

International Telecommunication Union

ITU-T

Technical Report

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(19 July 2019)

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

Technical Report D2.3

Web based data model for IoT and smart city

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 the 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.

© ITU 2019

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Technical Report D2.3

Web based data model for IoT and smart city

Summary

This Technical Report addresses the following issues:

- generic consideration on data formats;
- discussion on conventional categories of metadata;
- necessity of a new subcategory of metadata for IoT interoperability;
- useful information on concepts and types of data model;
- advantages and importance of Microdata formats for Web data management;
- a new metadata category, procedural metadata, with its fundamental principles, descriptions, primary uses, and example properties.

Acknowledgements

This Technical Report was researched and principally authored by Nakyoung Kim (KAIST of Korea, Rep. of), Hyeontaek Oh (KAIST of Korea, Rep. of) and Jun Kyun Choi (KAIST of Korea, Rep. of) under the chairmanship of Gyu Myoung Lee (Korea, Rep. of).

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

Keywords

Data, Metadata, Microdata format, Data model, Data interoperability, Web-based, IoT, Smart city

Technical Report D2.3

Web based data model for IoT and smart city

CONTENTS

- 1 Scope.....0
- 2 References.....0
- 3 Definitions.....0
 - 3.1 Terms defined elsewhere0
 - 3.2 Terms defined in this Technical Report.....1
- 4 Abbreviations and acronyms.....1
- 5 Conventions2
- 6 Overview.....2
- 7 General discussion on data formats and metadata3
 - 7.1 Considerations for standardized data format3
 - 7.2 Conventional metadata and their primary uses and properties4
 - 7.3 Web of Things and IoT interoperability5
- 8 Basic concepts and types of data model6
 - 8.1 Types of data model.....7
 - 8.2 Common data structures8
- 9 Microdata formats for Web data management.....10
 - 9.1 Structured data10
 - 9.2 Microdata format11
 - 9.3 Leading Microdata formats.....13
- 10 Procedural metadata for semantic Web of Things14
 - 10.1 Concept of procedural metadata15
 - 10.2 Principle of procedural metadata17
- Appendix I General description of Microdata formats and deployment statistics.....19
- Appendix II Activities for integration/interoperability in IoT and smart city.....21

Technical Report D2.3

Web based data model for IoT and smart city

1 Scope

This draft Technical Report provides Web based data model for IoT and smart city. More specifically, this draft Technical Report covers the followings:

- The suggestions for generic considerations of data format;
- The necessity for a new type of metadata for interoperability;
- The necessity and importance for a common data model for bridging existing data models;
- The necessity, importance, and adequacy of Microdata formats for data management on Web environments;
- Fundamental concepts and backgrounds on the current Web environments and Microdata formats in terms of structuring and managing data in details.
- A new category of metadata, called procedural metadata, and its basic principles.

2 References

None

3 Definitions

3.1 Terms defined elsewhere

3.1.1 application [b-FG-DPM TS D0.1]: A structured set of capabilities which provide value added functionality supported by one or more services, which may be supported by an API interface.

3.1.2 capabilities [b-FG-DPM TS D0.1]: Quality of being able to perform a given activity.

3.1.3 data [b-FG-DPM TS D0.1]: Information represented in a manner suitable for automatic processing.

3.1.4 data management [b-FG-DPM TS D0.1]: The activities of defining, creating, storing, maintaining and providing access to data and associated processes in one or more information systems.

3.1.5 data processing [b-FG-DPM TS D0.1]: Systematic performance of operations upon data

NOTE 1 - Management and converting of data from the raw state into a required output.

3.1.6 data processing and management [b-FG-DPM TS D0.1]: The combination of all activities either directly performed on or indirectly influencing data.

3.1.7 ecosystem [b-FG-DPM TS D0.1]: A network of interconnecting organisations, forming a distributed, adaptive, open socio-technical system with properties of self-organisation, scalability and sustainability.

3.1.8 Internet of Things [b-FG-DPM TS D0.1]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

3.1.9 interoperability [b-FG-DPM TS D0.1]: The ability of two or more systems or applications to exchange information and to mutually use the information that has been exchanged.

3.1.10 procedure [b-ISO 9000:2015]: specified way to carry out an activity or a process

3.1.11 process [b-ISO 9000:2015]: set of interrelated or interacting activities that use inputs to deliver an intended result

NOTE 1 - Whether the “intended result” of a process is called output, product or service depends on the context of the reference.

3.1.12 raw data [b-FG-DPM TS D0.1]: Data that has not been processed for use

3.1.13 requirements [b-FG-DPM TS D0.1]: Need or expectation that is stated, generally implied or obligatory.

3.1.14 security [b-FG-DPM TS D0.1]: Condition that results from the establishment and maintenance of protective measures that insure a state of inviolability from hostile acts or influences.

3.1.15 service [b-FG-DPM TS D0.1]: A set of functions and facilities offered to a user by a provider.

3.1.16 system [b-ISO 9000:2015]: A set of interrelated or interacting elements.

3.2 Terms defined in this Technical Report

3.2.1 Microdata format: it refers to structured data markups which describe and embed meanings of resources on the Web along with their properties and relationships by utilizing tags to convey additional metadata and other attributes in small data units.

3.2.2 Procedural Metadata: it is a new category of metadata which includes descriptive information on procedures and/or processes by utilizing other metadata and flags.

4 Abbreviations and acronyms

API	Access Point Interface
C3IM	Cross-Cutting Context Information Management
CIM	Context Information Management
CSS	Cascading Style Sheets
ETSI	European Telecommunications Standards Institute
HTML	Hypertext Markup Language
ICT	Information and Communication Technology
IoT	Internet of Things
JSON	Java Script Object Notation
JSON-LD	Java Script Object Notation Linked Data
NGSI-LD	NGSI with Linked Data
OCF	Open Connectivity Forum
OGC	Open Geospatial Consortium
oneM2M	one Machine-to-Machine
RDF	Resource Description Framework
RDFa	Resource Description Framework in Attributes
SAREF	Smart Appliance Reference
SensorML	Sensor Model Language
SWoT	Semantic Web of Things
TD	Thing Description

W3C	World Wide Web Consortium
WoT	Web of Things
XML	Extensible Markup Language

5 Conventions

None.

6 Overview

Data is now considered as the new oil for the emerging fourth industrial revolution, a new valuable asset, and an indispensable driving force of innovation [b-Forbes 1]. Both data volume and diversity are increasing overwhelmingly. Web has become the largest medium for information storage, processing, transportation, sharing, and discovery. Hence, it is critical to link the data on the web semantically. There are rising demands to intelligently manage data in a way that its properties and meanings can both be perceived and interpreted well, so that the performance and opportunity for Web data sharing and mining can be significantly improved.

However, it is becoming increasingly complex to be aware of data's contexts and relations. That is because the purposes, types, characteristics, and other properties of data are highly heterogeneous. Besides, how metadata are defined and used currently on the Web leans toward linking and connecting the resources based on their generalized properties and relationships rather than describing the context of the data in structured details and connecting them minutely.

Moreover, to cope with smart city and IoT applications toward future data-driven eco-society, large variety of new data formats will be emerging. Furthermore, different and heterogeneous metadata concepts and formats from consortia/fora or individual organizations are investigated without consensus of future data eco-society. Data interoperability and data format conversion will be one of pressing issues that need to be solved in the near future. Various industries like energy, transportation, health, biology, and geography are developing their own data formats and data models. For future convergence markets including IoT applications, the future data formats and models should be well defined and standardized for both ICT industries and other industries.

In IoT and smart cities, many different industries are collectively combined to provide converged services and applications, and so are the data from different fields. Moreover, as Cloud computing is emerging, opportunities of sharing resources from various industries and fields are getting more frequent than ever. Accordingly, the boundaries between industries are gradually disappearing. However, the data formats and models have been developed to satisfy the specific requirements of each industry and each application. Before IoT and Cloud computing, applications relied on each own proprietary database and application-specific software. The software applications are often developed at one point in time and for a particular group, thus optimized for their main function and unlikely considered data sharing as a primary requirement [b-HBR]. Therefore, data are stored and managed in silos. Data migration and integration are restricted.

