrates the functional components of the objects abstraction representation mediation function.

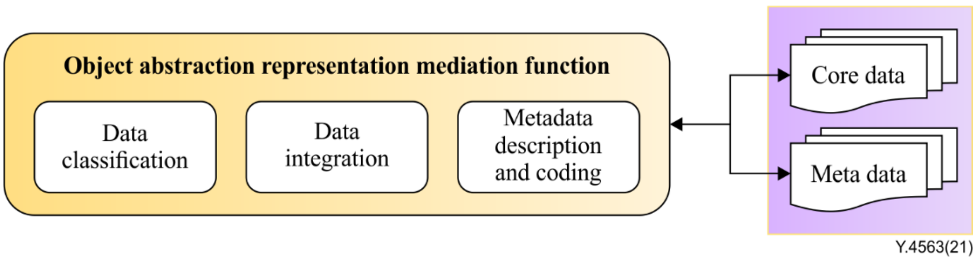

Figure 11-1 – Functional components of the objects abstraction representation mediation function

11.1 Data classification

Data classification function provides the capability for the classification and categorization of data. It enables the classified representation to be understandable through an abstract representation model.

The functional capabilities of the data classification functions are as follows:

– Classifies the data based on the metadata available with the core data;

– Tagging the data according to the abstract representation model.

11.2 Data integration

Data integration function provides the capability for the extraction and integration of data. If the data is from different sources, the function provides procedures to integrate the data in standard formats.

The functional capabilities of data integration functions are categorized into three categories:

– Integration based on the classification category, when data has been classified and the category is known;

– Integration based on the metadata description, when the data has not been classified, and the metadata provides sufficient information details to perform integration;

– Integration based on the fixed object abstraction ontology, when integration plan has already been setup to support the integration based on the object abstraction ontology.

11.3 Metadata description and coding

Metadata descriptions and the coding sub-function provides the capability to assign new metadata or assign additional metadata to the converted data. It enables to assign codes and other metadata description to the core data in order to identify its sources. Metadata helps in several tasks such as:

– Discovery, identification and classification of data;

– Describes the relationship and characteristics of data;

– Provides when the data is created and transformed and how, types of data files and other technical information are stored.

For sharing and reuse of data in IoT environments, ontology is a fundamental principle. A metadata ontology includes the elements of metadata and describes their relationships.

The metadata description function uses a metadata ontology as a library to describe and publish metadata. A metadata ontology allows users to search, refer and evaluate metadata, uses the metadata standard or the controlled vocabularies.

11.4 Interaction of object abstraction representation mediation functions

The functional interaction to generate object abstraction representation is depicted as shown in Figure 11-2. Here, the data representation processing function is the central processing component of an object abstraction which enables the instantiation of abstraction from an existing profile by calling on the generation function. It also provides the data classification and integration mechanism to categorize and integrate data respectively.

Data abstraction profiles are a set of profiles expressed as $P = \{P_1, ..., P_n\}$, each represents a unique profile based on the type of data. Object abstraction generation function generates an abstract representation by sending the selection profile to the representation configuration management function. The configuration function discovers the appropriate template that matches the profile and extracts the concepts from the representation ontology to generate the data abstraction. These generated abstractions are registered in the data representation repository to enable reuse whenever they are required.

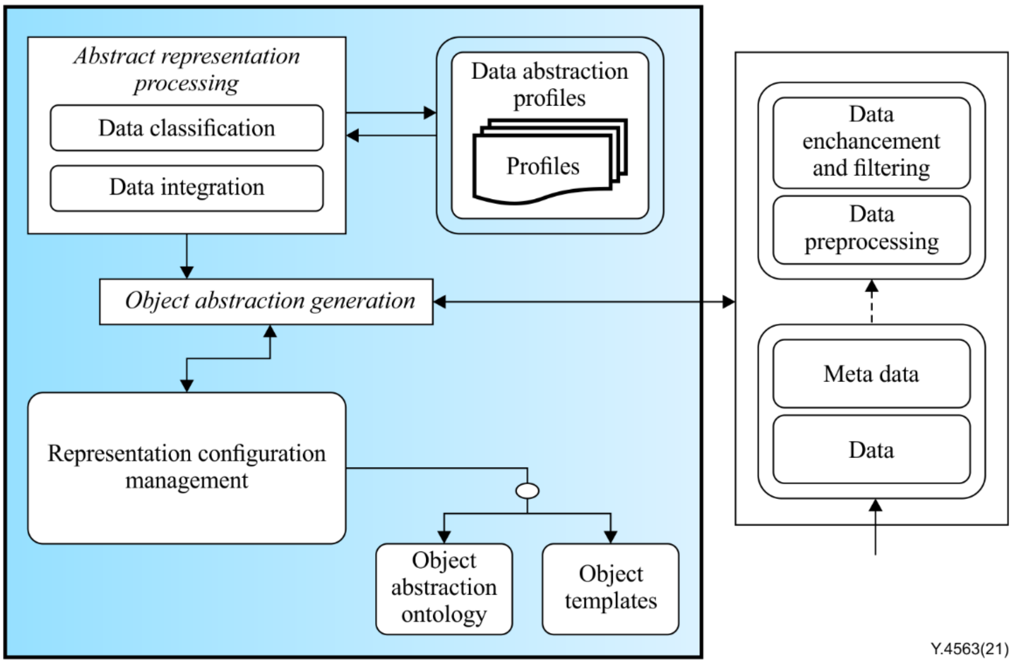

Figure 11-2 – Object abstraction mediation functions interaction

12 Functional components of the mediation repositories

This clause provides the details on data repositories to support interoperability functions in accordance with the functional model of data interoperability given in clause 8.

12.1 Base ontology repository

Base ontology repository provides the capability for persistent storage and retrieval on the entities of a base ontology model.

The base ontology model consists of the following capabilities:

– The semantic base ontology provides the generic concepts to support semantic interoperability;

– Base ontology model has an interface to the standard W3C SSN ontology and other external ontologies to express and provide interoperable features with existing systems.

Components in a base ontology model include:

– Base domain: defines the subject for which a fact is stated;

– Base range: defines the object of the fact of the base vocab;

– Base object properties: makes the relationship between subject and object in the base vocab;

– Base data properties: describes the instance using attributes of the instance in base vocab;

– Base annotations: annotates or describes various components in a base ontology.

NOTE – For ontologies and description models in IoT data interoperability, see Appendix I.

12.2 Domain ontology repository

To express the semantics of any concepts, and to organize them in a standard resource description framework (RDF) graph, semantic ontology is used. In a semantic ontology, concepts are expressed as a hierarchy of classes and subclasses, and properties are used to make relationships among them that represent an RDF graph. In the semantic web, RDF vocabularies are known as RDF schema or ontology.

A domain's semantic ontology provides the following features:

– Understanding the structure of information in an IoT domain;
– Capability to reuse the domain knowledge;
– Allows reuse of concepts and properties in the ontology;
– If concepts and properties are not available locally then similar concepts and properties are searched in other ontologies that can support similar functionalities;
– Enables inference on domain knowledge to make decisions.

Components in a domain ontology model include:

– Domain: defines the subject for which a fact is stated;
– Range: defines the object of the fact;
– Object properties: makes the relationship between subject and object;
– Data properties: describes the instance using attributes of the instance;
– Annotations: annotates or describes various components in an ontology.

