International Telecommunication Union

ITU-T Technical Specification

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(7 April 2019)

ITU-T Focus Group on Data Processing and Management to support IoT and Smart Cities & Communities

---

Technical Specification D3.3

Framework to support data interoperability in IoT environments

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 procedures for establishment of focus groups are defined in Recommendation ITU-T A.7. ITU-T Study Group 20 set up the ITU-T Focus Group on Data Processing and Management to support IoT and Smart Cities & Communities (FG-DPM) at its meeting in March 2017. ITU-T Study Group 20 is the parent group of FG-DPM.

Deliverables of focus groups can take the form of technical reports, specifications, etc., and aim to provide material for consideration by the parent group in its standardization activities. Deliverables of focus groups are not ITU-T Recommendations.

Technical Specification D3.3

Framework to support data interoperability in IoT environments

Summary

This Technical Specification specifies a framework to support data interoperability in IoT environments. The relevant requirements and technologies that support the data interoperability are defined in this Technical Specification.

Acknowledgements

This Technical Specification was researched and principally authored by Ilyoung Chong (Hankuk University of Foreign Studies) and Sajjad Ali (Hankuk University of Foreign Studies).

Additional information and materials relating to this Technical Specification can be found at: www.itu.int/go/tfgdpm. If you would like to provide any additional information, please contact Denis Andreev at tsbfgdpm@itu.int.

Keywords

Data interoperability; semantic web; Internet of Things (IoT), ontology.

Technical Specification D3.3

Framework to support data interoperability in IoT environments

Table of Contents

Technical Specification D3.3

Framework to support data interoperability in IoT environments

1 Scope

This Technical Specification addresses a framework for data interoperability. The scope of this Technical Specification covers several key requirements with respect to data interoperability in IoT environments and many important elements to fulfil these requirements. Specifically, it covers the following:

- Overview of data interoperability in IoT environments;
- Requirements to support data interoperability;
- Functional model to support data interoperability;
- Details on the semantic mediation functions;
- Details on the syntactic mediation functions;
- Details on the interoperable object abstractions functions;
- Details on the interoperable data repositories;

NOTE – For Data interoperability example through semantic mediation see Annex A and Appendix III.

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

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Technical Specification. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Technical Specification 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 Technical Specification does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.4111/Y.2076]	ITU-T Recommendation Y.4111/Y.2076 (02/2016), *Semantics based requirements and framework of the Internet of things.*
[ITU-T Y.4452]	ITU-T Recommendation ITU-T Y.4452 (09/2016), *Functional framework of Web of Objects.*

3 Definitions

3.1 Terms defined elsewhere

This Technical Specification uses the following terms 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. It provides a service architectural model to support simple development, deployment and operation of IoT services on the Web (for WoO example see Appendix III).

3.2 Terms defined in this Technical Specification

This Technical Specification defines the following terms:

3.2.1 Common data model: A common data model is a data model that allows transformation of data into a single common data format from different format that are collected from heterogeneous sources. For transformation into a common format or 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 semantic data model is a conceptual model that includes the semantic information of instances. Semantic information of data defines the meaning of 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 Technical Specification uses the following abbreviations and acronyms:

CDE	Common data element
CDM	Common data model
HTML	Hyper Text Markup Language
IoT	Internet of Things
JSON	JavaScript Object Notation
RDF	Resource Description Framework
REST	Representational State Transfer
URI	Uniform Resource Identifier
XML	Extensible Markup Language

5. Conventions

In this technical specification:

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

6. Data interoperability in IoT environments

6.1 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, the most challenging issue is to handle the interoperability of the information and services. Data interoperability is the ability of applications to exchange data between IoT environments and the use of 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, integration of data.

From the definition of data interoperability, two important concepts can be identified as follows:

- Exchange of data between interoperating IoT environments – implies how the data will be exchanged – this is a syntactic issue;
- Use of exchanged data semantically – implies how the exchanged data can be used in a meaningful and unambiguous way – this is a semantic issue.

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 V). Several types of interoperability can be considered to fully realize data interoperability in IoT platforms. These include semantic interoperability, syntactic interoperability, and organizational interoperability. In order to achieve a high level organizational interoperability, semantic and syntactical interoperability are necessary to be achieved. Figure 6-1 illustrates a view of multiplatform data interoperability and the interaction of core components. Achieving interoperability from all of these views results in organization level interoperability. The data representation in different formats and information models complicates its usability for IoT applications. This scenario 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. However to fulfill an integrated sensor data exchange and provide seamless data interoperability several mechanisms are required at different layers of IoT infrastructure. We define three types of functions to realize data interoperability among IoT platforms, namely, the semantic mediation functions, syntactic mediation functions and interoperable object abstraction functions. The details on these function is further elaborated in section 7.

From the data interoperability requirement point of view there are several requirements to be achieved for IoT environments. Such as the representation of IoT data is crucial. Modeling information is required to support not only the flexibility but also consider future needs such as technology evolution and mobility. Data Integration is one of the major aspects in interoperability, well-defined mechanisms for linking data such as the automated linking of relevant data sources and the linking of data sources which enables application integration and reuse of data are highly required. As well as mechanisms for the sharing and reusing of data would be necessary to support the interoperability. The search and discovery of IoT resources is another mechanism to support cross-domain access of resources. Moreover, the trust and security mechanisms are necessary for the interoperability of data as each domain has its own trust and security protocols, either the translation of these protocols in common substitutes are required to form seamless accessibility.

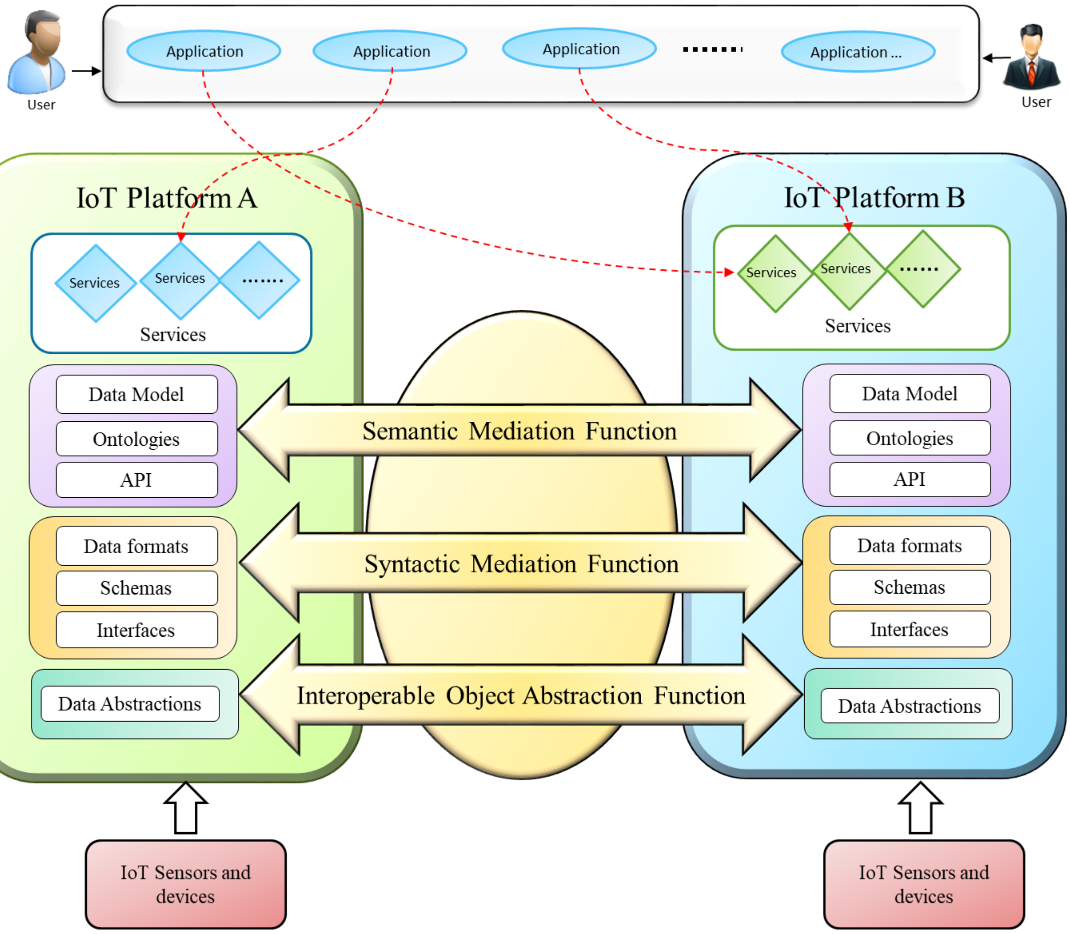


Figure 6-1. A view of data interoperability in heterogeneous IoT environments

6.2 Dimensions of data interoperability

- Syntactical Interoperability

Heterogeneous IoT devices generate data that are stored and used in different formats. Syntactical Interoperability is concerned with the data formats, syntax and coding methods. Protocols used by IoT devices use standard syntax for communication of data. These are expressed in diverse formats such as XML, JSON or HTML.

- Semantic Interoperability

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

- Object Abstraction Interoperability

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

7. Requirements to support data interoperability

To support data interoperability, many aspects are required to be considered. Following sections highlight some general requirements to support data interoperability and some requirements with respect to semantic interoperability.

7.1 General requirements for data interoperability

To ensure the appropriate level of data interoperability following general aspects are required.

- For the provision of data interoperability, use of standard vocabularies is required in order to provide a common understanding of data.
- Validation mechanisms to evaluate data translation and conversion process are required.
- Data interoperability mechanisms need to support integration and sharing of data for services among processes of same and different organizations.
- The data interoperability aspects related with the classification and aggregation of data using diverse taxonomies are required.
- To support security and privacy, additional data interoperability aspects need to be considered including interoperability of data within security schemes, as well data interoperability aspects in privacy protocols.

7.2 Requirements for semantic data interoperability

To ensure the appropriate level of semantic data interoperability following aspects are required:

- 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 are required. The linking mechanism are 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 diverse IoT data.
- Semantic data management: The abstract semantic representations of IoT data are required to be managed through a management services. A suit of well-defined services are required to manage the data allowing its access, retrieval and storage operations.
- Semantic Ontology alignment and mapping: There is a requirement to provide improved ontology alignment in order 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. 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 should also scale well in evolving IoT technology. 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 case of domain specific model, it is necessary to provide transformation service among heterogeneous semantic data models.

7.3 Requirements for syntactic data interoperability

To enable the data interoperability at syntactical level among heterogeneous IoT systems some requirement are required to be achieved which are elaborated as follows.