The current fragmented-data ecosystem requires high costs to process and manage data since most of the data analytic works are focused on data preparation. According to [b-Forbes 2], data preparation takes up about 80% of data scientists' work. Within the 80%, collecting data sets accounts for 19% and cleaning and organizing data accounts for 60%. Data integration and migration with interoperability in IoT and smart cities are become issues raised by many research literatures [b-Ahlgren], [b-Corici], [b-Broring]. While the needs of data migration and integration are rapidly arising, it is almost impossible to reformulate the existing data formats and models since they have been settled in the current forms over a period to resolve issues at each moment and fit one-off requirements. Therefore, data formats, or encodings, which allow data to be structure and explained

in details are important. With well-structure and –explained data, data models can have tools to achieve interoperability.

7 General discussion on data formats and metadata

To support interoperability and common understandings of data on the web, many web data formats, models, vocabularies, ontologies, and other technologies have been introduced. Not only the traditional information and communication technology (ICT)-related industries but also other non-ICT industries, such as energy, transportation, health, geography, etc., are developing their own ones. Consequently, more and more heterogeneous data concepts, formats, models, and ontologies from many consortia/fora and organizations are investigated without a consensus of the future data eco-society. Therefore, interoperability remains as the urgent issue in the process of moving towards a future data-driven eco-society.

7.1 Considerations for standardized data format

For future digital data format, the key considerations are that they should support data to be easily searched, discovered, and parsed from the big piles of data. The data should be well produced so that it can be found and used without additional expert skills, which can be the barriers for users to utilize the data and for data to be widely propagated. The descriptive data helps data to be retrieved from a large volume of data files. The management data also helps data to be used in appropriate domains by those who have proper rights. For improved IoT and smart city ecosystems, the data formats and metadata as well need to properly support the data. To do so, the generic considerations of future data formats can be suggested as the follows [b-ITU-T TP].

(Raw Data Format)

Various data formats are acceptably tuned with their own objectives for creation, delivery, processing, storing, sharing, and distribution. Their specifications for reading or writing the raw data files should be understood differently since all the data formats have their own rules and syntax. Additionally, the corresponding protocols should be understood as well while transferring, processing, converting, sharing, and distributing the data file, which are depending on hardware types of devices, network, and storage as well as software environments such operating systems, database, and application platforms. The future data applications for convergence of energy, transportation, health industries as well as IoT applications will inevitable create a lot of new data formats. In the results, too many new specifications may be published to handle data formats for future convergence applications, and the difficulties in understanding new data specifications and the relating protocols can be aroused.

Accordingly, the outstanding issues are how to minimize the understanding of data specifications and protocols while handling the data. The generic considerations of future raw data formats may include the followings:

- Minimize the interpretation and understanding of data formats and syntax;
- Minimize conversion and translation overheads among data formats;
- Minimize the dependency of hardware types and software environments including encoding and decoding technologies.

To meet these generic considerations, the well-known data formats are recommended, which are easily understood or interpreted by both human and computing systems. It means that future data formats should be well defined without complicate specifications or additional explanation. If future data formats have similar rules and syntax with natural human language, it is easier for people to understand new data specifications.

(Descriptive Data or Metadata Format)

For searching the data files or data sources, the data should be well identified by their physical or virtual locations. To find out the proper or correct contents, the data sources should be described with what kinds of contents are included. In addition, the data encoding formats should be declared. To solve these issues, the data sources should be well described and the corresponding metadata should contain the semantic information for identifying, processing, and managing the data files. Some data files or sources may include tag or index information which is linked by metadata information and/or detected by a searching machine.

Currently, there are many different metadata standards for specific purposes, specific domains or particular types of data. The metadata specifies the meaning of data sources, data format, and representation rules. Additionally, many different metadata schemes are developed according to applications such as e-commerce, education, science and engineering, etc. In future convergence applications, people with domain-specific or industrial-specific knowledge need new metadata format. Similar with raw data format, the outstanding issues are how to minimize the metadata format without any additional interpretation. Therefore, the generic considerations of metadata formats may include the followings:

- Well specified without any confusions and misinterpretation;
- Minimize the interpretation and understanding of metadata formats and syntax;
- Minimize the searching, sorting, classification capabilities of raw data file or data sources;
- Be flexible or safe on technology developments toward future knowledge society;
- Optionally, the descriptive metadata is located at inside data file or linked to other separate forms.

The metadata specifications are tightly aligned with the usage of raw data format, especially for the level of intelligence on target applications.

(Management Data Format)

Some data may include private or sensitive information and need data security and protect a copyright. To protect the data and restrict the usage, many different security and copyright protection mechanisms are widely developed and used at current digital market. However, some mechanisms are only belonging to specific market and applications. For future convergence applications, multiple security and protection technologies should be developed. Therefore, the generic considerations of management data formats may include the followings:

- Minimize the data field format for management purposes both in raw data format and metadata format
- Be convertible or interchangeable between the original data sources and encrypted data sources for management if the data and metadata information are designed to be open to public.
- Minimize the interpretation and understanding of management formats and syntax

7.2 Conventional metadata and their primary uses and properties

In order to empower IoT to intelligently and actively engage in smart systems, IoT devices and data need additional descriptions of not only their various aspects but also their roles within smart systems. Many approaches have targeted providing common descriptions of devices and data, and metadata has been the widely used solution to complement data and other resources with additional information. There are various types of metadata for different purposes and characteristics as described in the Table 1 [b-Riley].

However, these conventional types of metadata face some shortcomings, as more autonomic systems are desired. Recent services and applications are expected to perform intelligent decision-making and

actions, which require information on know-how or accumulated knowledge melted into data, devices, and systems. Correspondingly, the decision-making and task-performing procedures should be able to be described in machine- and human- readable ways.

Table 1: Types, primary uses, and properties of conventional metadata [b-Riley]

Metadata		Description	Primary Uses	Example Properties
Descriptive metadata		It includes basic information for finding or understanding a resource, such as title, author, subjects, keywords, publishers.	<ul style="list-style-type: none"> - Discovery - Display - Interoperability 	<ul style="list-style-type: none"> - Subject - Genre - Publication data
Administrative metadata	Technical	It includes information to help manage the data resource. It provides information about not only how the data can be opened, read, used, etc. but also how it should be managed for future use and rights of the data.	<ul style="list-style-type: none"> - Interoperability - Digital object management - Preservation 	<ul style="list-style-type: none"> - File type, size - Creation date/time - Compression scheme
	Preservation		<ul style="list-style-type: none"> - Interoperability - Digital object management - Preservation 	<ul style="list-style-type: none"> - Checksum - Preservation event
	Rights		<ul style="list-style-type: none"> - Interoperability - Digital object management 	<ul style="list-style-type: none"> - Copyright status - License terms - Rights holder
Structural metadata		It includes information on how the components of an object are organized or structured and describes the relationships of parts of resources to one another.	<ul style="list-style-type: none"> - Navigation 	<ul style="list-style-type: none"> - Sequence - Place in hierarchy
Markup languages		It provides structural or semantic features within the data content by integrating metadata and flags (tags/indexes).	<ul style="list-style-type: none"> - Navigation - Interoperability 	<ul style="list-style-type: none"> - Paragraph - Heading - List

7.3 Web of Things and IoT interoperability

Due to the wide-varying nature of IoT, the data is currently overwhelming in not only its volume but also its diversity. Generated from heterogeneous sources, manifold data is collected, stored, processed, transported, shared, and used across different platforms and application domains. Therefore, it is difficult to achieve interoperability, and the efforts to homogenize the data ecosystem and empower semantic linkages are needed.

Through the concepts of Web of Things (WoT) and semantic web of things (SWoT), enhanced interoperability in IoT ecosystems can be fulfilled on the web, an information space consisting of data resources. The web has been developed for data semantics in order to embrace heterogeneous data, platforms, and application domains. Accordingly, the web provides technologies that support the interoperability of IoT. WoT is based on the idea that connected devices are accessible via the web to enable interoperability across heterogeneous IoT platforms and application domains [b-Raggett], [b-Kajimoto]. Furthermore, SWoT brings semantic web and WoT together to associate semantically annotated information to web-enabled IoT [b-Kamilaris], [b-Scioscia].

To achieve semantic annotation and linkage of data, many WoT approaches have targeted providing common descriptions of IoT devices and data. However, today’s advanced IoT and smart city systems

and services require intelligent decision-making and task performance, which involve bidirectional interactions of sensing and actuating. In order to empower IoT devices to intelligently and actively engage in smart systems, devices and data need additional descriptions of not only their various aspects but also their roles within smart systems, specifically in the areas of decision-making and performing tasks.