W3C has a standardized number of RDF vocabularies to organize and to express concepts. An ontology allows designing a model for expressing concepts, constraints and relationships based on domain specifications.

12.3 Semantic registry

The semantic registry provides the following capabilities:

– To discover the interoperable object alignment records;
– To store the interoperable object alignment records;
– To retrieve the interoperable object alignment records.

12.4 Syntactical description formats DB

Syntactic description DB is the storage repository to facilitate the persistence of different data formats. It provides the following capabilities:

– Persistent storage of syntactical profiles of registered syntactical formats;
– Capability to store syntactical descriptions of different data;
– Capability to retrieve syntactical descriptions of different data;
– Capability to store and maintain syntactical descriptions metadata.

Annex A

Interoperability with web of objects (WoO) based model

(This annex forms an integral part of this Recommendation.)

This Annex A provides details on the web of object (WoO) based approach for interoperability. The underlying IoT framework in the following model is defined in [ITU-T Y.4452].

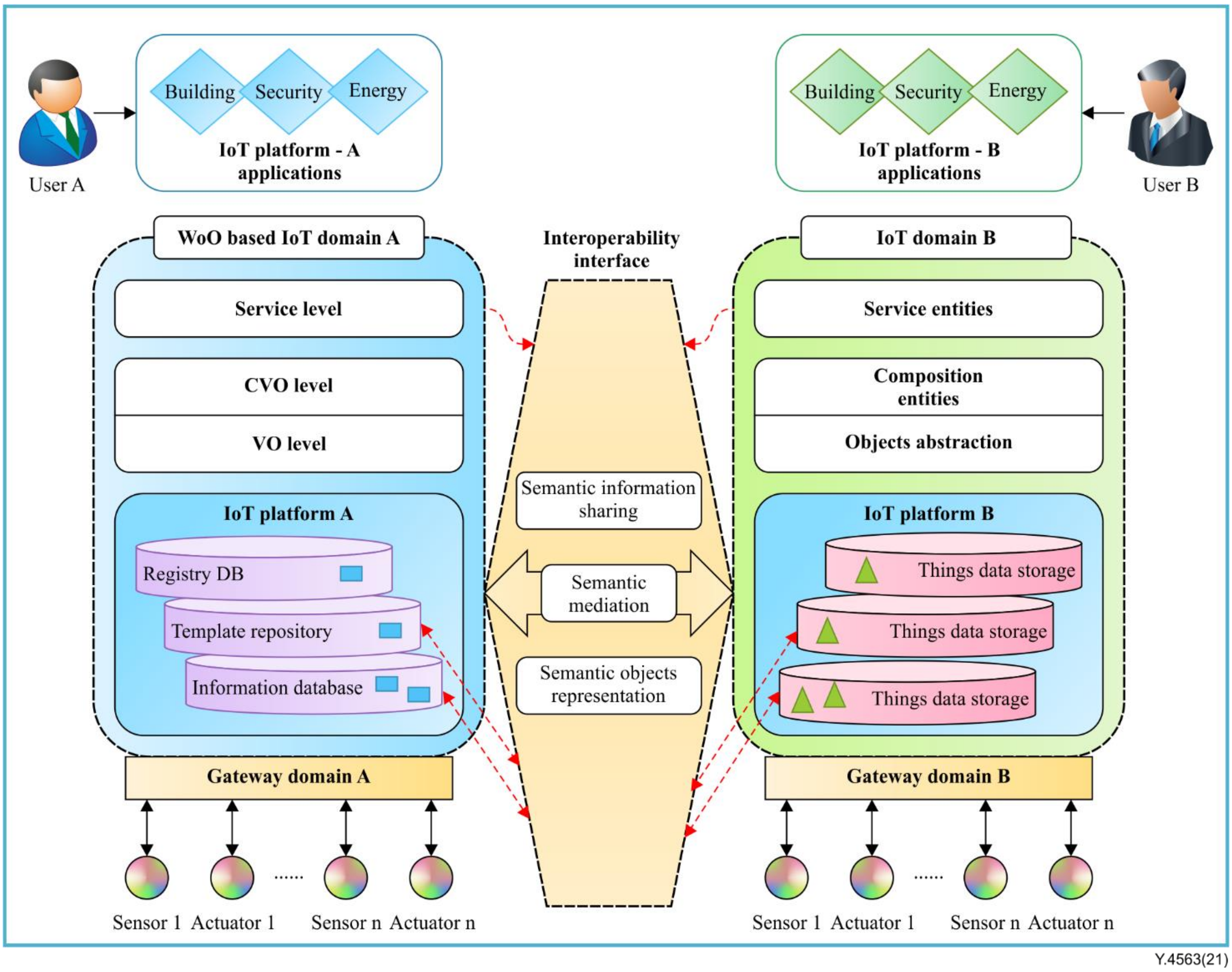

Figure A.1 – Interoperability with web of objects (WoO) based model

A.1 Web of objects based IoT domain

Figure A.1 describes the architecture of an interoperable WoO IoT environment. The WoO is based on the following functions as indicated in [ITU-T Y.4452].

– WoO service level supports functions to create and manage IoT service entities in [ITU-T Y.4452].

– Composite virtual object (CVO) sub-level is responsible for control and management of CVOs, with respect to the functional capabilities of the CVO sub-level based on the WoO reference model [ITU-T Y.4452].

– WoO virtual object (VO) sub-level provides the functional capabilities responsible for the control and management of VOs [ITU-T Y.4452].

– The registry DB, template repository, information database enables access to the WoO resources [ITU-T Y.4452].

Figure A.1 shows the interface which presents the interoperability interface between WoO based IoT platform and an external IoT platform.

A.2 Interoperability interface

It is required that the interoperability interface between WoO IoT (A) domain and external IoT domain (B) allow access to the IoT data and services that are available in any of the IoT domains.

It is required that this interface includes the data representation, translation, sharing mechanisms to enable applications for cross-domain data interoperability.

It is recommended that this interface enables access to the internal data of the domains and the capabilities for the extraction of IoT data.

This interface needs to provide the semantic interoperability that can provide the following functions.

– Semantic information sharing

  The semantic information sharing enables access to data and metadata of sensors registered in the domains.

– Semantic mediation

  Semantic mediation enables the translation of the data and the metadata of the information model defined by the domain.

– Semantic object representation

  Semantic object representation provides the common representation of the sensor observation for the information models defined by the domains.

A.3 External IoT platform

Figure A.1 describes the external IoT platform on the right side of the illustration.

The external IoT platform is based on the following functions as indicated in [ITU-T Y.4401].

– The service entities support the service functions based on the service provisioning capabilities indicated in clause 8.1 of [ITU-T Y.4401].

– The composition of entities in an external IoT platform will provide IoT object orchestration for higher level service workflows based on the application capabilities defined in [ITU-T Y.4401].

– Object abstraction and data handling functions in the external IoT platform provide capabilities that are responsible for the management of object handling based on the data management capabilities in [ITU-T Y.4401].

Annex B

Data interoperability of semantic and non-semantic data

(This annex forms an integral part of this Recommendation.)