- 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 object 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 meta data schema and their mapping mechanisms
- Verification methods for format translation and conversion process to validate the effectiveness of translation mechanism.

7.4 Requirements for object abstraction interoperability

The object abstraction mechanisms are required to be defined in order to provide interoperable sharing of data in heterogeneous IoT environment. Several requirements are crucial to be considered which as characterized as follows.

- 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 which 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 functional model of data interoperability provides the mechanisms to support interoperability among different domain. The functional design of this model is shown in figure 8-1, each components of the model provides an individual functionality which are further described as below:

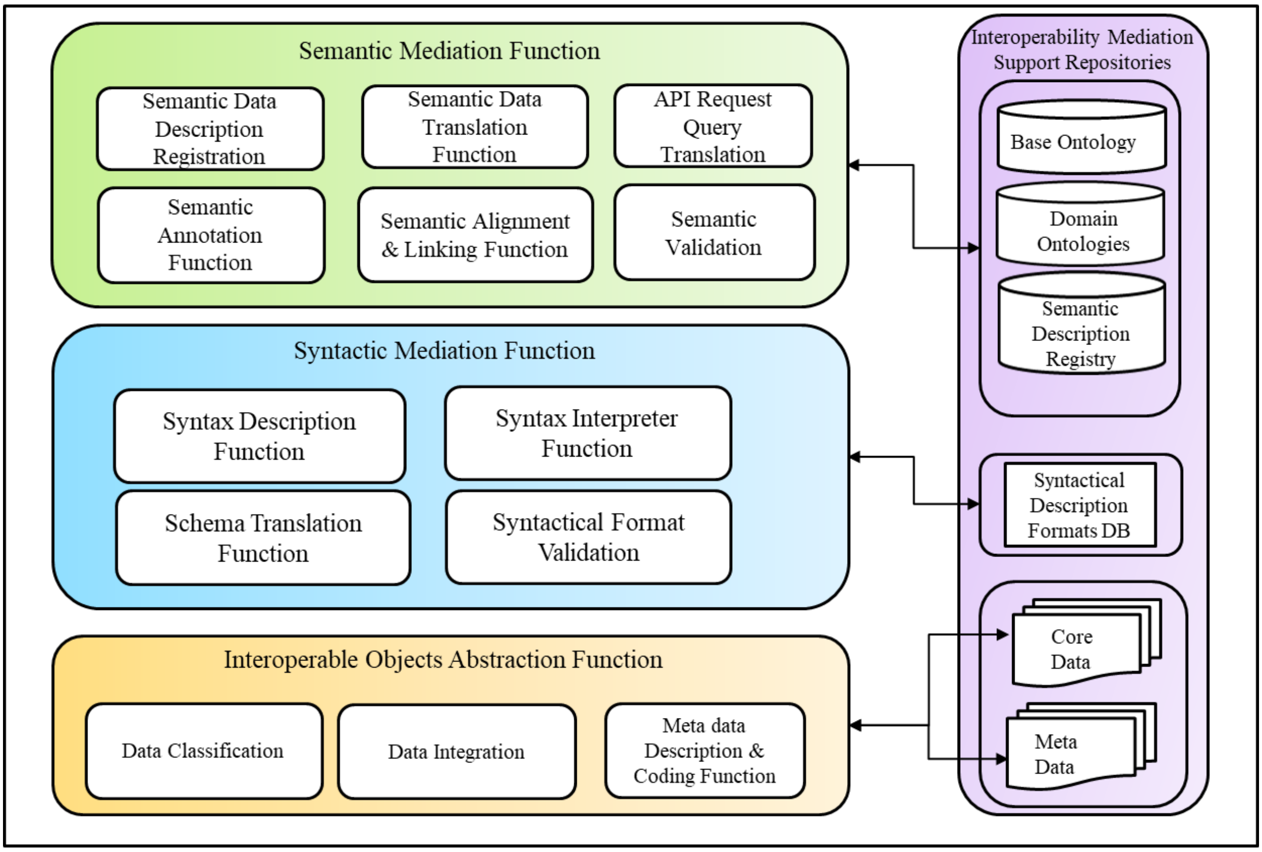


Figure 8-1 Functional mediation model to support data interoperability

- Semantic mediation function: This function offers the mechanisms for semantic translation and linking of the data from different platforms (see clause 7.1).
- Syntactic mediation functions: The function provides a syntax level mediation of different data formats, schemas and interfaces (see clause 7.2).
- Interoperable object abstraction function: The function consists of the mechanisms that provide the heterogeneous objects data abstractions, classification and categorization capability (see clause 7.3).

8.1 Semantic mediation functions

- Semantic data description registration: The Semantic Data Description Registration (SDDR) 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 semantic translation function.
- Semantic data translation function: The semantic translation function enables the translation of data formats to semantic formats that have been registered by SDDR. The translation of data to semantic formats is achieved through the defined domain ontology.
- API request query translation: The function provides interface to receive requests for the data to be translated to particular registered format. The request for data cannot be directly provisioned as different platform have diverse formats. The API Request translation function provides a common way to translate the requests to specific API format, in order to provision seamless interoperability.
- Semantic annotation function: This 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. The semantic annotation provisions data interoperability through specifying the particular function of IoT resources, their information and their operations which can be understandable by other services.
- Semantic alignment and linking function: The semantic alignments provisions an alignment strategy based on the defined alignment algorithm. The goal here is to align the information model or the semantic schema with another schema build to represent the same information but with different semantics. The Semantic alignment and linking function enables to resolve semantic heterogeneity in system where models have different meanings or ontologies.
- Semantic validation function: The function 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 validation test case defined on the bases of semantic ontology.

8.2 Syntactic mediation functions

- Syntax description function: The syntax description function provides a registration of syntaxes for the platforms. Using this function platform level syntax description formats are registered in the registry database. This function enable a useful records of syntactical formats for the platforms which enables syntax level interoperability.
- Syntax interpreter function: This function 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: This function enables the translation mechanism at schema level. Different platforms make use of different schemas to describe the data. The function provides the conversion to interoperate the schemas at platform level.
- Syntactical validation function: The function 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 Interoperable object abstraction functions

The (IOAF) provides the functional capabilities to support diverse object abstractions. The sub functions of IOAF are described as follows.

- Data Classification function: The Function provides capability for the classification and categorization of data. It enables the classified representation to be understandable through abstract representation model.
- Data Integration function: This function enables mechanisms for the extraction and integration of data. As data can be from different sources, the function provides procedures to integrate the data in standard formats such as a common data model.
- Metadata descriptions and coding function: This function allows the Metadata to be assigned to the data converted to the representation format. It enables to assign codes and other meta-data description to the core data in order to identify its sources.

8.4 Interoperable mediation support repositories

The interoperable mediation support repositories provide the capabilities to support storage and access for the translation and caching in the interoperability mechanisms. It consists of several components including:

- Base ontology model to define core model for semantic interoperability of platforms.
- Domain ontology models are the domain specific models of the platforms.
- Semantic registry to register the description of semantic data.
- Syntactic description DB is the storage repository to facilitate persistence of different data formats.

9. Details on the semantic mediation functions

This chapter provides the details on semantic mediation functions in accordance with the functional model of data interoperability (as in chapter 8).

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 semantic translation function.

The functional capabilities of Semantic data description registration 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.

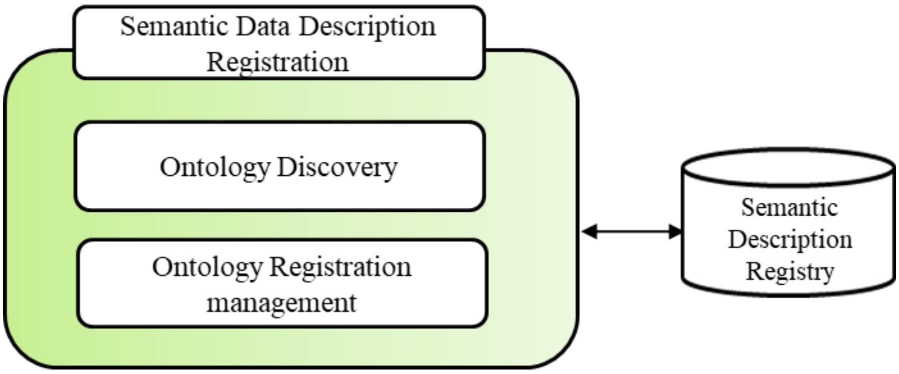


Figure 9-1. Semantic Data description registration functional components

9.2 Semantic data translation function

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

The functional capabilities of semantic data translation 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;

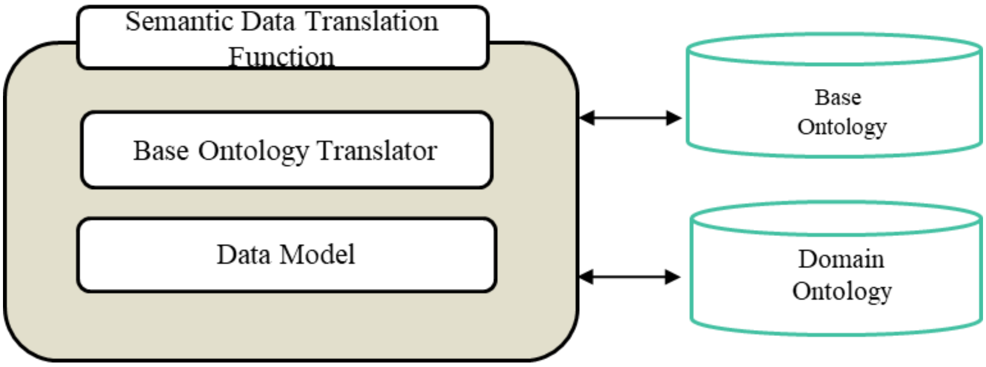


Figure 9-2. Semantic Data Translation functional components

- **Semantic data model**

A semantic data model describes the data using semantic information regarding the data. The meaning of data differs from one context to another context. Semantic information of a data defines the meaning of data based on the context of the data. Different applications use 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 is related to the real world object. A semantic data model has been shown in Figure 9-3.