Therefore, there are needs methods to provide common understandings and descriptions of logic and workflows of aggregated devices in smart systems. Self-management and autonomic capabilities are considered to be the driver in the development of solutions to the scalability and heterogeneity problems of IoT [b-Miorandi]. Merely providing the descriptions of properties, relations, and functionalities of devices and data is not enough for intelligent smart systems and services. Accordingly, common understandings of not only data and devices but also cooperative decision logic and action workflows need to be established for automation and interoperability. Consequently, approaches to achieve interoperability in processes are emerging, and a new category of metadata also needs to be specified to cope them.

8 Basic concepts and types of data model

Data models play a critical role in data, application, system, and business across different industries as they provide the definition and format of data to support data, computer systems, and related businesses. Currently, the needs and importance of managing metadata is recognised to efficiently handle a massive amount of data in various industries, and data models are used as one of the most prevailing metadata management tools or sources [b-Dataversity 1], [b-Dataversity 2].

The necessity of data migration and integration between existing data models in a perspective of interoperability by shifting the current application-dedicated to a data-driven approach is shown Figure 1. If same data models are used for data storage and access then different applications can easily share data. Figure 2 describes the benefits that data models deliver to data, system, and business. However, systems and interfaces often cost more than they should and may constrain the business rather than support it since data models in applications and systems are arbitrarily different. This results in complex interfaces to share data between the applications and the systems, which can account for between 25-70% of the cost of current systems [b-EPISTLE].

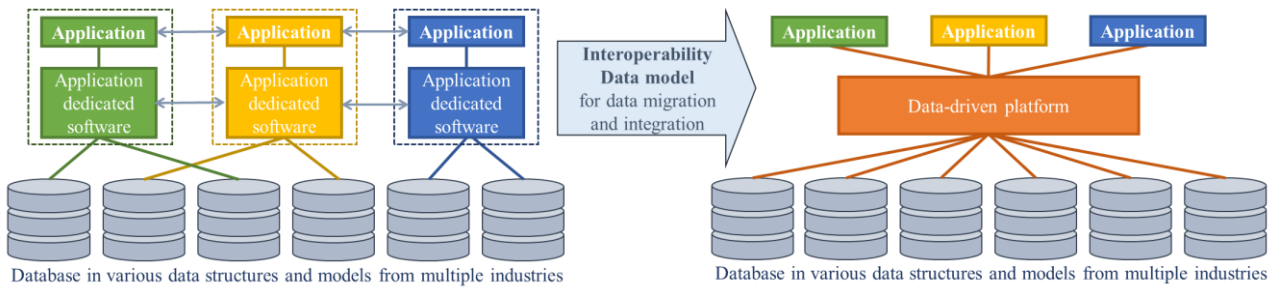


Figure 1: Data-driven approach for interoperability data models

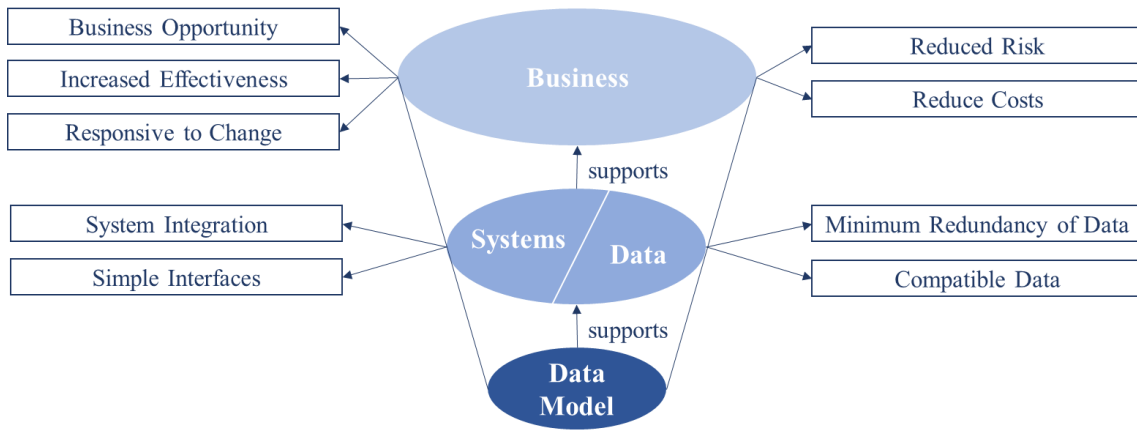


Figure 2: The benefits that data models deliver [b-EPISTLE]

8.1 Types of data model

In 1975 ANSI described the data models in three levels of data-model instance: conceptual, logical, and physical. Conceptual data model is an underlying model that describes the semantics of a domain; logical data model adds details to the conceptual model to describe the semantics; and physical data model represents the way in which data is physically stored [b-ANSI]. Table 2, Figure 3 and Figure 4 describe the data modelling today and the features of three-level the data models.

Table 2: Feature of three levels of data modeling [b-DataWarehouse]

Feature	Conceptual	Logical	Physical
Entity Names	✓	✓	
Entity Relationships	✓	✓	
Attributes		✓	
Primary Keys		✓	✓
Foreign Keys		✓	✓
Table Names			✓
Column Names			✓
Column Data Types			✓

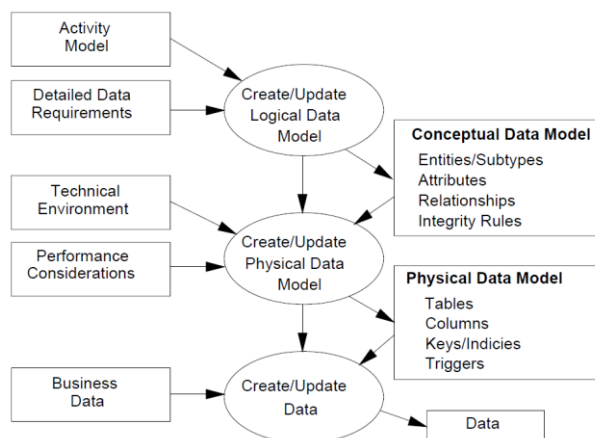


Figure 3: Data modeling [b-EPISTLE]

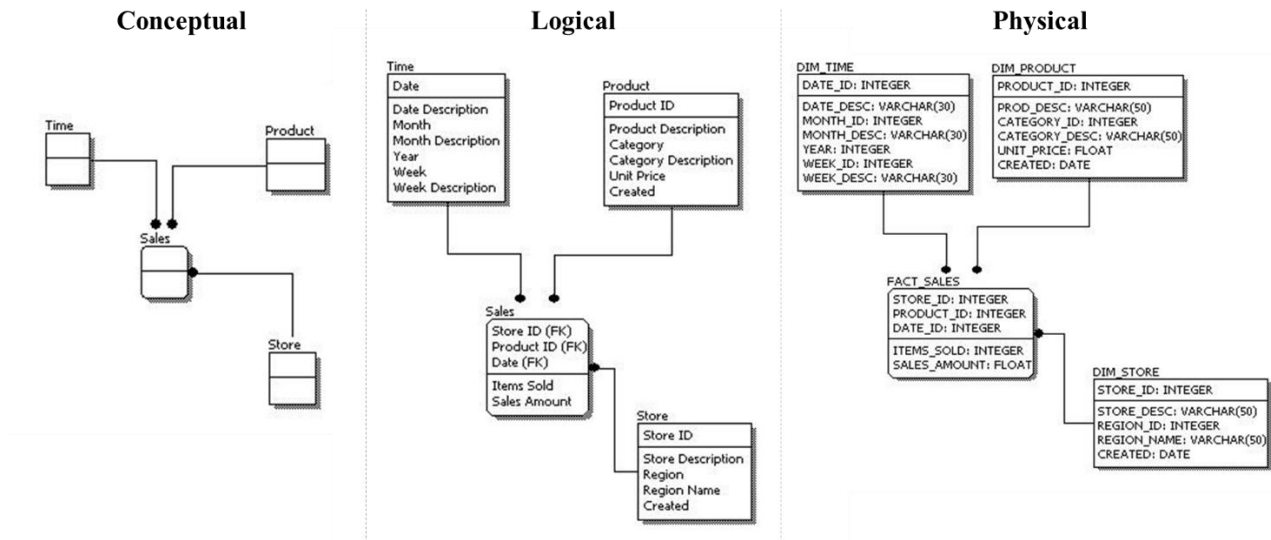


Figure 4: Example of three levels of data model [b-DataWarehouse]

As ICT and intelligence era has been emerging, how the data can be semantically described in its domain has been continuously evolved in conceptual and logical data modelling. Moreover, as IoT era also arrived, a tremendous amount of data is being generated from different fields more than ever before. Accordingly, physical data model, how the data is stored in database and how the database is structured to efficiently embrace the data, has entered upon a new phase of the study.