Nowadays complex IoT systems such as in a city domain involves different types of data including semantic and non-semantic sources. They raise the opportunity to combine diverse sources of data and make them interoperable so that uniform information retrieval and knowledge extraction can be carried out. Not all IoT data holds semantics; extracting knowledge from such data requires pre-processing and transformation to make them semantically interoperable for high-level applications in order to achieve intelligent decision making based on interoperable data. Annex B delivers processing considerations to support interoperability among semantic and non-semantic data in an IoT environment.

B.1 Data processing for semantic and non-semantic data interoperability

It is important to consider data from heterogeneous sources to share and access information for IoT service provisioning. Both semantic and non-semantic data need to be considered in IoT environments as well as analyzing them for extracting knowledge for high level applications. To support semantic interoperability the following data processing and management characteristics must be considered:

– Consider both data from semantic and non-semantic sources;

– Transformation of semantic and non-semantic data (e.g., data from relational databases), integration and mapping to RDF triples (subject-predicate-object);

– For transformation, integration, and mapping, data requires schema that might not be available all the time or have a complex schema that is represented by ontological languages, such as the web ontology language (OWL);

– Consider semantic data sources, independent schema and focus on the meaningful relationship of data for data interoperability;

– SPARQL queries are made on the knowledge database to support data analysis and create application services. Besides supporting semantic data in RDF format, relational database, comma separated values (CSV) format, as well as other formats need to be supported.

The following figure shows an example of data collection, processing and transforming of data to make semantically interoperable IoT data available and provide this data to the IoT applications.

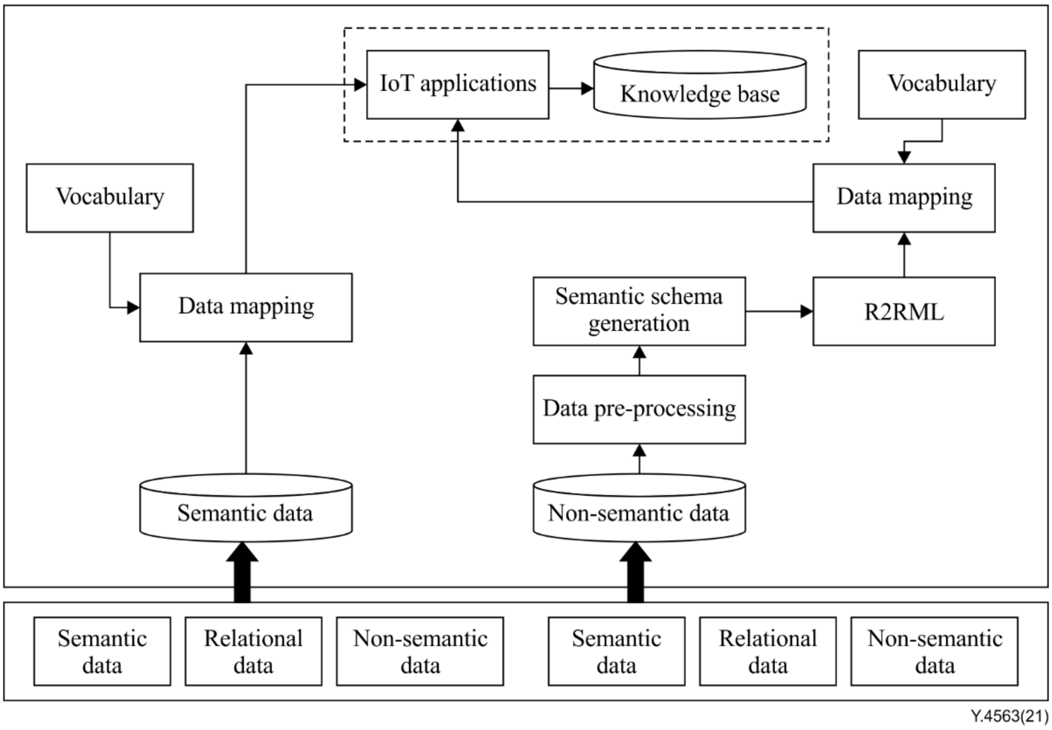

Figure B.1 – IoT data processing for semantic and non-semantic data interoperability

Enabling interoperability of semantic and non-semantic data includes different elements, few of the elements have been included below:

Semantic data source

– Represents a knowledge base;

– Allows the use of semantic ontology to express knowledge in the knowledge base;

– Data is represented in RDF format and stored in OWL format;

– Knowledge base is composed of domain terminology (concept definition) and their instances.

Non-semantic data source

– Represents a relational database;

– Source of the data includes distributed data, streaming data, text data, machine-generated data, sensed data, etc.;

– Format of the data includes XML, CSV, etc.

Schema mappings

– Specifies the relationship between two schemas;

– Mapping between source and target schema based on requirements;

– Similarity of concepts, properties and instances of source schema are compared with the concepts, properties and instances of target schema.

Semantic schema generation

– Semantic schema need to be generated from non-semantic schema;

– In the case of a relational database, or XML data, semantic schemas are constructed from the relational schema to build a semantic ontology;

– Schema from a relational database is the source schema and semantic schema is the target schema that has been defined by the user.

B.2 Transformation of a non-semantic to the semantic data schema

Semantic definition of domain terminology or semantic data schema is defined by domain experts or developers based on a common vocabulary. Thus, if the source data schema is already in the semantic format, it does not require any processing for mapping to the target schema. If it is a non-semantic data schema then it requires conversion into semantic data schema. This means non-semantic data schema or data from a relational database (e.g., RDB) is converted into semantic data schema (RDF graph). For conversion of non-semantic data schema (relational schema) into semantic data schema, RDB-to-RDF mapping language such as (R2RML) is used.

Appendix I

Ontologies and description models for IoT data interoperability

(This appendix does not form an integral part of this Recommendation.)

With respect to the semantic information model defined in clause 8.3, Appendix I provides available examples of the semantic sensor ontology model and the web of things information model. These information models can be used and extended to support data interoperability with respect to defining a common information model that can be used in IoT domains.

I.1 Semantic sensor ontology model (A joint W3C and OGC standard)

The main goal behind semantic sensor web is to use the web for linking sensors around different networks and make their observations available on the web. To model sensors, semantic sensor network (SSN) ontology [b-Compton], [b-Haller] defines a set of domain independent concepts. SSN is defined by the W3C consortium. The SSN ontology is conceptually divided into ten modules as shown in Figure I.1. The ontology contains 41 concepts and 39 object properties. The ontology describes the sensors and the capabilities of those sensors. It also describes the observations and techniques used for sensing. The notion of operating ranges and survival ranges of sensors and their performance in those ranges are also included. To represent deployment lifetime and sensing purpose, structures are also included. Moreover, external concepts are included to allow linking of to external ontologies such as the observations of a particular property of a feature. In this case, the observations are described but the feature and property are left as vacant placeholders.