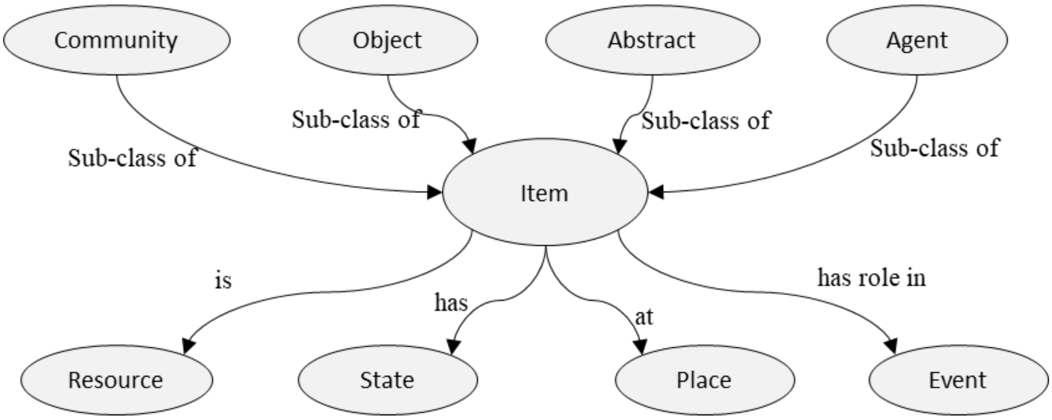


Figure 9-3. A semantic model of ‘Item’.

A semantic data model has 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 object is expressed as a triple format;
- In a triple, the relationship between an object with other object is expressed using binary relation;
- The triple includes subject-predicate-object relationship form. The first term in the triple is subject that is an object for which a statement is expressed. The second term in the triple is a predicate that is the property that makes a relationship between subject and object. The third term

in the triple is an object that represents a resource in the fact. So, the example of the fact becomes 'Richard hasA Smartphone';

- A fact must include a kind of relationship between a subject and multiple objects. Meaning of the relationship is required to interpret the meaning of the fact;
- Kind of relationship needs to be standardized.

9.3 API request query translation

This function delivers an interface to receive requests for the data to be translated to particular registered semantic format. As different platform has diverse formats to support requests their translation need to be provided. The API Request Translation function provides a common way to translate the requests to specific API format, in order to provision seamless interoperability.

The functional capabilities of API request query translation 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 query
- API Formats: Provide the API formats of the registered platforms

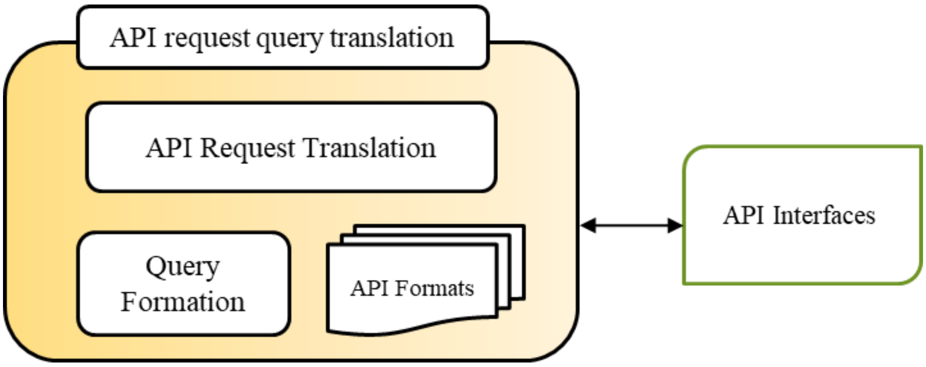


Figure 9-4. API request query translation

9.4 Semantic annotation function

This 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 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.

Features of semantic annotation capability includes:

- Describing the relationship between concepts and ontologies;
- Links information source to an ontology;
- Assignd semantic concepts and properties to the target data;

9.5 Semantic alignment and linking function

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

The functional capabilities of semantic alignment and liking 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 source and target ontologies.

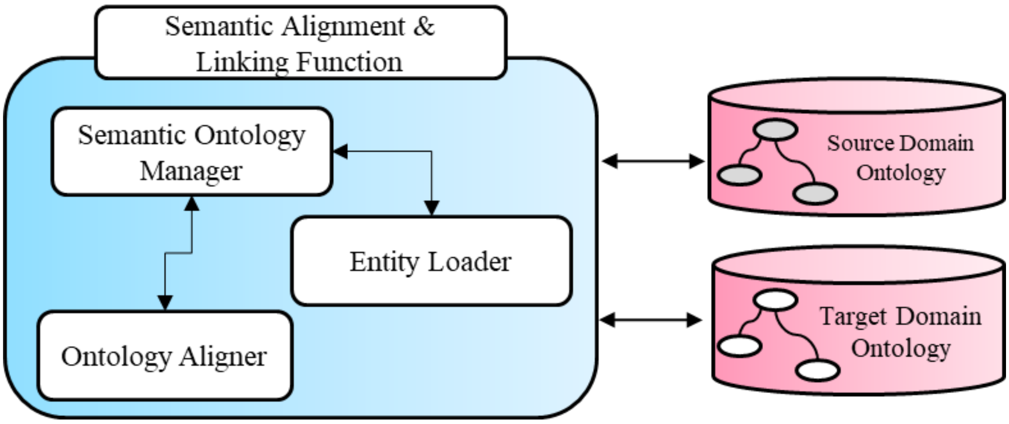


Figure 9-5. Semantic Alignment and linking

Any IoT application that deals with several ontologies must provide a semantic mapping or matching mechanism to ensure the interoperability. When ontologies are matched they are indeed aligned, which represents 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 following figure. The names in the circles represent the concepts whereas the values on the arrows define the confidence among the concepts. The confidence is also referend as the value of similarity. The mapping of matching techniques generates this similarity value. The higher the value between two entities the more exact match is assumed.

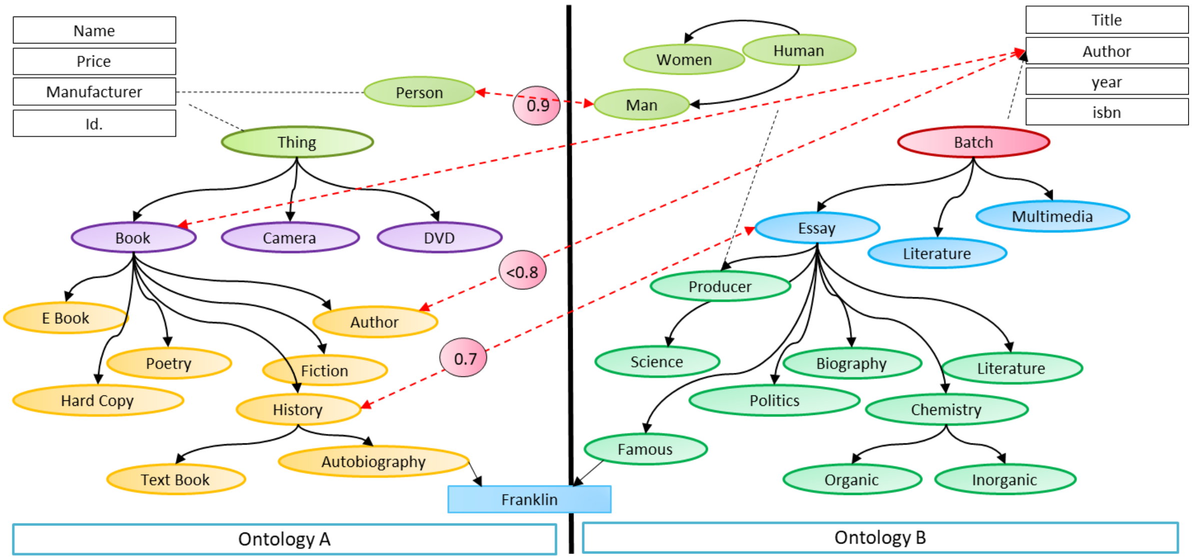


Figure 9-6. Example of ontology alignment

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 function:

The 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 validation test case defined on the bases of semantic ontology.

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

- Validator: this functional capability 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

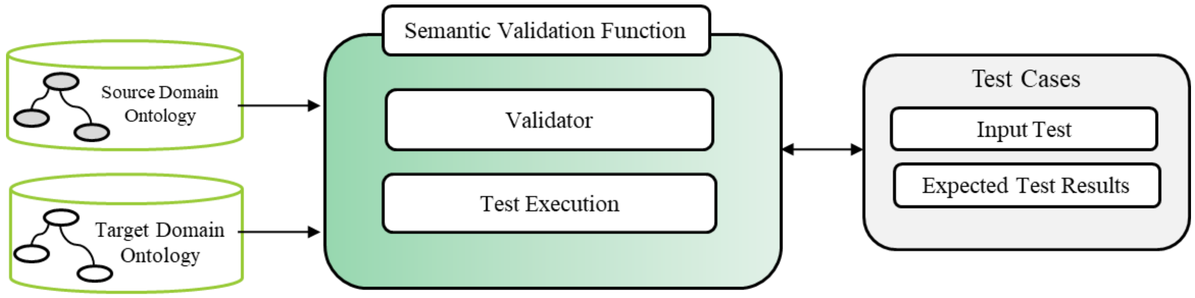


Figure 9-7. Semantic Validation functional components

10. Details on syntactic mediation functions

This chapter provides the details on syntactic mediation functions in accordance with the functional model of data interoperability.

In heterogeneous IoT environment, syntactical 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 REST API exposes interface based on REST methods.

We can realize 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.

Following section describes the functional aspects with respect to syntactic mediation.

10.1 Syntax description function

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

The functional capabilities of syntax description function are as follows:

- Syntax registration interface: provides interface to record the platform syntax profiles in syntactical description formats DB.
- Template discovery function: 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 represents the diverse formats in a unified way. A common data model (CDM) that is a set of data elements that can be mapped among different interoperating systems. 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 requirements;
- A CDM allows 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 10-1.

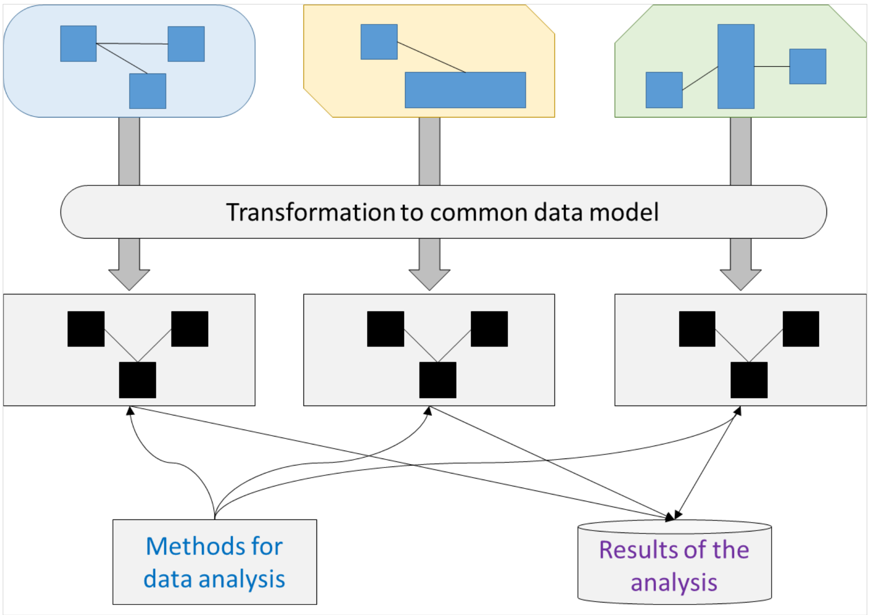


Figure 10-1. An example of conceptual model of a CDM.