8.2 Common data structures

Data structure is a way in which data is stored, so that the data can be efficiently used. As data structure is fixed as stored, the initial processing to format the data in appropriate structures for the future use at the time of collection or generation is important. Many data structures have been suggested, but common ones are the followings and their visual representations are shown in Figure 5:

- **Array:** It systematically arranges similar object to store values, often in rows and columns.
- **Hash table:** It implements an associative array abstract data type and maps keys to values by indexing.
- **Linked list:** It consists of a group of nodes representing a sequence. The sequence is a linear collection of data elements which of each points to the next.
- **Stack:** It is an abstract data type that serves as a collection of elements with two principal operations, push to add an element on the top of the collection and pop to remove the top element of the collection.

Database model is how a database is structured and used. Database models are designed to suit specific requirements or purposes of applications and closely related to data query, algorithm, and other processing and management techniques. Accordingly, database models play the key role in data processing and management along with data models. Many database models have been suggested as well, but common ones are the followings and their visual representations are shown in Figure 6:

- **Flat:** It consists of two-dimensional array of data elements organised in relationships.
- **Star:** It consists of a few fact table and referenced dimension tables.
- **Hierarchical:** It forms a tree structure. Similar to network model, but hierarchical model does not allow arbitrary graph.
- **Network:** it organises data with records and sets. Records contain fields and sets define one-to-many relationships between the records.

- **Relational:** It is based on the first-order predicate and describes a database as a collection of predicates over a set of predicate variables.
- **Object-relational:** As a middle ground between relational databases and object-oriented databases, it is similar to a relational model, but directly support objects, classes, and inheritance.

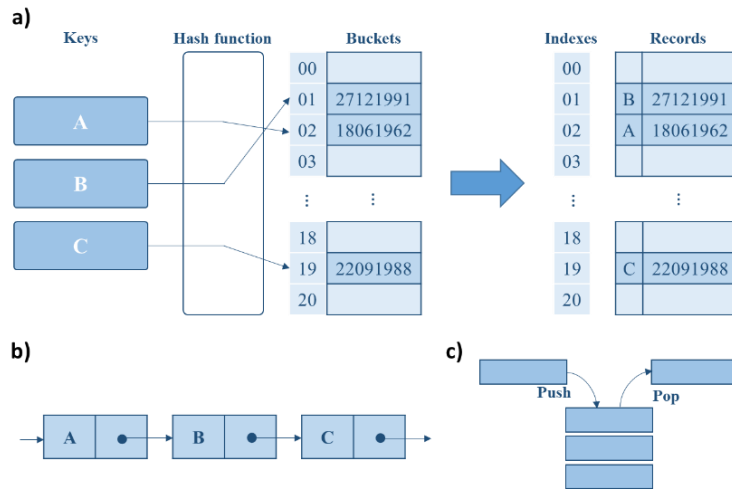


Figure 5: Visual representation of the common data structure; a) Hash table, b) Linked list, c) Stack

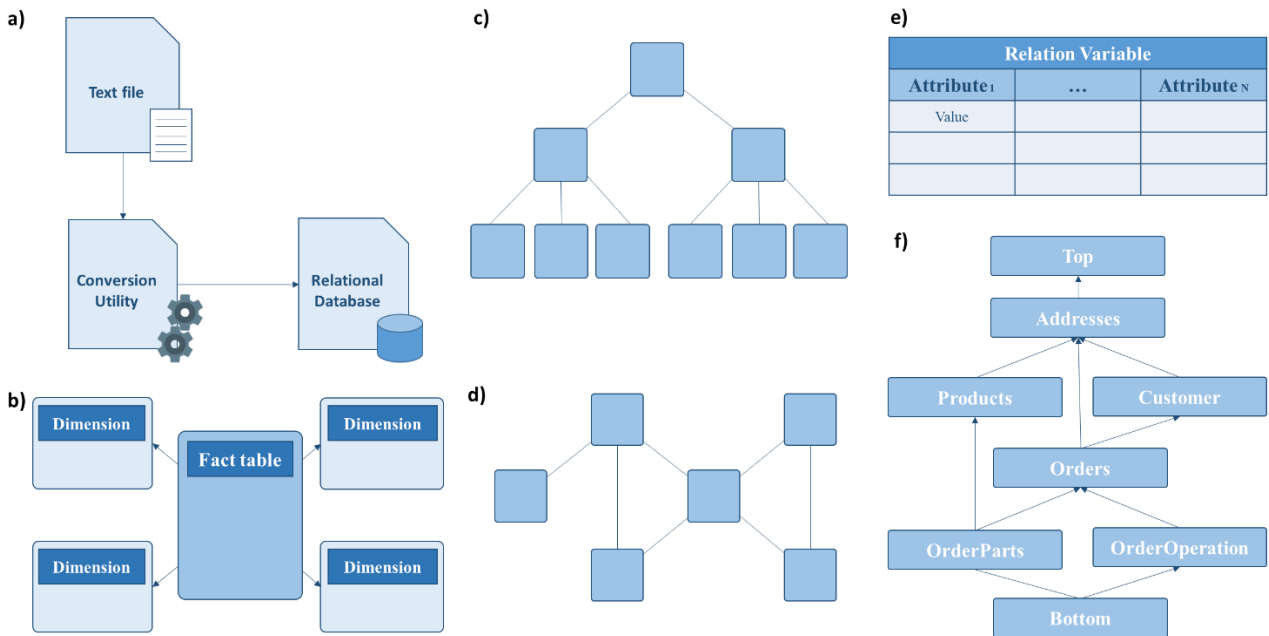


Figure 6: Visual representation of the common database models; a) Flat, b) Star, c) Hierarchical, d) Network, e) Relational, f) Concept-oriented

Data models, data structures, and database models are highly related to the data types, which also depend on applications. For an instance, in telecommunication and broadcast industry, encrypted digital data are delivered and recorded; in location-related industry, not only GPS but also two-dimensional and three-dimensional geographic data are handled along with data from transportation, agriculture, and many other fields; Internet and the Web embrace most kinds of existing data including file, image, video, documents, etc. Depending on what types of data are being handled, data models and structures have been evolved in application-dedicated directions to suit the service requirements. Therefore, there exist holes in the data ecosystem that disconnect data industries.

However, the integration and migration of data between different industries and fields are crucial to create added value to the data, especially in IoT and smart city where all kinds of data are needed together to operate. Moreover, bridging the holes between different industries are inevitable issue to solve in order to support businesses well with reduced unnecessary interface costs, which are caused by arbitrarily different data models in applications and systems. However, at this moment of time, there is not a solid solution or approach to resolve this issue with a consensus yet, and how to support interoperability between data from different industries on which environment is the issue that remains for the emerging IoT and smart city era.

9 Microdata formats for Web data management

The Web is initially developed for data exchanges and focused on rendering and visualizing data in hierarchical structures. Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript together are the triad cornerstone technologies for the Web, and they are used by most web pages for user interfaces of many web and mobile applications to create visually engaging environments. However, how the data is related and structured have been evolved and complicated, and the needs have been risen for more complicated and advanced methods to manage data. As the root of the Web, HTML was designed for visual representation of data with hierarchical structures. CSS facilitates the separation of the style from the content to manage the visual style of documents written in markup languages. JavaScript adds dynamics and interactivity as its code is written inside and sent within HTML documents to the browser to execute certain works.

However, while these triad cornerstone technologies still are the basis of the Web, they are short on the abilities to apprehend the meanings, properties, hierarchies of data and to represent or visualize the structure of data in meaningful ways. They do not have enough capability to precisely impart the meanings and relations of the information nor to intelligently visualize the data according to its semantic meanings. To complement the shortages, many approaches have been developed for representing structured data on the Web.

9.1 Structured data

The structured form refers to the structured data with a high level of organization. The structuring process of linked data saves time, reduces errors, and improves data integrity. The well-structured data and markups with link information provide better understanding of data without ambiguity. Unstructured data hinders visualization, semantics, context-awareness, and intelligent decision making based on contents. Therefore, in advance of data analytic, classification, annotation, searching, and other data processing, the data needs to be organised well in structured forms to improve processing performance and utilization. HTML was initially designed to describe presentation for organization and display of the elements on web browsers rather than semantically link or utilize the data on the Web. Hence, HTML focuses on content visual representation and often meets limitations to semantically define structures in order to attach properties and relations of data minutely and precisely. According to the HTML structure, elements are assigned only in hierarchical manners: headings, paragraphs, images, etc.; therefore, it is inadequate to minutely nor precisely describe the information and intelligently render the contents according to their semantic meanings. However, when the data on web pages are organised well in structured forms, the meanings of each element become explicit and suitable to be linked as described Figure 7.