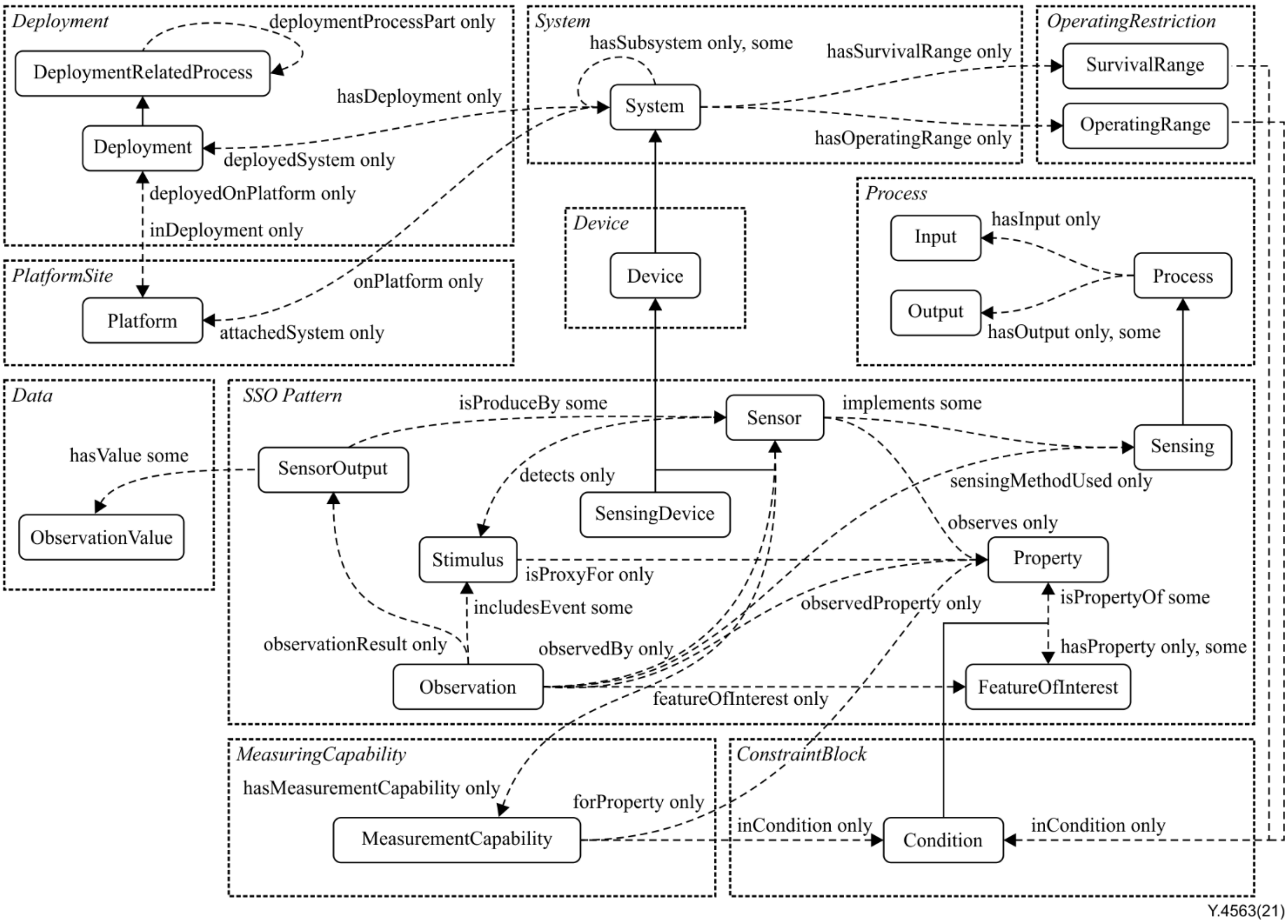

Figure I.1 – The SSN ontology, key concepts and relations [b-Compton]

I.2 Web of things (WoT) description model

The W3C web of things (WoT) [b-Kaebisch] enables interoperability across IoT platforms and application domains. It provisions mechanisms to describe IoT interfaces to enable IoT devices and services to communicate with each other, independent of their underlying implementation, and across multiple networking protocols. It also provisions a standardized way to define the behaviour of IoT programmes.

WoT thing description

The WoT thing description (TD) is the structured data that fills the gap between linked data vocabularies and APIs of IoT systems. A TD provides general metadata of a thing, also interactions metadata, data model, communication, and security mechanisms of a thing. TDs uses domain-specific metadata for WoT. JSON-LD is used to serialize thing descriptions. JSON-LD provides a good option for machine-understandable semantics as well as the usability for developers. Thing descriptions are maintained in thing directories which provide an interface for registration and removal. A web-based interface for lookups is also provided by thing directories which includes a SPARQL endpoint for semantic queries. The WoT thing description provides interoperability in two ways. Firstly, TDs enable machine-to-machine communication in the web of things. Secondly, TDs can provide a common format for developers to recommend and retrieve the information required to access IoT devices and their data.

The following figure provides a view of how a WoT description can be used in a WoT architecture together with building blocks.

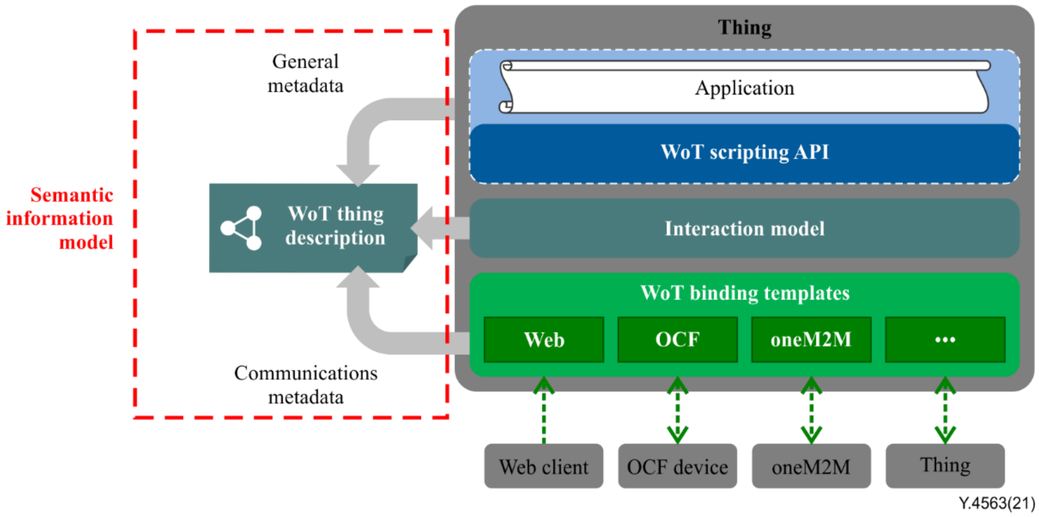

Figure I.2 – The model of thing description with WoT architecture [b-Lagally]

I.3 Hierarchical classification of things

Taxonomy

Taxonomy is a hierarchical classification of things in which a content object can be placed in one node. In contrast, a metadata is a flat model of information associated with the content.

In data interoperability, taxonomy is required for:

– Aggregating data;
– Classification of information;
– Filtering of data;
– Searching the data;
– Knowledge management;

– Relationship among tables.

Vocabularies

For a common understanding of data in IoT environments, developing ontology requires commonly used terminologies that can be achieved by following a vocabulary.

Several vocabularies have been based developed by several organizations that cover diverse domains. Different organizations may build their own data model based on a vocabulary that is independent and application-specific. But these vocabularies are not linked, implemented and maintained as separate entities at a national and local level.