Conventions, features and characteristics of a CDM have been 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 of data elements;
- Due to increased number of dataset, the CDM should be scalable so that it can support increasing number of dataset for data processing and management;
- CDM should be platform independent;
- One convention should be followed across different tables for naming variable;

To improve the common understanding of data elements, quality of data and for sharing among multiple application domains, a CDM has an important role. Depending on the types of data elements and data, different types of CDM are required that have been listed here:

- A universal CDM is used for study purpose, not for specific interest (e.g subjects in health study) that require extraction of information from multiple sources regardless of boundary or countries (e.g. disease may be common to those 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 of them;
- A required CDM is requested for institutional policy or for keeping record for the institution;
- Domain specific CDM focusing on specific study.

10.2 Syntax interpreter function

Syntax interpreter function provides conversion and translation among diverse data formats. In 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 capabilities of Syntax interpreter function are as follows:

- API syntax convertor: provides capability to translate the API syntax description to standard format.
- Profile manager: delivers the capability to the store and retrieve syntax profiles.

10.3 Schema translation function

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

The functional capabilities of Schema translation function are as follows:

- Core schema translator: provides the functional capability to translate the core data schema with respect to syntactical conversion in CDM.
- Meta data 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 a data such as data types, description, etc. A metadata is an application oriented predefined model that describes the data, data types, data files, logics and constraints that are required to analyse the data for application specific decision making. Due to increase in amount of data, metadata model describes the data as well as links them with other model as knowledge base.

10.4 Syntactical validation function

The function 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 core schema.

The functional capabilities of semantic validation function are as follows:

- Syntax format validator: this capability 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 syntactical mediation functions

The component based interaction of the syntactic mediation functional unit has been shown in the illustration of Figure 10-2. To provide syntactical interoperability a syntax translation microservice generates the conversion schema through the syntax interpreter service. The main goal of SIS 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 includes the discovery of syntax template from the template repository, the syntax registration once a conversion has been performed successfully, and syntactical Meta data management to support additional syntax level checking. To provide the verification of syntactical conversion syntax format validation function tests the generated translation schema against a pre-validated test case in order to insure the integrity of mediation process.

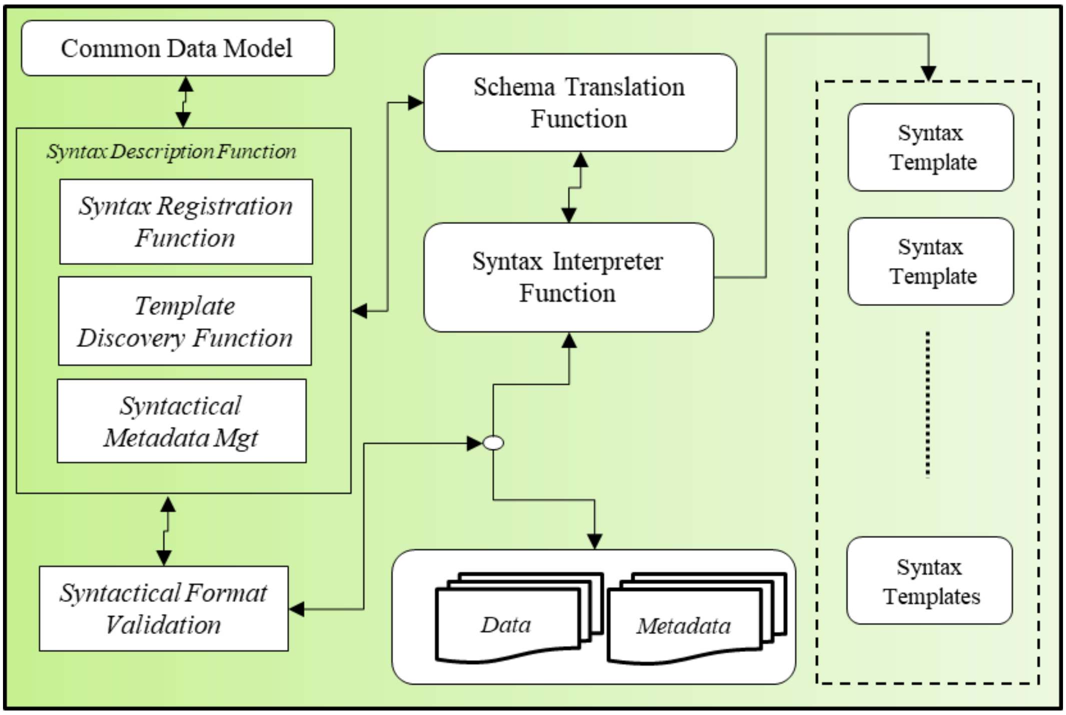


Figure 10-2. Syntactical mediation components' Interaction

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

11. Details on the interoperable object abstraction functions

This chapter provides the details on interoperable object abstraction functions in accordance with the functional model of data interoperability.

11.1 Data Classification

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

The functional capabilities of data classification function are as follows:

- Classifies the data based on the metadata available with the core data.
- Tagging the data according to the abstract representation model.

11.3 Data Integration

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

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

- Integration based on the classification category, when data has been classified and category is known
- Integration based on the metadata description, when the data has not been classified, and metadata provides the 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.

11.2 Meta data description and coding

Metadata descriptions and coding function provides the capability to assign new metadata or assign additional metadata to the converted data. It enables to assign codes and other meta-data description to the core data in order to identify its sources. Meta data 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 file 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 describe their relationships.

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

11.4 Interaction of object abstraction functions

The implementation component and their functional interaction to generate abstraction object representation can be realized through the following functional diagram (as shown in Figure 10-3). Here, the data representation processing function is the central processing component of object abstraction which enables the instantiation of abstraction from existing profile by calling on 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 $P = \{P_1, \ldots, P_n\}$, 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 extract the concepts from the representation ontology to generate the data abstraction. These generated abstractions are registered in the data representation repository to enable the reuse whenever they are required.

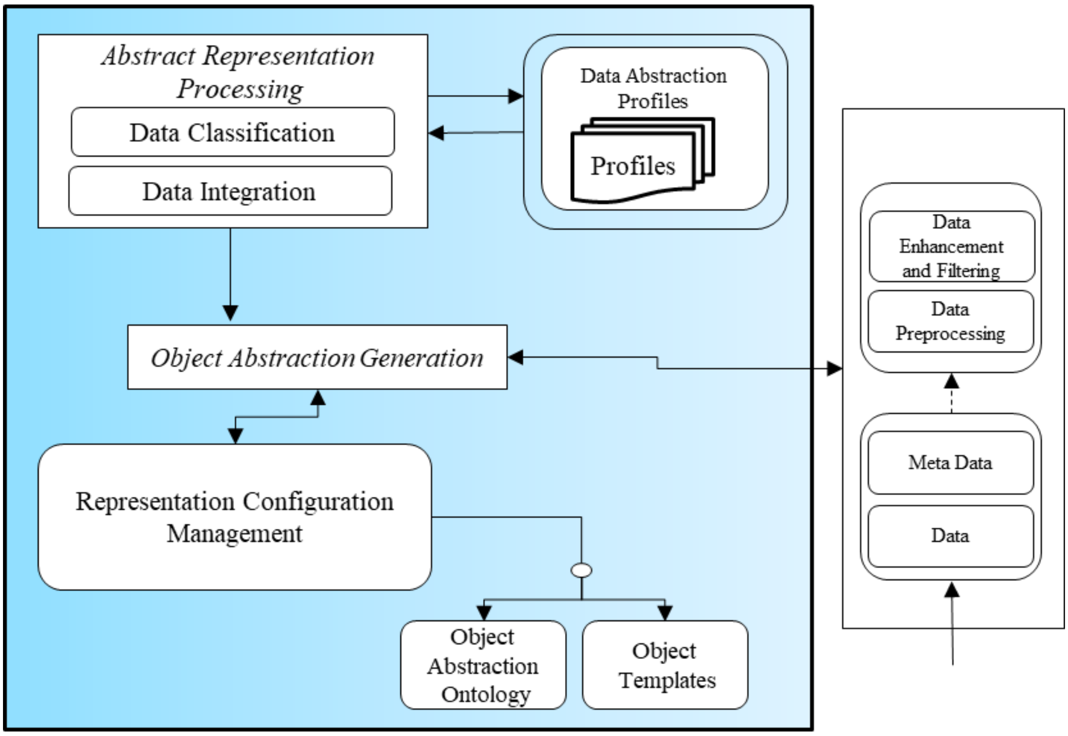


Figure 10-3. Object abstraction mediation functions interaction

12. Details on the interoperable mediation support repositories

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

12.1 Base ontology repository

Base ontology repository provides the capability for the persistent storage, retrieval on the entities of base ontology model provides

The base ontology model consists of the following capabilities

- The semantic base ontology provides the generic concepts to support the semantic interoperability.
- Base Ontology model have 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: It defines the subject for which a fact is stated;
- Base range: It defines object of the fact of base vocab;
- Base object properties: It make relationship between subject and object in base vocab;
- Base data properties: Describes the instance using attributes of the instance in base vocab;
- Base annotations: Used to annotate or describe various components in an base ontology.

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

12.2 Domain ontology repository

To express semantics of any concepts, and to organize them in a standard 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 relationship among them that represent a RDF graph. In semantic web, RDF vocabularies are known as RDF schema or ontology.

A domain's semantic ontology provides following features:

- understanding of structure of information IoT domain;
- capability to reuse of domain knowledge
- allows reuse of concepts and properties in the ontology;
- If concepts and properties are not available in local then similar concepts and properties are searched in other ontologies that can supports similar functionalities.
- enables inference on domain knowledge to make decisions;

Components in a domain ontology model include:

- Domain: It defines the subject for which a fact is stated;
- Range: It defines object of the fact;
- Object properties: It make relationship between subject and object;
- Data properties: Describes the instance using attributes of the instance;
- Annotations: Used to annotate or describe various components in an ontology.