To support structured data on the Web, many approaches are developed, such as XML markup and Resource Description Framework (RDF) metadata data model. Beyond simply linking resources with hyperlinks, these approaches enable data to be organised in structured forms and attach semantics to the Web by coupling structured data with their properties and linking resources with their relations. However, they alone are yet insufficient to fully complement the limitations of HTML to represent and describe data minutely and precisely in a unit that is small enough for machines and programs to easily parse and extract information from the resources.

These approaches focus more on how to semantically link resources rather than how to fragment bulk data into small sizes in advance of defining relations and imparting semantics to ease the processes. Therefore, the resulting syntax and data model are more likely to generalize or summarize the overall contents of the resources, not to distinguish specific information from data chunks in details to minutely attach meanings for each different part of the content. On the other hand, while organizing data in well-structured form, microdata formats break down block of data into small pieces as shown in Figure 8 to enable data items to be managed in small data units; hence, they have apparent advantages in semantically and intelligently processing, managing, and visualizing data according to meanings and relevant annotations nested in a simple way to lower the barriers for new data formats.

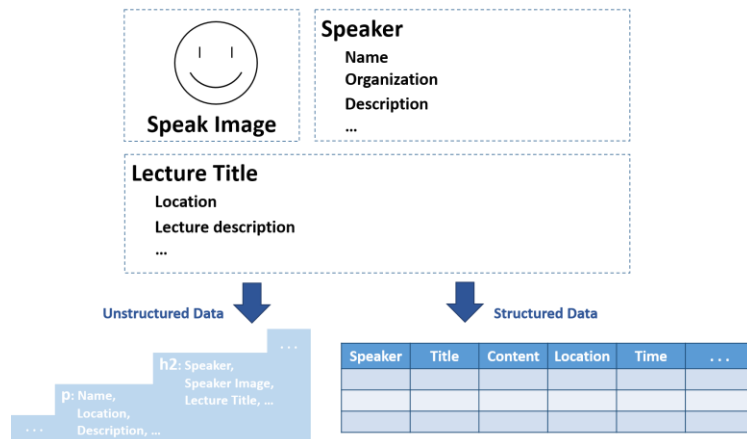


Figure 7: Hierarchical unstructured data and structured data

```
HTML:
<h2> Speaker Image </h2>
<img src = "SpeakerImage.jpeg">

<div>

  <h2>
  Speaker
  </h2>
  <p>
  Name
  Organization
  Description
  </p>

  <h2>
  Lecture Title
  </h2>
  <p>
  Location
  Lecture description
  </p>
</div>

Microdata:
<span itemscope><meta itemprop = "name" content = "Speaker Image"> </span>
<figure>
  <img src = "SpeakerImage.jpeg">
  <figcaption> Speaker Image </figcaption>
</figure>

<div itemtype="http://schema.org/Event">

  <h2 itemprop="performer">
  Speaker
  </h2>
  <div itemtype="http://schema.org/Person">
  <p>
  <span itemprop = "name"> Name </span>
  <span itemprop = "memberOf"> Organization </span>
  <span itemprop = "description"> Description </span>
  </p>
  </div>

  <h2 itemprop="name">
  Lecture Title
  </h2>
  <p>
  <span itemprop = "location"> Location </span>
  <span itemprop = "description"> Lecture description </span>
  </p>
</div>
```

Figure 8: Comparison between HTML and Microdata

9.2 Microdata format

Microdata formats, or structured data markups, describe and embed meanings of resources on the Web along with their properties and relationships. These microdata formats utilize HTML or XML tags to convey additional metadata and other attributes in web pages so that data items can be managed in small data units. Microdata formats break down paragraph or blockquote data into small pieces in structured forms, so that the contents can be more parseable and the programs can more easily extract information from it. As described in Figure 9, when the data in a resource coalesces in an unstructured form, the metadata meets the limitations to represent the content information in details; however, as structured and annotated with microdata formats, the contents can be described in much more detail manners and the meanings of the data become more apprehensible for enhanced data processing and management.

Imparting more semantic meanings to content and data, microdata formats provide elaborate means to express the contents of web pages and help machines and programs understand the detailed meanings of the web contents. Now, microdata formats are widely used and recommended to use for resource locating and retrieval and for microservices as suites of small services on the Web. Microservices is an approach to design a software architecture that builds a large complex application from small independently deployable components [b-Fowler]. For data locating and retrieval, the data needs to be extracted and its meanings should be understood by machines and programs with given metadata. Therefore, more precise metadata provided by microdata formats facilitates resource locating and retrieval on the Web. Moreover, microdata formats not only help to quickly and accurately locate the resources but also raise the opportunity for the resource usability when the contents of the resources are structured and described well enough. As shown in Figure 10, when the content of a resource is segmented into small pieces and the microdata-formatted metadata explicitly describes meaning and information of each data piece in minute details, the appropriate resources can be effectively retrieved by utilizing the given metadata, and the resources gain higher chances to be found and used. On the other hand, with metadata whose descriptions generalize the overall paragraph or blockquote data, the resource is hardly searchable nor retrievable even though the resource has the exactly identical content.

Moreover, in microservices, of which approach is to develop a single application as a suite of small services, microdata formats are used to assemble the small services together. Each small service runs individually and communicates with lightweight mechanism; therefore, these services are independently deployable as a plug and play approach. Compared to monolithic services of which application is built as a single unit, microservices have advantages in managing, updating, and scaling services with fewer overheads. With microdata formats, detailed information on each service, such as purpose, performance, characteristics, etc., can be well described, thus well integrated and visualized as a single application with high interoperability, accessibility, and readability.

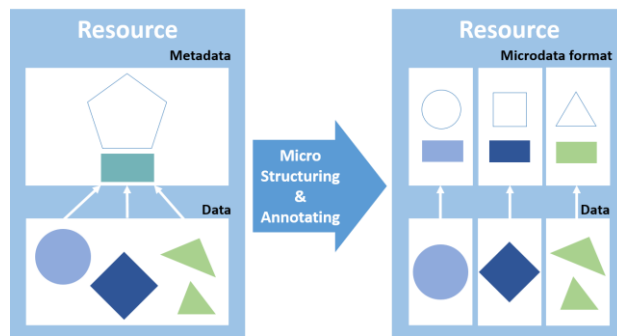


Figure 9: Structuring and micro-annotating a resource with Microdata format

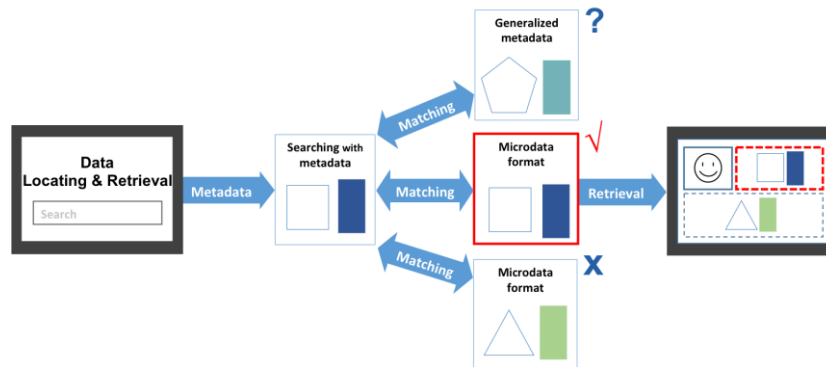


Figure 10: Use of Microdata format in search engine data retrieval

9.3 Leading Microdata formats

To structure and markup data in small pieces on the Web, several microdata format approaches have been introduced over years, and currently a couple of them are settled and pervasively used: Microdata, RDFa, and JSON-LD. Among them, Microdata suits the emerging Web environment since the simplicity and the richness are balanced well. The characteristics of each format are described in Table 3 [b-Google].

Microdata minutely attaches nested value pairs to documents along with the original contents and allows semantic and intelligent process of information intended for end-users such as contact information, geographic coordinates, calendar events, etc. Moreover, it facilitates the encoding and extraction of detailed data, which improves data query, retrieval, reusability, etc.