To share a common understanding and unambiguous meaning of data, and to support data interoperability, the following are possible solutions that need to be considered:

– A common specification or data model to build a vocabulary;
– If a new vocabulary is developed it should follow a common specification;
– Existing vocabulary needs to be updated following the specification;
– An association of vocabularies that links the standard vocabularies.

A vocabulary includes detailed information about the concepts that uniquely identify and describe all the fundamental elements in the data model. The vocabulary does not depend on the data model, rather a data model is built according to a standard vocabulary.

A standard vocabulary has the following characteristics:

– A vocabulary contains different terminologies from different sources;
– Each terminology in the vocabulary is defined as a concept;
– Each entity in the vocabulary is identified by unique identifiers (e.g., uniform resource identifier (URI) for concept) to link the concept;
– Relationships among the concepts are maintained for the semantics of the concepts (e.g., the meaning of the same concept varies depending on the context);
– For naming a concept, meaningful synonyms should be included for the meaning in different contexts among multiple domains;
– It should include concepts as standard concepts that will describe the information regarding the data model, source of the model, creator of the data mode, organization, etc.

An entity-relationship model for a concept in a vocabulary has been shown in Figure I.3.

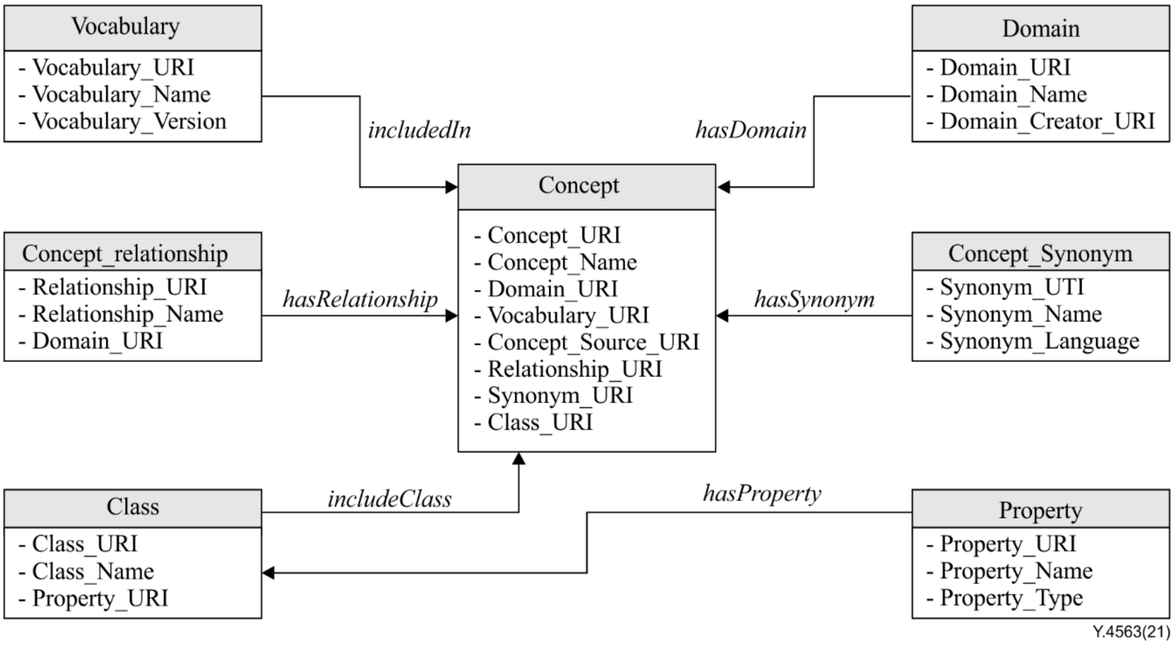

Figure I.3 – A model of a vocabulary concept

Linked open data

– **Linked open data**: Linked open data is a semantic framework that establishes the links among different data sets. The data sets can be geographically scattered or bound to organizational territories. Moreover, a linked data infrastructure can be established between diverse databases located in different data centres. Linked data provides a way to publish data in a machine-readable and human understandable format on the internet. In linked data design hypertext transfer protocol (HTTP) and URIs are used to expose and publish the data on the web. W3C recommended semantic representations format (RDF) is used to organize the data as linked open data [b-W3C, b-Wikipedia].

– **Linked open vocabularies (LOV)**: Linked open vocabularies (LOV) are RDF-based repositories in which data publishers and consumers can lookup vocabularies [b-Vandenbussche, b-LOV]. These vocabularies can be used to annotate and parse messages. Many works for defining domain-dependent and independent vocabularies have been performed, such as schema.org1. Other initiatives are similar to MobiVoc2, which is a domain-dependent RDF vocabulary for mobility. Moreover, semantic sensor network ontology (SSN) provides high level semantic specifications of things and their relations.

Appendix II

Provision of interoperability among applications across buildings – Project Haystack approach

(This appendix does not form an integral part of this Recommendation.)

There have been some initiatives that cater to interoperability in different domains. One of these is the project Haystack [b-Haystack] which provides interoperability among applications across buildings.

Data models and formats in Haystack project

The project is an open source initiative to promote the interoperability of applications [b-Haystack]. It describes data models, data formats and data structures to share data on HTTP with the REST APIs. Haystack project introduces concepts like TagModel, Structure, TimeZones and Units. The TagModel is a meta-model that uses tags. The tags are name/value pairs that are associated with an entity. As tags are simple and dynamic, they provide a flexible way to build standardized models and are customizable according to any project or equipment. Further, the tagging models are integrated and layered above existing models such as object-oriented classes or relational schemas. While the notion of entity is the real-world object abstraction in a haystack. The entity may include sensors, weather stations, equipments, buildings, etc. From a software systems point of view an entity can be modelled as a record entry in a database or an object.

Moreover, we describe some of the key concepts associated with interoperable data formats introduced in the Haystack project below.

Tags:

It is a pair of names/values which are associated with an entity. It describes any facts or attributes about an entity. If we associate a *site* tag with an entity then it is considered as a building. If we include *geoAddr* then we declare the street address for that building. An example of an entity describing a site is presented below which includes seven tags including id, site, dis, area, geoAddr, tz and weather.

```
id: @whitehouse
dis: "White House"
site
area: 55000ft²
geoAddr: "1600 Pennsylvania Avenue NW,  Washington, DC"
tz: "New_York"
weatherRef: @weather.washington
```

Figure II.1 – Example for an entity describing a site with corresponding tags [b-Haystack]

Grids:

– A tabular two-dimensional data structure is used to serialize tagged data. Tabular representation of tagged entities in a grid is serialized over HTTP using the Rest API. The grid constitutes the metadata at grid level;

– Columns of a programmatic name and metadata;

–	Rows are defined as scalar cells.

An example of a grid representing three entities all having id, dis and site tags. The entities when combined in a grid, forms five columns and three rows:

id	dis	site	area	phone
@site-a	"Site A"	✓	45000ft²	
@site-b	"Site B"	✓		
@site-c	"Site C"	✓	62000ft²	"(804) 555-1234"

Figure II.2 – Example for a grid [b-Haystack]

Haystack project provides data formats for easy exchange of data. It introduces some more common data structures and text formats as below:

–	Filters: A predicate language used for querying tagged data;

–	Zinc: A format for exchanging plain text data;

–	JSON: JavaScript object notation;

–	CSV: encoding data with comma separated values.