W3C has standardized number of RDF vocabularies to organize and to express concepts. An ontology allows designing a model for expressing concepts, constraints and relationship 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 database

Syntactic description DB is the storage repository to facilitate 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 store and maintain syntactical descriptions metadata

Annex A
Interoperability with Web of Objects (WoO) based Model

(This annex forms an integral part of this Technical Specification.)

This section 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].

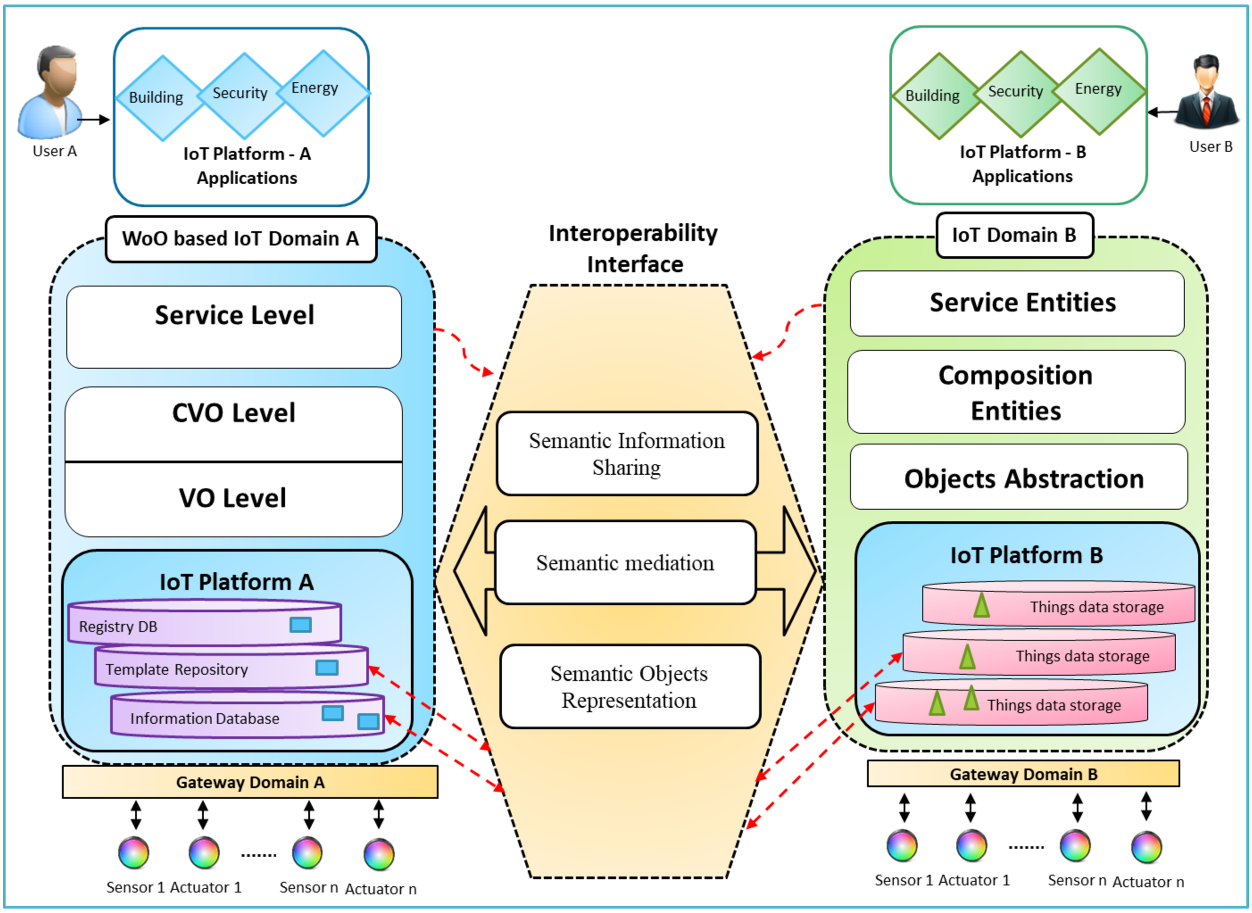


Figure A-1. Interoperability with Web of Objects (WoO) based Model

A.1 Web of Objects based IoT domain

Figure A-1 describes the architecture of 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 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 enable 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 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 of 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 of data and metadata of sensors registered in the domains

- Semantic Mediation

The 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 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.2068].

- The service entities support the service functions based on the service provisioning capabilities indicated in section 8.1 of [ITU-T Y.2068].
- The composition of entities in external IoT platform will provide IoT object orchestration for higher level service workflows based on the application capabilities defined in [ITU-T Y.2068].
- Object abstraction and data handling functions in the external IoT platform provides capabilities responsible for the management of object handling based on the data management capabilities in [ITU-T Y.2068].

Annex B

Data Interoperability of semantic and non-semantic data

(This annex forms an integral part of this Technical Specification.)

Nowadays complex IoT systems such as in smart city domain involve 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. This chapter describes some aspects based on the above view and provides a use case scenario in the appendix V which shows processing for interoperability among semantic and non-semantic data in 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 analyzing them for extracting knowledge for high level applications. To support semantic interoperability following data processing and management characteristics should 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 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, CSV format, as well as other formats need to be supported.

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

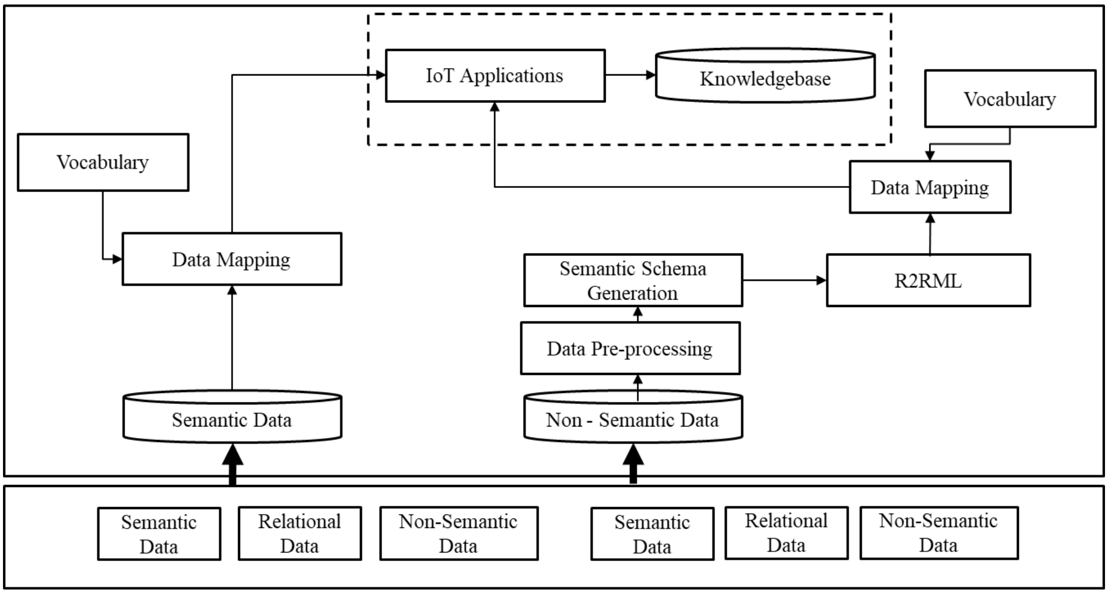


Figure B-1. IoT Data processing for semantic and non-semantic data Interoperability

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 knowledge base;
- Data is represented in RDF format and stored in OWL format;
- Knowledge base 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 include XML, CSV, etc.

Schema mappings:

- specify 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 concepts, properties and instances of target schema.

Semantic schema generation:

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

- in case of relational database, or XML data, semantic schemas are constructed from 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 user.

Transformation of non-semantic to semantic data schema:

- Semantic definition of domain terminology or semantic data schema is defined by domain expert or developer based on common vocabulary. So, if the source data schema is already in semantic format then in does not require any processing for mapping to target schema. But, if it is non-semantic data schema then it requires conversion into semantic data schema. That means non-semantic data schema or data from 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 Technical Specification.)

With respect to the semantic information model defined in section 8.3, this section provides available examples of Semantic Sensor Ontology model and 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-3], [b-15] defines a set of domain independent concepts. SSN is defined by the W3C consortium. The SSN ontology is conceptually divided into 10 modules as shown in figure I-1. The ontology contains 41 concepts and 39 object properties. The ontology describes sensors, also 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 observations of particular property of a feature. In this case the observations are described but feature and property are left vacant place holders.

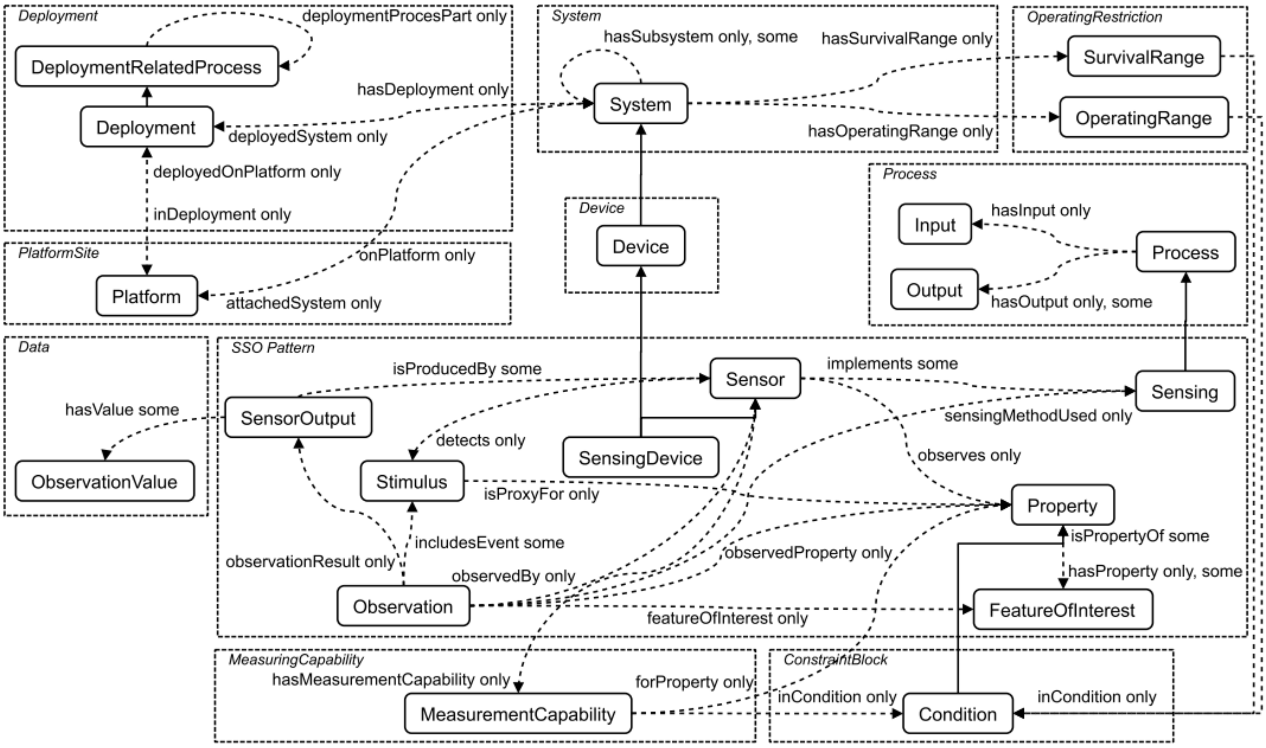


Figure I-1. The SSN ontology, key concepts and relations [b-3]

I-2 Web Of Things (WoT) Description Model

The W3C Web of Things (WoT) [b-7] 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 and behavior of IoT program.