Maintaining the traditional RDF expressions, RDFa is one of the earliest models of microdata formats and considered to be the most comprehensive, rich standard. However, the complexity of RDFa is considered to be too much for the Web. As schema.org was launched in 2011, which provides vocabularies for structured data, Microdata has been henceforth developed [b-Wetherill].

On the one hand, JSON-LD is recently emerging as a simpler and more lightweight linked data format. It facilitates JSON data to interoperate on the Web and separates the presentation layer from contents so that it allows blocks of code can be injected easily. Since JSON-LD directly provides the separate metadata blocks, it does not require a parser to endeavour the contents to uncover and extract meanings of data like RDFa and Microdata, which use attributes of HTML tags to express data.

Table 3: Microdata formats and descriptions [b-Google]

Microdata format	Descriptions
Microdata	Microdata is a nested structured data within HTML content. It uses HTML tag attributes to name the properties of the structured data.
RDFa	RDFa is an HTML5 extension that uses HTML tag attributes that correspond to the user-visible contents to support linked data.
JSON-LD	JSON-LD is a JavaScript notation separate from the HTML body. The markup can be detached from the user-visible text and dynamically injected into the contents.

NOTE – Appendix I introduces general description of microdata formats and statistics about their deployment [b-Bizer].

However, ambivalently JSON-LD’s detachment of the metadata from contents can lead to some overheads and problems in validating, locating, and visualizing the data on web browsers. Uncoupling the metadata from the content impedes data and metadata validation and can lead to inconstancy between what the data truly is and what metadata describes it is. Moreover, the process to automatically uncouple the metadata from the contents is straightforward, but the inverse process is not. When the nested metadata get uncoupled, some detail information on the location of particular data inside the content can be lost, and it is difficult to restore the loss in coupling them back together as described in Figure 11.

Therefore, JSON-LD has an advantage in data transferring and exchanging because of its lightweight format, but it results in higher overheads to locate the particular data residing in contents and to render or visualize the data on web browsers. In other words, JSON-LD facilitates resource searching in a big database, but Microdata is more suitable to parse data residing in the searched resources in order to visualize or make a use of the data for services on the Web.

As the leading microdata formats, Microdata, RDFa, and JSON-LD are widely used and recommended to optimize the Web environment. These microdata formats convey additional metadata and other attributes in web pages, and they enable data items to be indexed, searched, stored, or cross-referenced in small data units so that the detail data can be minutely utilized. Microdata,

RDFa, and JSON-LD have each different advantage and strength over the others in a view of the trade-offs of structured data format requirements described in Figure 12.

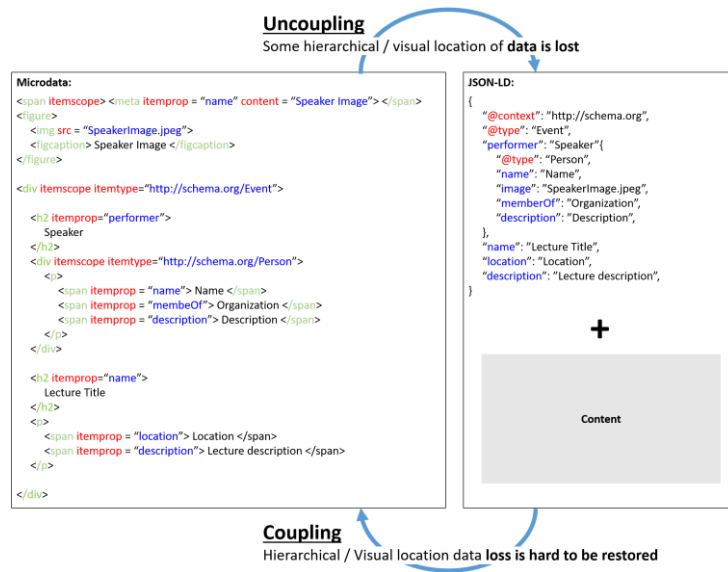


Figure 11: Uncoupling and coupling data and metadata

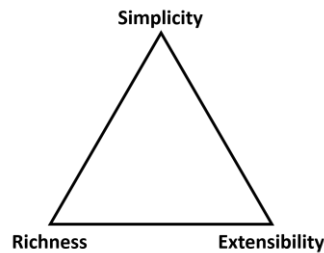


Figure 12: Trade-offs of structured data format requirements

10 Procedural metadata for semantic Web of Things

In WoT, devices and data should be properly managed and engaged by relations and procedures. Utilizing Microdata formats, the devices and data can be well-defined to be granularly managed, but the data models and ontologies with conventional metadata have shortcomings to convey common descriptions on procedures among data elements, devices, and systems. Accordingly, a new metadata category needs to be introduced, so-called procedural metadata [b-ITU journal]. The goal of procedural metadata is to provide the common descriptions on composable procedures of not only individual devices but also smart systems as a whole based on existing data models and ontologies. It's general description, primary uses, and example properties are provided in Table 4. With procedural metadata, a device or a system can make decisions and perform tasks automatically by cooperating with heterogeneous devices and systems. Since procedural metadata helps different types of data, devices, and systems to collaborate as long as they support micro- formatted metatag structures, it facilitates a high- level of interoperability in WoT. All emerging approaches that share the same concept of procedural metadata and meet the principles can be categorized as procedural metadata.

Table 4: Description, primary uses, and properties of procedural metadata

Metadata	Description	Primary Uses	Example Properties
Procedural metadata	It includes information for common descriptions on procedures and/or processes by utilizing metadata and flags (tags/indexes).	<ul style="list-style-type: none"> - Navigation - Interoperability - Procedure automation 	<ul style="list-style-type: none"> - Function - Algorithm - Work instruction - Work flow

10.1 Concept of procedural metadata

Smart systems indicate IoT environments consisting of one or more connected devices, in which applicable data models and ontologies are not restricted to one type of solution. It is assumed that the common descriptions on the data and device level are accomplished based on other previous and ongoing works. Instead, the concept of procedural metadata aims at a higher level of interoperability, with common descriptions on procedures. The procedure in procedural metadata refers to a specified way to carry out an activity or a process, which is a set of interrelated or interacting activities that use inputs to deliver an intended outputs [b-ISO 9000:2015]. In the context of IoT, the activity can be seen as not only simply processing data to compute and estimate values for properties and defining their relationships but also making decisions and performing tasks based upon them. Heterogeneous types of data, devices, and systems can engage interoperably and support automation by composing a set of procedures described in granular components of data, devices, and systems.

As a category of metadata, procedural metadata refers to the additional data that describes composable logic, functions, and workflows among data, devices, and systems. With the descriptions, the IoT data, devices, and systems can interoperate and automatically engage together to make decisions and perform tasks. To empower semantics, WoT data is often micro-formatted and structured in parsable forms so that the data can be expressed and managed with their relationships and properties in granular scales. The procedural metadata utilizes the microdata and provides the composable knowledge to make decisions and perform tasks by expressing the procedural relations among data elements, devices, and systems in granular scales. Consequently, procedural metadata facilitates data, devices, and systems to be fully or partially composed efficiently. As described in Figure 13, a set of procedural metadata can be composed as a part of data, devices, and systems for complex decision-making and performing tasks. These procedures are not necessarily defined for specific environments, applications, services, systems, devices and data, but they rather can be designed to be generally applied in diverse situations for various data from multiple domains. Emerging approaches that share the same concept, *i.e.* process description with Sensor Model Language (SensorML) or context information with NGSI with Linked Data (NGSI-LD) and Cross-Cutting Context Information Management (C3IM), can be considered to fit in procedural metadata category.

Procedural metadata can be internally and externally managed as a part of the data itself or accompanying data. As described in the Figure 14, the procedural metadata can be directly attached to the devices and data published by the devices or it can be retrieved from external source on needs. An other example in Figure 15 shows that how a set of procedural metadata can be integrated with an other approach, W3C's Thing Description (TD). W3C provides the semantic metadata and functional description of devices in events, properties, and actions [b-Charpenay], [b-Kaebisch]. In this example, procedural metadata can be externally stored and managed in a repository like TD, and the procedural metadata can be utilized on proper platforms and APIs.