Appendix III

Use case of data interoperability provisioning in web of objects (WoO) based IoT environments

(This appendix does not form an integral part of this Recommendation.)

III.1 VO information model in WoO based IoT environment

To support data interoperability, an information model is required for the semantic meaning of data. A VO information model is a general model that is used to virtualize real world objects in the virtual world.

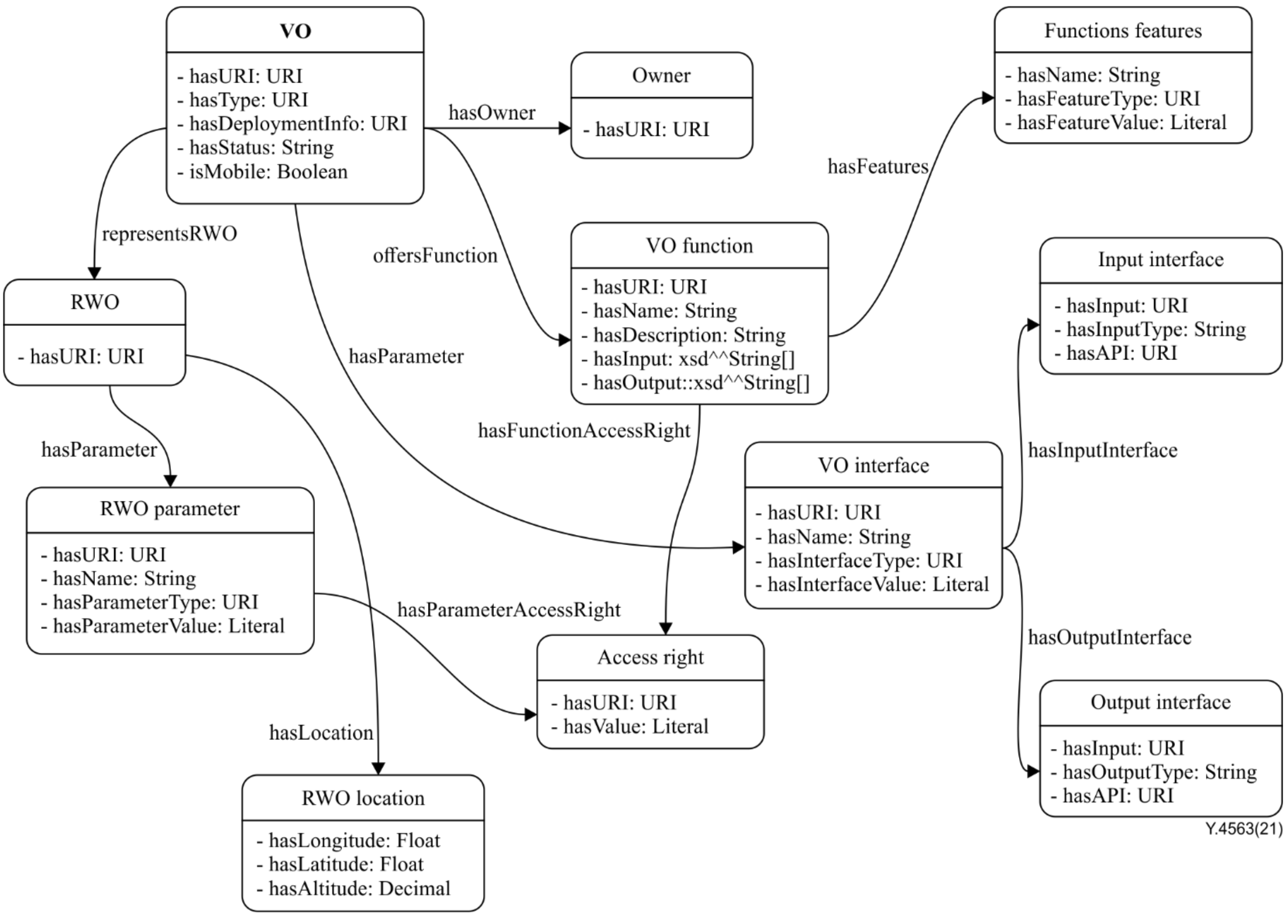

Figure III.1 – VO information model

Features of a VO information model:

– Provides interrelationship among the resources and attributes, IoT service entities and interfaces to the real world;
– Enables a real world object to be virtualized to form a VO;
– Through the object virtualization, the real world objects can be accessed internally and externally;
– Provides interfacing of the real world object to the external world;
– Includes a set of metadata so that an object can be represented semantically.

III.2 CVO information model in WoO based IoT environment

To support data interoperability and reusability, a composite virtual object (CVO) information model allows service publicizing, reusability and interoperability among heterogeneous application domains.

A CVO information model is a general model that is used to develop domain specific CVO to offer service relevant functionalities. Figure III.2 shows a CVO information model.

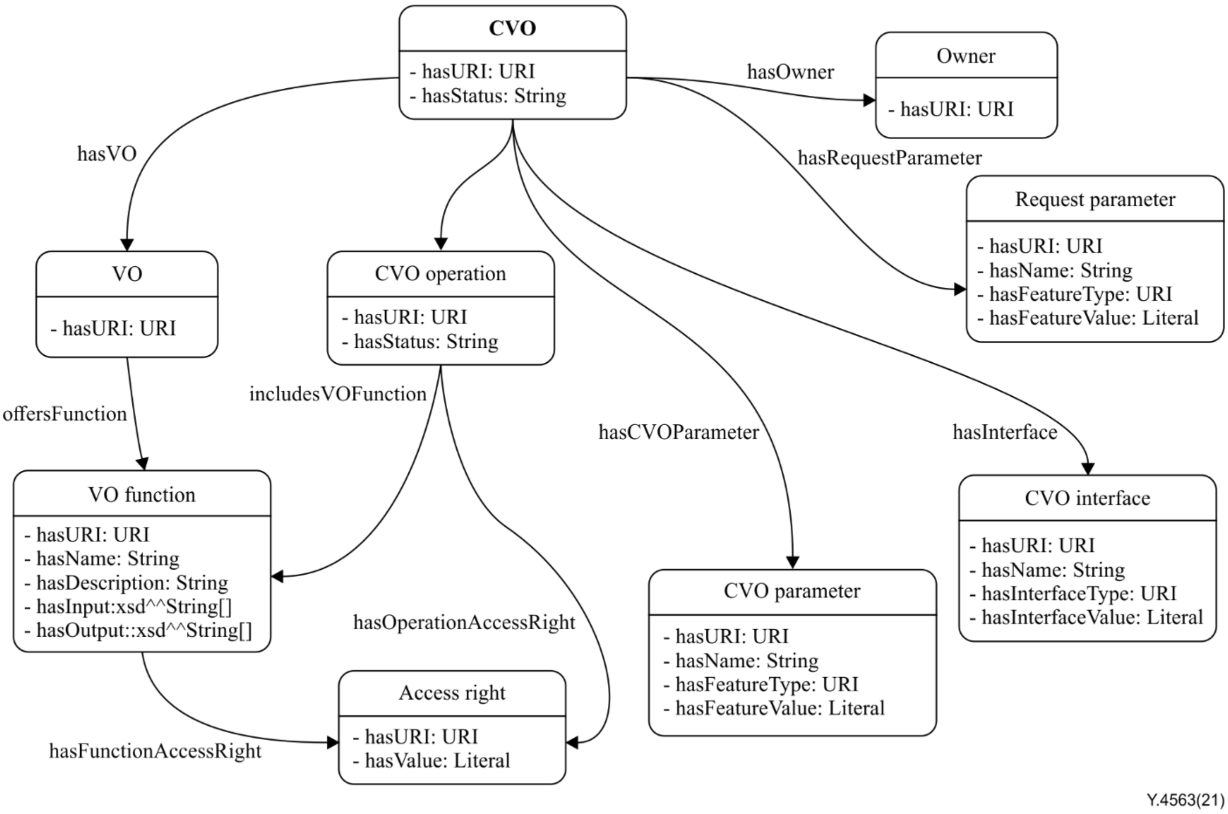

Figure III.2 – CVO information model

Features of a VO information model:

– Allows interrelationship among service relevant VOs, VO functions, CVO operation, parameters, etc.;

– Includes a set of metadata that describes the attributes for unambiguous representation of the attributes.

III.3 Semantic ontology for data interoperability

For data interoperability, linking, storing, querying and searching data in a machine readable and automated manner requires the use of semantic ontology that should be registered and maintained by the semantic ontology registry (SOR). W3C standards on semantic ontology and ISO/IEC metadata standards have been adopted for the SOR.

To allow data interoperability in IoT environments, the semantic ontologies require to be interlinked to each other so that they can enable sharing and reusing of existing ontologies across the application domains that are enabled by the SOR framework.

III.3.1 SOR framework

To support data interoperability a semantic ontology registry (SOR) framework has been proposed that enables the following functionalities:

– Registration of multiple ontologies in the SOR cloud;
– Each ontology will be assigned with a unique identifier using URI;
– Searching of data elements in the ontology that are maintained in the SOR;
– Extraction of the description of the data element from the SOR;
– Reusing of the data element by linking the respective data element;

To support data interoperability efficiently across the IoT environments, the following additional functionalities are supported by the SOR:

– Linking and associating the data element across different SORs according to well-defined terminologies from standard vocabularies;
– Query the relationship of the data element across the SORs;
– Linked data principles have been applied for the semantically linked SOR framework. Linked data allows URIs and RDFs for linking and describing the data elements for sharing and linking objects. Linked data exposes the data elements in the ontology maintained by the SOR so that the data elements can be interlinked with other data elements in different ontologies in the SOR cloud.

III.3.2 Mapping of data element to the terminology server (vocabulary)

To support data interoperability, for common understanding and unambiguous meaning of data, it is important to follow common metadata and terminologies that have been described in the vocabulary. There are several terminology servers that follow the linked data principle.

Requirements for mapping data elements in SOR with the terminology servers include:

– Each of the data elements in the SOR needs to be annotated and linked with an external terminology server using a URI in the linked data cloud;
– To interrelate data elements in the SOR cloud with the terminology server, some components are used to link between them that includes:
  • Object classification (OC) that identify the object of the data element;
  • Object classification item (OCI) that includes a description of the vocabulary, specific object in the vocabulary and URI of the object.
– Name of the vocabulary if the source from where the object is imported;
– URI is used to identify the object in the vocabulary that is inserted as a value;
– Simple knowledge organization system (SKOS) is used to interrelate one data element in one SOR with another data element in a different SOR;
– OCI is used to make a relationship between data elements in different SORs;
– Source data element is identified by source URI, and the target data element is identified by the target URI.

This mechanism of linking the data element with the terminology server allows the data element across the SOR in the cloud to use the same terminology, same descriptions with the same resources to provide the same meaning of the data element and the value of the data element across the SORs. The OC and OCI have been adopted from ISO/IEC 11179 classification scheme and classification scheme item.

Figure III.3 shows the linking of data elements with the terminology server.

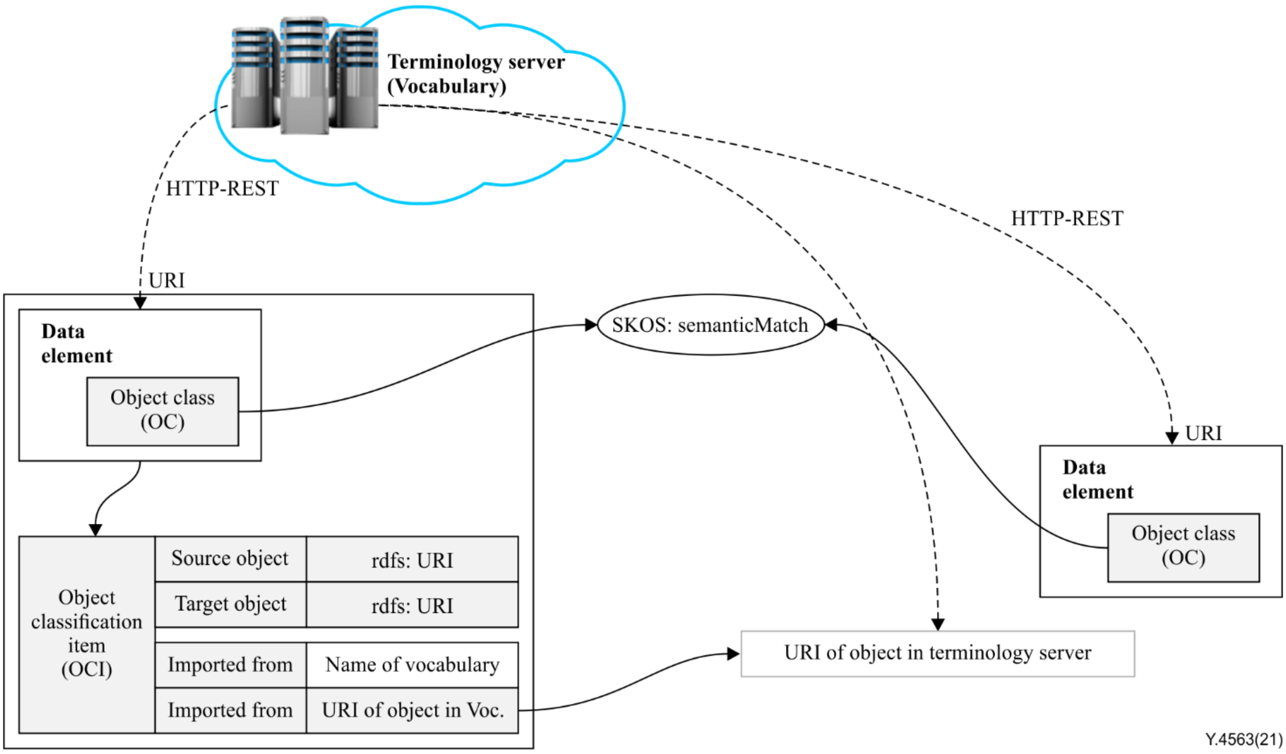

Figure III.3 – Mapping of data element to the terminology server for common vocabulary to support data interoperability

III.3.3 Utilization of linked SORs for data interoperability

In data interoperability, data needs to be collected and analysed based on a domain-specific format, but it should be automatic to manage huge amounts of data from the large number of data sources. SOR enables the system to populate the data automatically coming from heterogeneous sources based on the specification of the data model using SOR query services. Hence, searching and linking data elements in the SOR is necessary.