WoT Thing Description

The WoT Thing Description (TD) is structured data which 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 by two ways. Firstly, TDs enable machine-to-machine communication in the Web of Things. Secondly, TDs can provide a common format for developers to document and retrieve the information required to access IoT devices and their data.

The following figure provides a view on how WoT description can be used in WoT Architecture with building blocks.

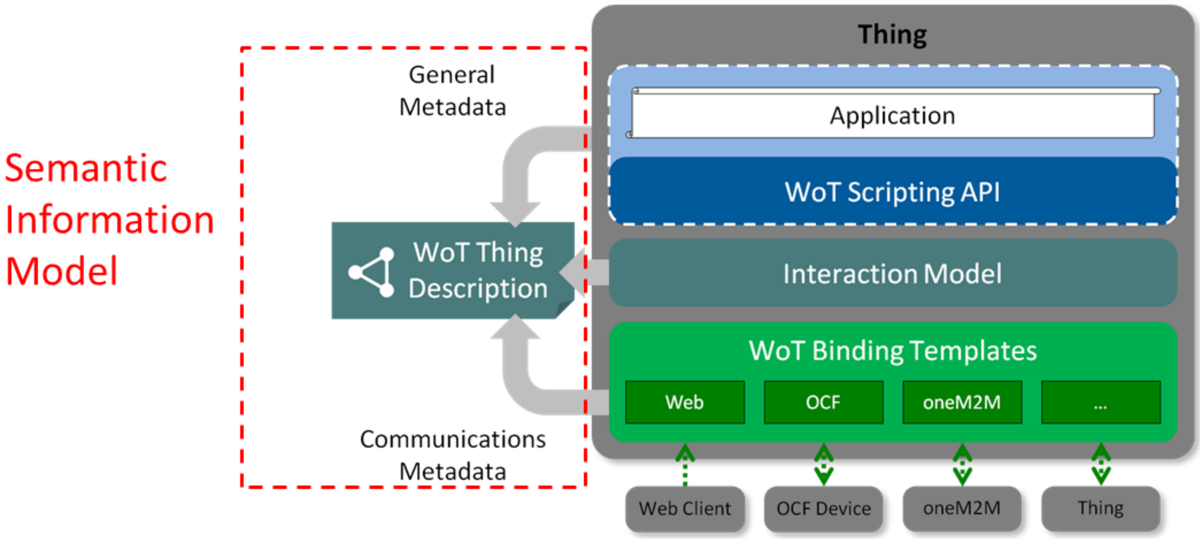


Figure I-2. The model of thing description with WoT architecture [b-14]

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 a 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 and implemented and maintained as a separate entities at a national and local level.

To share common understanding and unambiguous meaning of data, and to support data interoperability, 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 that should follow the 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 data model, rather a data model is build according to a standard vocabulary.

A standard vocabulary have following characteristics:

- A vocabulary contains different terminologies from different sources;
- Each terminology in vocabulary is defined as a concept;
- Each entity in vocabulary is identified by unique identifiers (e.g. URI for concept) to link the concept;
- Relationships among the concepts are maintained for the semantics of the concepts (e.g. meaning of a same concept varies depending on the context);

- For naming a concept, synonyms should be included for meaning in different context among multiple domains as meaningful;
- Should include concepts as standard concepts that will describe the information regarding the data model, source of the model, creator the data mode, organization, etc.

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

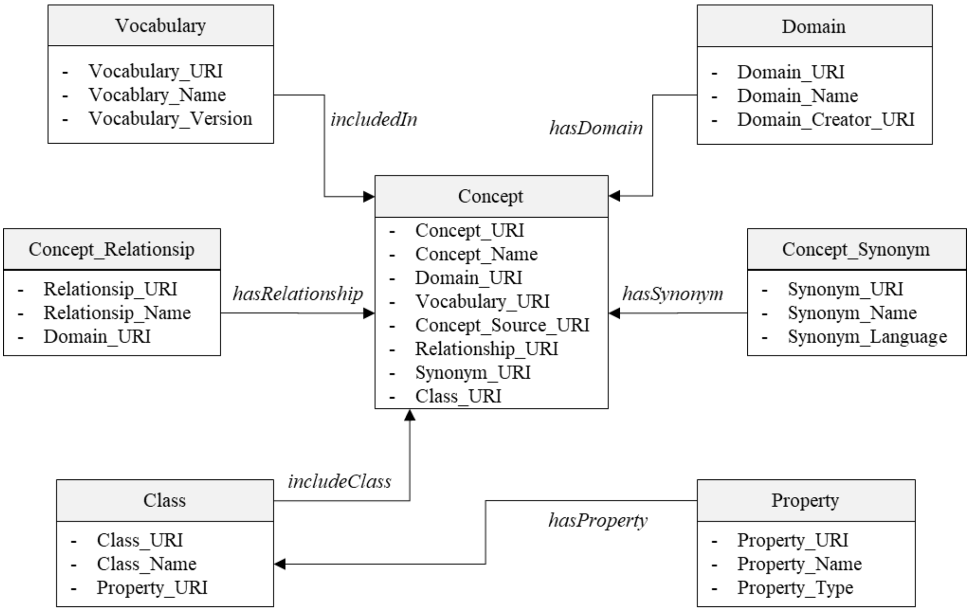


Figure I-3. A model of vocabulary concept

Linked Open data

- Linked open data: Linked open data is a semantic framework which 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 machine-readable and human understandable format on the internet. In Linked data design 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-8, b-9].

- Linked open vocabularies (LOV): Linked Open Vocabularies (LOV) are a RDF-based repositories in which data publishers and consumers can lookup vocabularies [b-10, b-11]. 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 initiative are like 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 Technical Specification.)

There have been some initiatives that cater interoperability in different domains. One of these is the project Haystack [b-2] which provides interoperability among applications across buildings.

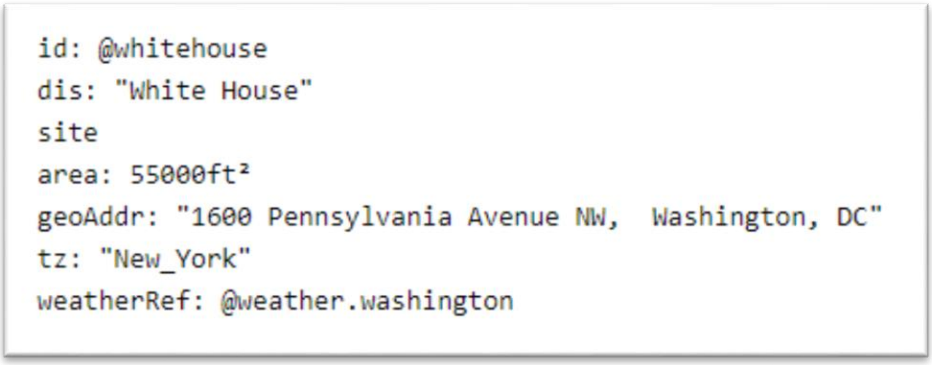
Data models and formats in Haystack Project

The project is an open source initiative to promote interoperability of applications [b-2]. It describes data models, data formats and data structures to share data on HTTP with REST APIs. Haystack Project introduces concepts like TagModel, Structure, TimeZones and Units. The TagModel is a meta-model which uses tags. The tags are name/value pairs which are associated with an entity. As tags are simple and dynamic therefore they provide flexible way to build standardized models and are customizable according to any project or any equipment. Further the tagging models are integrated and layered above existing models such as object oriented classes or relational schemas. On the other hand, the notion of entity is the real world object abstraction in haystack. Entity may include sensors, weather stations, equipment, buildings etc. From a software systems point of view an entity can be modelled as record entry in a database or an object.

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

Tags:

It is a pair of name/value which is associated with an entity. It describes any fact or attribute about an entity. If we associate a *site* tag with an entity than it is considered as building. If we include *geoAddr* then we declare 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
```

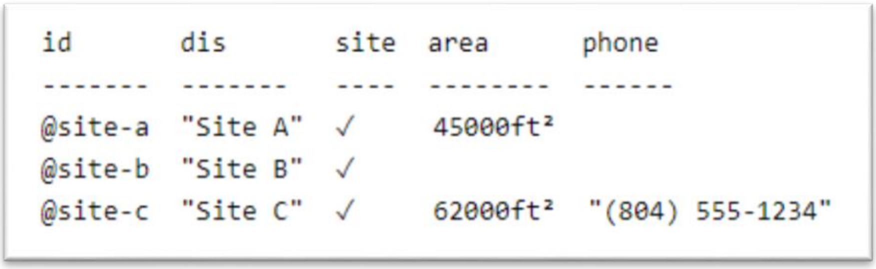
Figure II-1. Example for an entity describing a site with corresponding tags [b-2]

Grids:

A tabular two dimensional data structure used to serialize tagged data. Tabular representation of tagged entities in a grid are serialized over HTTP using the Rest API. The grid constitutes of:

- Meta data at grid level
- Columns of a programmatic name and Meta-data
- Rows defined as scalar cells

An example of 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-2]

Haystack project provides data formats for easy exchange of data. It introduce 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: Java script 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 Technical Specification.)

III.1 VO information model in WoO based IoT environment

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

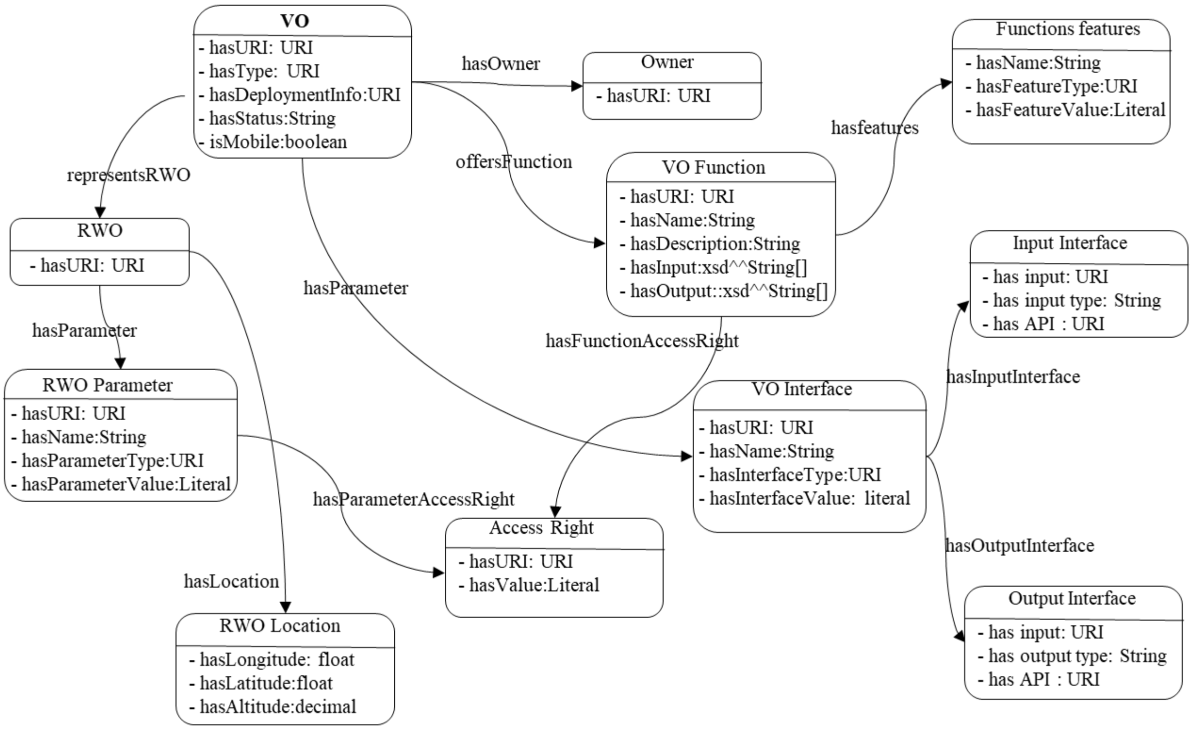


Figure III-1 VO information model.

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 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 CVO information model allows service publicizing, reusability and interoperability among heterogeneous application domain.

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.

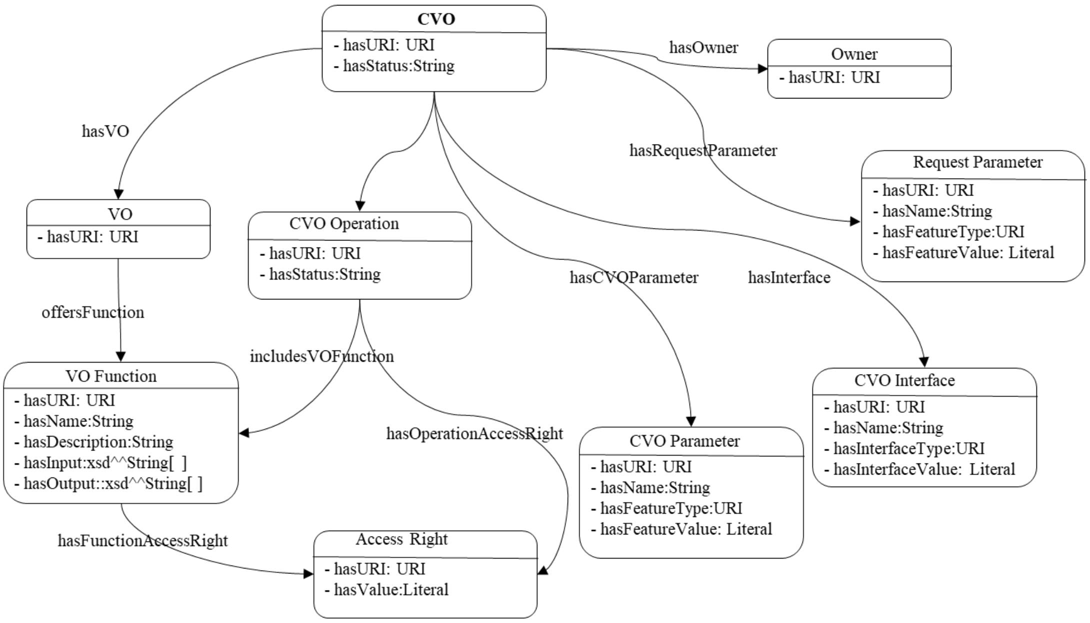


Figure III-2 CVO information model.

Features of a VO information model:

- Allows interrelationship among service relevant VOs, VO functions, CVO operation, parameters, etc.;
- Includes a set of metadata that describe 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 use of semantic ontology that should be registered and maintained by semantic ontology registry (SOR). W3C standards on semantic ontology and ISO/IEC metadata standards have been adopted for the SOR.

To allow the 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 is enabled by SOR framework.

III.3.1 SOR framework

To support data interoperability a 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 ontology that are maintained in SOR;
- Extraction of the description of the data element from SOR;
- Reusing of the data element by linking the respective data element;

To support the 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 terminology from standard vocabularies;
- Query the relationship of the data element across the SORs;
- Linked data principles has 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 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 vocabulary. There are several terminology servers that follow linked data principle.

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

- Each of the data element in the SOR needs to be annotated and linked with external terminology server using URI in the linked data cloud;
- To interrelate data elements in SOR cloud with terminology server, some components are used to link between them that include"
  - Object classification (OC) that identify the object of the data element;
  - Object classification item (OCI) that includes description of vocabulary, specific object in the vocabulary, 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 value;
- SKOS is used to interrelate one data element in one SOR with another data element in different SOR;
- OCI is also to make relationship between data elements in different SORs;

- Source data element is identified by source URI, and target data element is identified by target URI.

This mechanism of linking data element with the terminology server allows the data element across the SOR in the cloud to use same terminology, same descriptions with same resources that provides same meaning of data element and 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 element with terminology server.

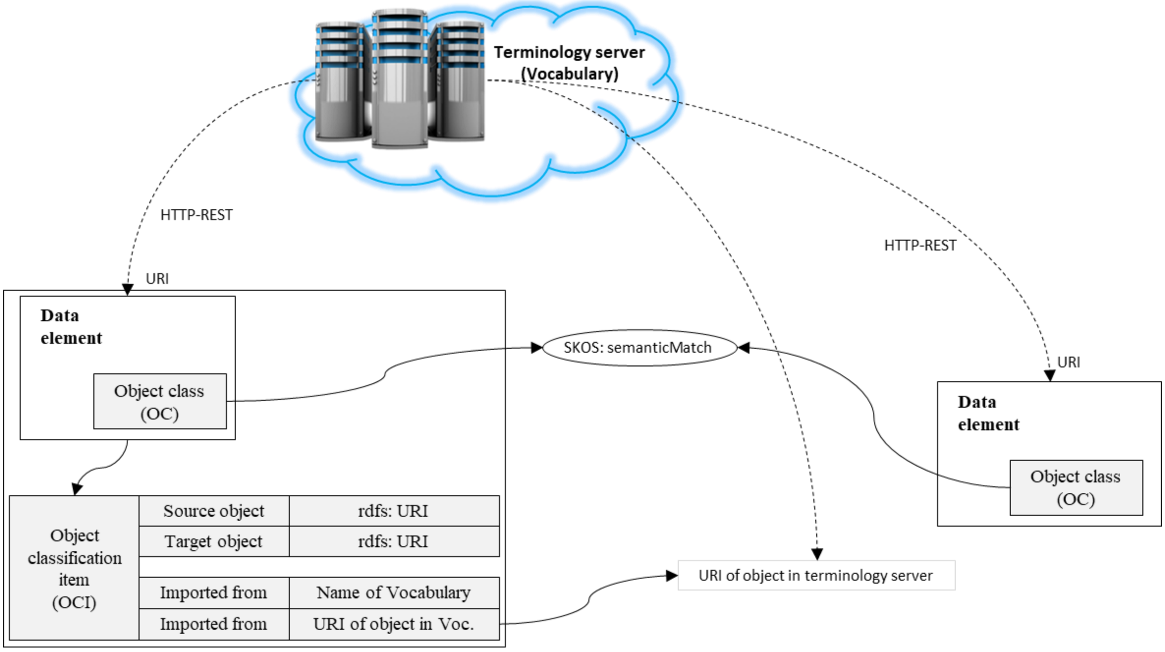


Figure III-3 Mapping of data element to terminology server for common vocabulary to support data interoperability.

III.3.3 Utilization of linked SORs for data interoperability

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

Following steps describes the process to search and link data elements in SOR:

- Application server searches local SOR to retrieve data element along with theirs 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 SOR;

- Semantic ontology manager responses Application server with the matching set of data elements according to requirements of the ontology model;
- Application server searches the similar terminology based on extraction specification of data elements;
- SPARQL query manager is used to query the SOR in cloud for the specifications of data elements. It searches specifications of each registered data elements 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 cloud responses with the URI of the specifications of data element to the query manager, and then the manager responses to application server with the URI of specifications.

Data interoperability provisioning in WoO based IoT environments supports:

- to aggregates, process and manage data;
- to follow semantic data model to describe concept and their relationship;
- to use semantic schema for describing relationship among the concepts;
- to virtualize real world objects to Semantic ontology;
- to format XML/RDF format to express virtual objects that are stored in 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 specification of domain;
- to allow use of common terminology in the vocabulary to develop ontology.

Appendix IV

Legacy approaches for data interoperability in IoT environments

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

To support data interoperability in the IoT environment, several approaches have been discussed in the literature which are briefly described as follows:

IV.1 Semantic information sharing approach to support semantic interoperability