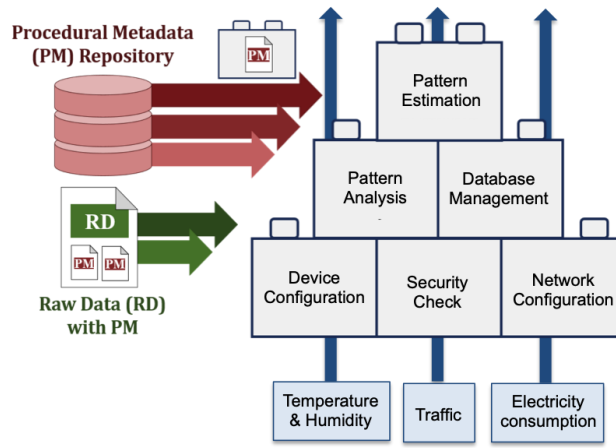


Figure 13: Example of procedural metadata composing [b-ITU journal]

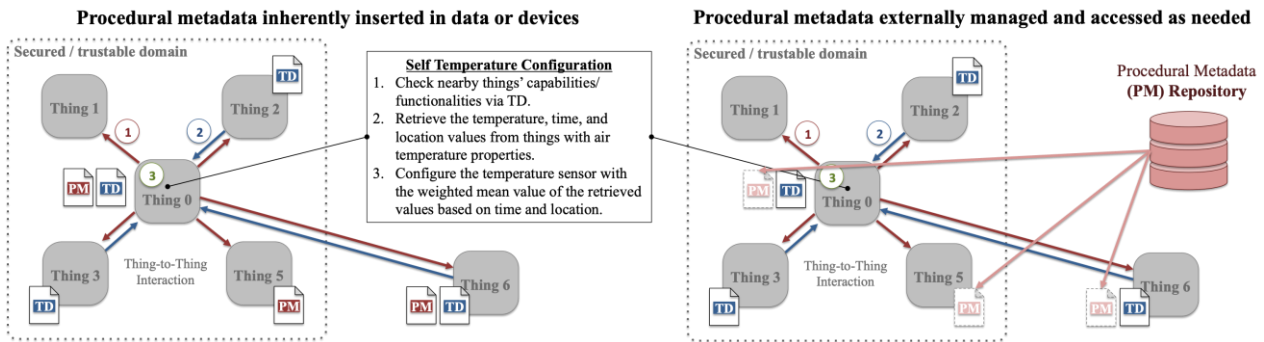


Figure 14: Example of procedural metadata embedding [b-ITU journal]

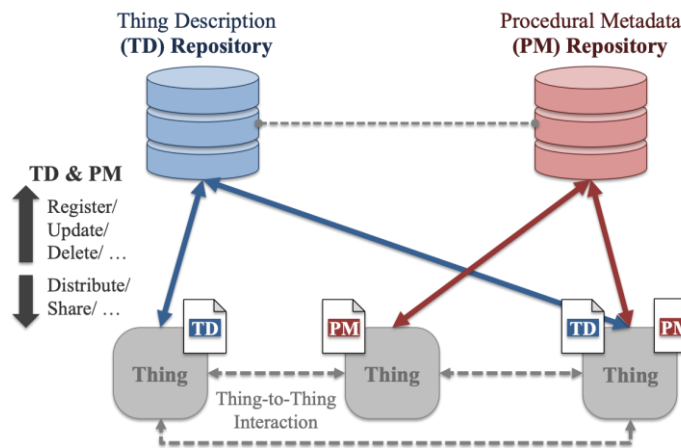


Figure 15: Applying procedural metadata in W3C's TD concept [b-ITU journal]

To achieve process automation and data interoperability of a process and a system across domains, enhanced IoT applications and services have been studied and developed. A brief summary on the other organizations' works is provided in Appendix II. Among them, SensorML and NGS-ILD share a same concept with procedural metadata.

SensorML aims to provide semantically-tied means of defining process and processing components associated with the measurement and post-measurement transformation of observations [b-OGC SensorML]. SensorML defines a process that take one or more inputs, to generates one or more outputs based on a set of parameters and a methodology. In the perspective of SensorML, a process is the same concept for a procedure in procedural metadata since it considers that a process can be a single atomic operation, or an explicitly defined network of operations. In SensorML, components

are modeled as process, which can be divided into two types: physical and computational operations. The physical processes are those of which information regarding their positions may be relevant, such as detectors, actuators, and sensor systems. The computational processes are those which can be treated as mathematical operations or functions. They both receive input and generated output, which may be digital numbers or physical stimuli, based on configurable parameters and methodologies. With a process defined in SensorML, the automation of the processes for IoT devices, systems and services are achievable.

NGSI-LD and C3IM concept enable to link disparate but related information across various domains. Context information is defined as informational representation of a set of entities with which an entity has defined relationships, together with the categories and properties of these entities, their relationships and their properties [b-ETSI CIM UC]. Through accessing, gathering and merging a set of context information from multiple domains, IoT systems, applications, and services can be enhanced with cross-domain data utilization and process automation. For example, in a smart street light usecase ETSI provides in [b-ETSI CIM UC], weather monitoring, lighting-management, traffic management, street-monitoring systems can be engaged together through a context information compliant platform. Integrating the context information offered from various systems, more enhanced and diverse street light services can be developed.

In these approaches, the defined process, context information, procedure, or whatever it is called in each approach can solely stand alone or be sequentially linked with one another to represent both simple and complex processes of functions, applications, services, and systems. This concept of interoperability and composability in procedures, processes or contexts is the intersection of SensorML, NGSI-LD, and procedural metadata. However, procedural metadata presents a slightly different approaches as it pursues to achieve the interoperability in a process itself, not specifically limited to a certain domain, device, property or attribute. This means that a single defined procedural metadata could be applied to multiple systems, devices and properties without requiring any modification.

10.2 Principle of procedural metadata

In order to not only support interoperability but also manage procedural metadata efficiently and accessibly, there should be basic principles in generation, organization, and utilization of procedural metadata. Some underlying principles of procedural metadata are provided as the followings. The data that provide additional information on procedure and comply with these principles can be considered to be procedural metadata.

- Procedural metadata is not a mandatory part of data, devices, and systems, but it is optional. [L]
[SEP]
- Procedural metadata should not affect data's original formats, models, vocabularies, ontologies, etc.
- The inputs and outputs of each procedural metadata should be clarified so that a set of procedural metadata can be composed through output-input connections. [L]
[SEP]
- The inputs and outputs of procedural metadata need to be a set of IoT resource or data and a sort of action needed to be taken, including actuating, sensing, data managing, etc. [L]
[SEP]
- As an executable script-type data, procedural metadata needs to be described based on sets of conditions and actions of IoT resources, vocabularies, relationships, and controls both when it is managed internally and externally.
- If inserted in data internally, procedural metadata needs to be marked with proper tags so that it is efficiently searchable and parsable. [L]
[SEP]
- Internal embedment in data or a device is recommended when the procedure is specific to a

- certain data, device, system, or application domain. [L]
[SEP]
- For generic usage, it is recommended to manage procedural metadata externally and in universal vocabularies. [L]
[SEP]
 - For the procedural metadata externally managed in a repository, the modification made after the first creation and its version should be specified. [L]
[SEP]
 - The maintenance policies should be articulated, especially the manner in which access, change, usage, etc. of the procedural metadata are allowed. [L]
[SEP]

Appendix I

General description of Microdata formats and deployment statistics

This appendix explains general description of some of existing microdata formats and statistics about their deployment [b-Bizer].

Table I-1: General description of microdata formats [b-Bizer]

Format	Description	Identifier
Microdata	Microdata allows nested groups of name-value pairs to be added to HTML documents, in parallel with the existing content.	html-microdata
JSON-LD	JSON-LD is a lightweight Linked Data format. It is easy for humans to read and write.	html-embeddedjsonld
RDFa	RDFa is a specification for attributes to express structured data in any markup language, e.g HTML. The underlying abstract representation is RDF, which lets publishers build their own vocabulary, extend others, and evolve their vocabulary with maximal interoperability over time.	html-rdfa
hCard Microformat	hCard is a format for representing people, companies, organizations, and places, using a 1:1 representation of vCard (RFC2426) properties and values in HTML.	html-mf-hcard
hCalendar Microformat	hCalendar is a calendaring and events format, using a 1:1 representation of standard iCalendar (RFC2445) VEVENT properties and values in HTML.	html-mf-hcard
hReview Microformat	hReview is a format suitable for embedding reviews (of products, services, businesses, events, etc.) in HTML.	html-mf-hreview
XFN Microformat	XFN (XHTML Friends Network) is a simple format to represent human relationships using hyperlinks.	html-mf-xfn
Geo Microformat	Geo a 1:1 representation of the "geo" property from the vCard standard, reusing the geo property and sub-properties as-is from the hCard microformat. It can be used to markup latitude/longitude coordinates in HTML.	html-mf-geo