The following steps describe the process to search and link data elements in the SOR:

– Application server searches the local SOR to retrieve data elements along with their descriptions for the selected set of data element. Local SOR returns a list of data elements descriptions, including their URIs of matching data elements maintained by the SOR;
– Semantic ontology manager responds to the application server with the matching set of data elements according to the requirements of the ontology model;
– Application server searches the similar terminology based on the extraction specification of the data elements;
– SPARQL query manager is used to query the SOR in the cloud for the specifications of data elements. It searches specifications of each registered data element in the SOR cloud;
– Based on the query request, the URI of the specification of a data element is searched in the registered SORs;
– SOR in the cloud responds with the URI of the specifications of the data element to the query manager, and then the manager responds to the application server with the URI of specifications.

Data interoperability provisioning in WoO based IoT environments supports:

– To aggregate, process and manage data;
– To follow semantic data model to describe the concepts and their relationship;
– To use semantic schema for describing the relationship among the concepts;

- To virtualize real world objects to semantic ontology;
- To format the XML/RDF format to express virtual objects that are stored in an OWL format;
- Hierarchy of classes and sub-classes in semantic ontology;
- To map classes in one ontology with external classes in different ontologies for reusing data and information;
- To allow linked data concept for linking other ontologies to enhance knowledge and capabilities;
- Semantic annotations to describe the classes, properties and specifications of a domain;
- To allow the use of common terminology in the vocabulary to develop ontology.

Appendix IV

Common data model (CDM) to transform into a common model

(This appendix does not form an integral part of this Recommendation.)

A common data model (CDM) is a set of data elements that can be mapped among different interoperating systems. A commonly agreed common data model is necessary so that a heterogeneous interoperating system can interact and understand based on a CDM. A CDM is the core of a data service that is populated with different data elements and useful in multiple application domains. Features of CDM include:

– A CDM is scalable that allows adding new data elements based on the requirements;

– A CDM allows the transformation of data into a single common data format from a different format that is collected from heterogeneous sources. For transformation into a common format or CDM, common terminologies from standard vocabularies need to be followed.

A conceptual model of a CDM has been shown in Figure IV.1.

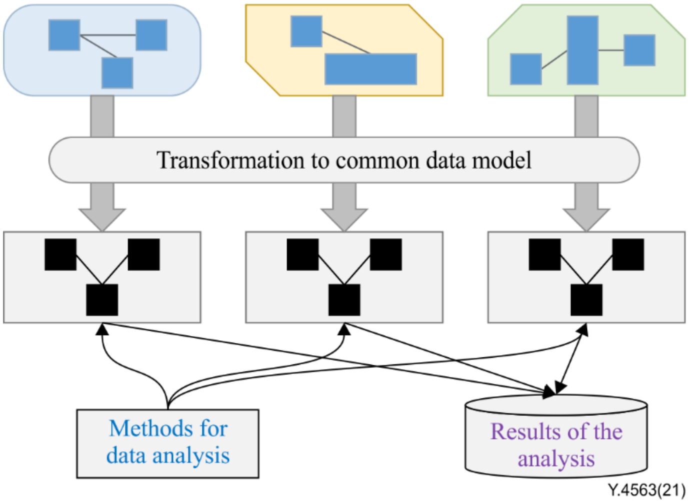

Figure IV.1 – An example of a conceptual model of a CDM

Conventions, features and characteristics of a CDM are listed below:

– CDM consists of a set of commonly used data elements;

– Terminology for data elements used in the CDM should be taken from standardized terminology;

– Should have the capacity to reuse data elements;

– Due to increased number of datasets, the CDM should be scalable so that it can support an increasing number of datasets for data processing and management;

– CDM should be platform independent;

– One convention should be followed across different tables for naming a variable.

CDM plays an important role in improving the common understanding of data elements, quality of data, and for sharing among multiple application domains. CDM has been designed and applied to control and exchange data in the domains of food, drug, clinical healthcare, and different medical applications. Depending on the types of data elements and data, different types of CDM are required that are listed here:

– A universal CDM is used for study purposes and not for a specific interest (e.g., subjects in health study) that require extraction of information from multiple sources regardless of boundaries or countries (e.g., the disease may be common to specific countries around the world) for general outcomes;

– A domain-specific CDM is used for specific domain or studies in specific topics, such as health, medicine, body system and also for classification;

– A required CDM is requested for an institutional policy or for keeping a record for the institution;

– Domain specific CDM focusing on a specific study.

Bibliography

[b-Bröring] Bröring, Arne, et al. "Enabling IoT ecosystems through platform interoperability." IEEE software 34.1 (2017): 54-61.

[b-Compton] Compton, M., Barnaghi, P., et al. (2012), The SSN ontology of the W3C semantic sensor network incubator group. *Journal of Web Semantics*, Volume 17, December, pp 25-32. <https://doi.org/10.1016/j.websem.2012.05.003>

[b-Euzenat] Jerome Euzenat & Pavel Shvaiko. Ontology matching. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[b-Haller] Haller, Armin., Janowicz, K., et al. (2018), The SOSA/SSN ontology: a joint W3C and OGC standard specifying the semantics of sensors, observations, actuation, and sampling. *Semantic Web X IOS Press*. < http://www.semantic-web-journal.net/system/files/swj1804.pdf>

[b-Haystack] Project Haystack. (2014), An open source initiative to streamline working with IoT data.

[b-Hoebeke] Hoebeke J, De Poorter E, Bouckaert S, Moerman I, Demeester P (2011) Managed ecosystems of networked objects. Wirel. Pers. Commun. 58(1):125-143.

[b-Ishaq] Ishaq I, Hoebeke J, Moerman I, Demeester P (2012) Internet of things virtual networks: bringing network virtualization to constrained devices. 2012 IEEE Int Conf Cyber Phys Soc Comput.

[b-Kaebisch] Kaebisch, S., Kamiya, T., et al. (2020), Web of Things (WoT) Thing Description, *WRC Recommendation*. <https://www.w3.org/TR/wot-thing-description/>

[b-Kiljander] Kiljander, Jussi, et al. "Semantic interoperability architecture for pervasive computing and internet of things."IEEE access 2 (2014): 856-873.

[b-Lagally] Lagally, M., Matsukura, R., et al. (2021), Web of Things (WoT) Architecture 1.1, *WRC Editor's Draft*. <https://w3c.github.io/wot-architecture/#sec-building-blocks>

[b-LOV] Linked Open Vocabularies (2016), *Semantic Web Journal*. < https://lov.linkeddata.es/dataset/lov/>

[b-Vandenbussche] Vandenbussche, P. Y., Atemezing, G. A., et al. (2014), Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web, *IOS Press Semantic Web Journal*. <http://www.semantic-web-journal.net/system/files/swj1178.pdf>

[b-W3C] W3C Semantic Web (2015), Linked Data. <https://www.w3.org/standards/semanticweb/data>

[b-Wikipedia] Wikipedia, The Free Encyclopedia. <https://en.wikipedia.org/wiki/Triplestore>

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	**Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities**
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2022