The semantic information sharing model is based on autonomous agents which interact by sharing semantic information with one another with RDF/OWL. To achieve semantic level interoperability and support information reusability information is described in RDF and OWL form. In one of the proposed architecture [b-4], a model is defined on four main entities in the system (presented in fig. IV-1). These include Sematic Information Broker (SIB), Agent, Virtual Entity (VE), and Resolution Infrastructure (RI). The SIB is a semantic information sharing service which enables semantic level communication channel for Agents to interact. The work in [b-4] describes the protocol providing the rules and syntax for Agent-SIB communication which is known as Knowledge Access and Management Protocol (KAMP). In KAMP the operations to access and manipulate the knowledge are based on SPARQL 1.1 Query and Update languages. Information in SIB represents the context for a smart space. Smart space is physical space monitored by agents where agents share information and capabilities via a SIB. Here agent has three functions: Semantic Interface (SI), ucode Resolution Client (uRC), and application logic. The Semantic Interface facilitates the transformation information from local data formats to semantic format. It also provides interaction with the local SIBs and the SIB Resolution Service. The uRC interacts with the ucode Resolution Server to locate a SIB service hosting information regarding a real world object. An agent can interact with the physical sensors and actuator and collect their information. Agent publish information about real-word objects into a SIB which is collection of RDF triples representing the object which is called VE. The VE can represent sensors, actuators, buildings and people, etc.

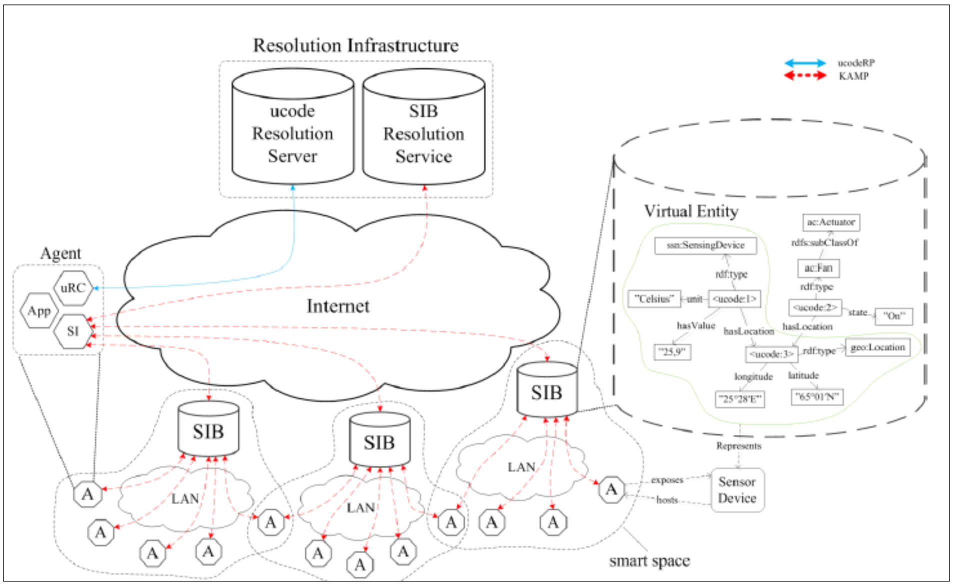


Figure IV-1. Semantic level interoperability architecture for IoT [b-4]

In this architecture ontology is used to represent VEs. Moreover, VEs are identified with ucodes and the same ucode is stored into a tag attached to the related physical object which connects the Virtual Entity to its physical counterpart. Agents locate Semantic Information Brokers with Resolution infrastructure. The Resolution Infrastructure consists of two components ucode Resolution Server and SIB Resolution Service. The SIB Resolution Service maintains semantic descriptions of other SIBs and faclitates discovery for the Agents. The ucode Resolution Server provides lookup functionality by mapping the physical object identifiers to SIB address where about VE is available.

IV.2 Open API provisioning

Instead of providing their own interfaces, IoT platforms can be enhanced with IoT API [b-5] which can provide common interfaces to cover required interoperability functions.

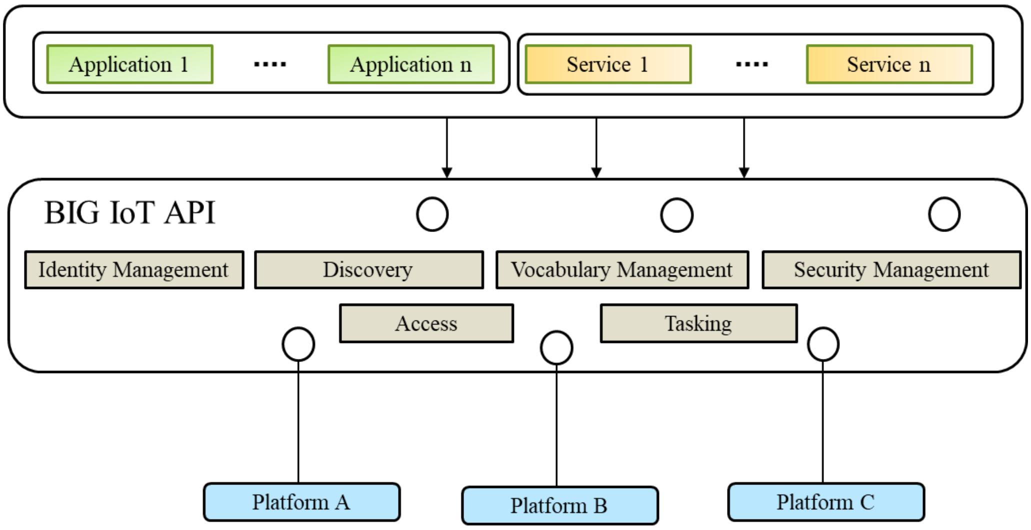


Figure IV-2. BIG IoT (Bridging the Interoperability Gap of the IoT) API.

In figure IV-2 depicts a model of the basic ecosystem architecture presented in [b-5]. A variety of devices are associated with different IoT platforms. However, rather than providing own interfaces, the platforms are supported with the BIG IoT API. This API is a common interface that provides the functionalities required for interoperability with other platforms. Platforms can operate different levels of the system. It can operate on the cloud level such as, a server or datacentre, fog level such as, a gateway or base station, or device level like, wearable, a Raspberry Pi or smartphone. The BIG IoT API can help to develop software components which serve as clients for different platforms.

IV.3 Hub based approach to support data interoperability

Another approach to achieve interoperability is use existing web technologies such as HTTP, JSON and Representational State Transfer (REST) architecture. The use of the web enables interoperability by connecting isolated things in different domains. In this way, developers can connect things using web tools and technologies and develop innovative applications and mashups.

The work proposed in [b-6] defines a four stages toward interoperability between IoT hubs.

- **IoT Core.** The web architecture and RESTful services are used to form the Hubs of web of things which expose things and their Meta data;

- **IoT Model.** In this stage, an agreement of fundamental methods and models on what things and data a hub will contain is developed. Fulfilling this stage will result in the development of integration tools to achieve hub interoperability;
- **IoT Hub.** At this stage consensus on the implementation issues is developed on the representations of data, schemas to describe and query data from hub and URLs. This stage also considers security mechanisms to control access on providing and consuming things and data to and from the hub;
- **IoT Profiles.** This stage defines agreement on the things semantics and associated data exposed on a hub. Such as a temperature sensor in one hub provides the same quality and units of temperature similar to another hub. This stage defines the taxonomy of things and the ontological models which will be supported by the hubs.

An abstract representation of an IoT hub is shown in the following figure. After reaching agreement an integration of application is possible which will allow hubs and things to link to and communicate directly with one another.

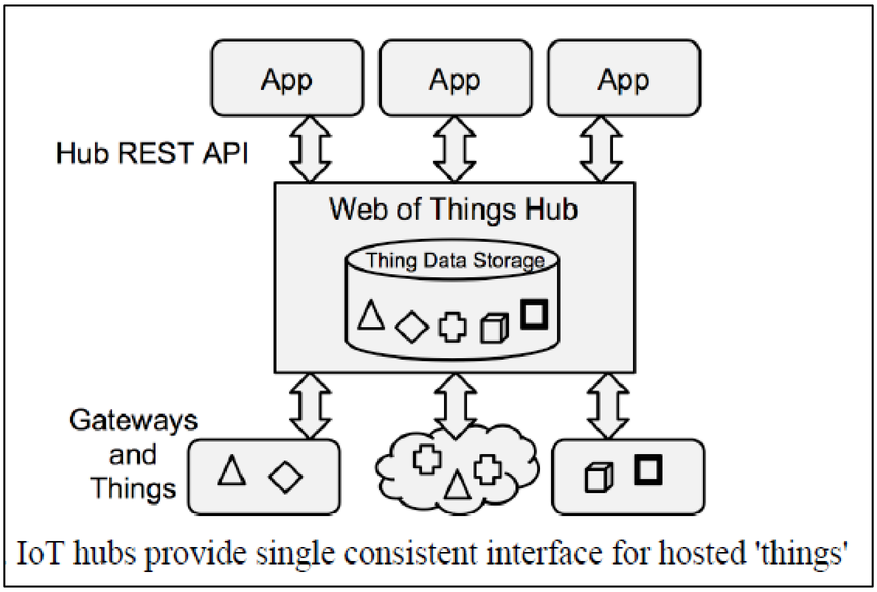


Figure IV-3. IoT HUB [b-6]

Bibliography

[b-1] Jerome Euzenat & Pavel Shvaiko. Ontology matching. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[b-2] https://project-haystack.org/

[b-3] Compton, Michael, et al. "The SSN ontology of the W3C semantic sensor network incubator group." Web semantics: science, services and agents on the World Wide Web 17 (2012): 25-32.

[b-4] Kiljander, Jussi, et al. "Semantic interoperability architecture for pervasive computing and internet of things." IEEE access 2 (2014): 856-873.

[b-5] Bröring, Arne, et al. "Enabling IoT ecosystems through platform interoperability." IEEE software 34.1 (2017): 54-61.

[b-6] Blackstock, Michael, and Rodger Lea. "IoT interoperability: A hub-based approach." Internet of Things (IOT), 2014 International Conference on the. IEEE, 2014.

[b-7] https://www.w3.org/TR/wot-thing-description/

[b-8] https://www.w3.org/standards/semanticweb/data

[b-9] https://en.wikipedia.org/wiki/Triplestore

[b-10] Vandenbussche, Pierre-Yves, et al. "Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web." Semantic Web 8.3 (2017): 437-452.

[b-11] https://lov.linkeddata.es/dataset/lov/

[b-12] 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-13] 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-14] https://w3c.github.io/wot-architecture/#sec-building-blocks

[b-15] Haller, Armin, et al. "The SOSA/SSN ontology: a joint WeC and OGC standard specifying the semantics of sensors observations actuation and sampling." Semantic Web. Vol. 1. IOS Press, 2018. 1-19.

[b-16] ITU-T Technical Paper (11/2016), *Analysis of Digital Data Technologies Toward Future Data Eco-Society*.