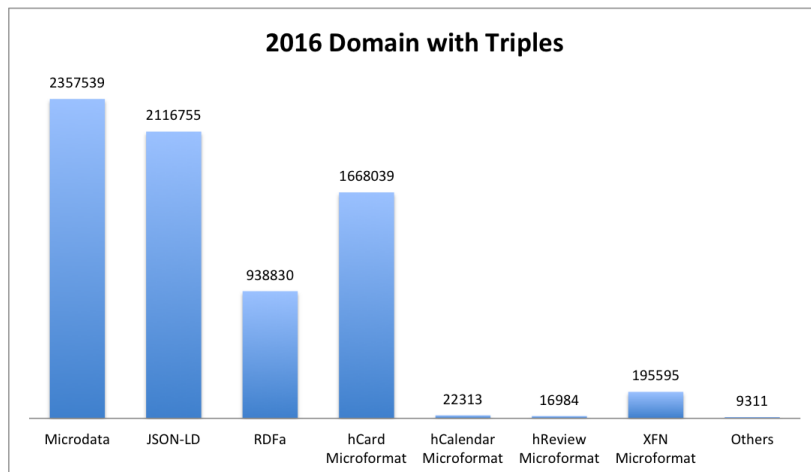


Figure I-1: Statistics about domains deployment of different microdata formats in 2016

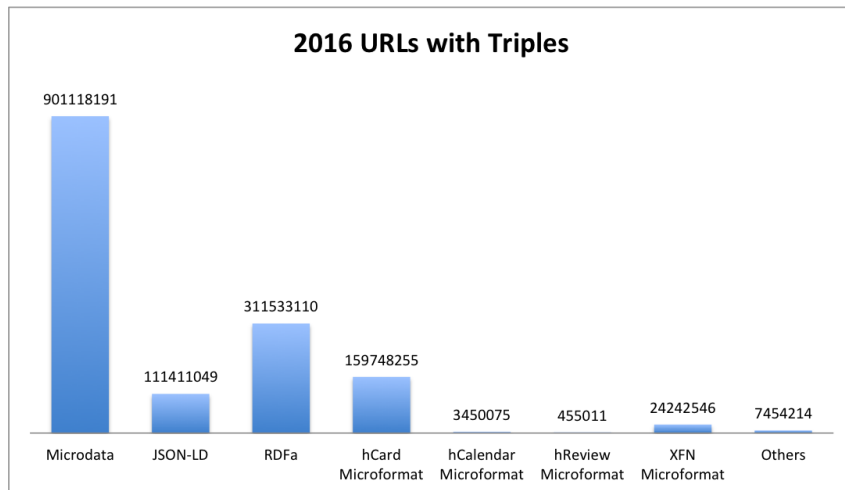


Figure I-2: Statistics about URLs deployment of different microdata formats in 2016

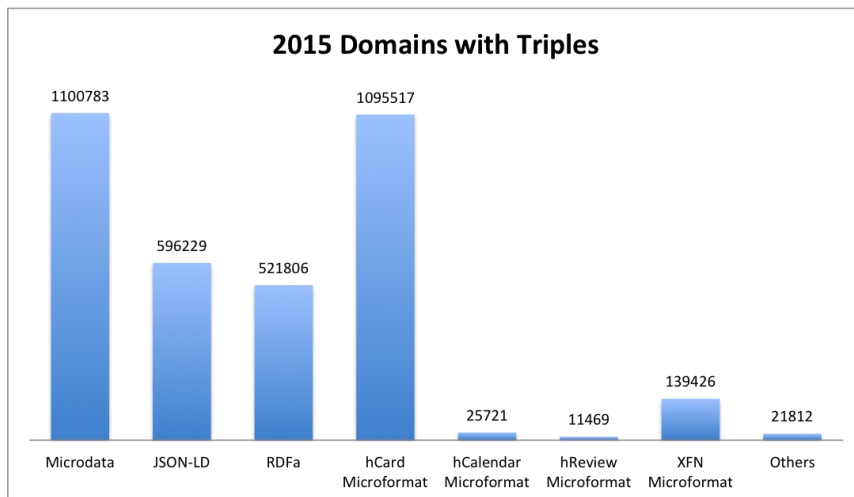


Figure I-3: Statistics about domains deployment of different microdata formats in 2015

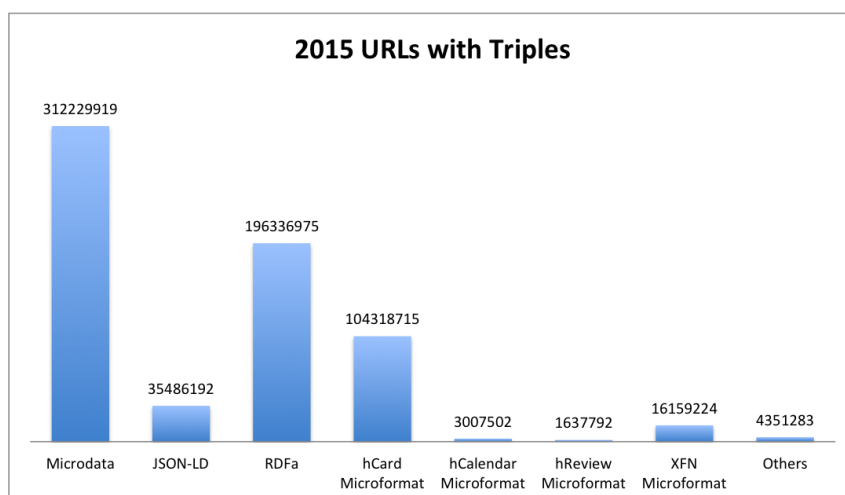


Figure I-4: Statistics about URLs deployment of different microdata formats in 2015

Appendix II

Activities for integration/interoperability in IoT and smart city

This appendix provides a brief summary of recent activities of other organizations related to integration/interoperability in IoT.

II.1 OCF

Open Connectivity Foundation (OCF) architecture defines two logical roles that devices can take, and an OCF device implements one or both roles [b-OCF].

- Server: a logical entity that exposes hosted resources, is discoverable, and responds to client initiated transitions.
- Client: a logical entity that interacts with resources on a server via discovery and actions.

The OCF device concept is composed of three components, resource, complex resource, and device [b-OCF].

- Resource is a fundamental building block, and it is the minimum interoperable component.
- Complex resource is a specific collection of resources likely to be frequently used.
- Device is a specific collection of resources composed to meet the needs of real-world requirements.

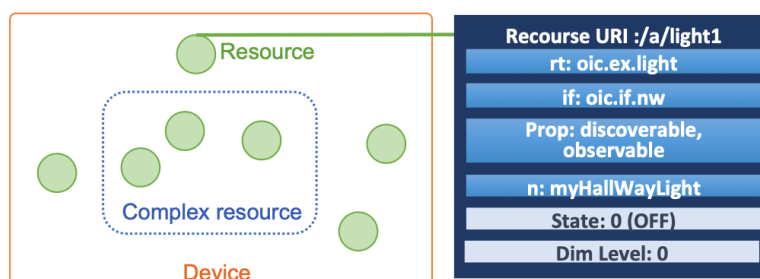


Figure II-1: OCF concept and components [b-OCF]

OCF bridging specification specifies a framework for bi-directional translation between devices in OCF and non-OCF ecosystems. An OCF Bridge is a device that represents one or more non-OCF devices as virtual OCF devices on the OCF network in order to support multiple bi-directional translation bridges.

II.1.1 IoTivity [b-OCF IoTivity]

OCF provides an implementation (IoTivity project) of the specified software stack allowing both application developers and device manufacturers to deliver interoperable products across various platforms. IoTivity is an open source software framework enabling seamless device-to-device connectivity to address the emerging needs of the IoT. It will deliver an open source reference implementation of the OCF standard specifications but not be limited to them. IoTivity assumes a full-featured device with reasonably large memory to accommodate OCF 1.3 specification features, and IoTivity-Lite is light-weight implementation of IoTivity for resource-constrained devices.

II.1.2 oneIoTa [b-OCF oneIoTa]

OCF also seeks interoperability at the data level by providing an online tool (oneIoTa Data Model Tool) that encourage the design on interoperable device data models for the IoT. oneIoTa is an open Web-based tool for interoperable device data models for the IoT. It is instantly compatible with OCF RESTful architecture. In oneIoTa, OCF data models are used as the common data model, and the conversion between OCF and other IoT ecosystems can be achieved with proposed and accepted conversion mapping.

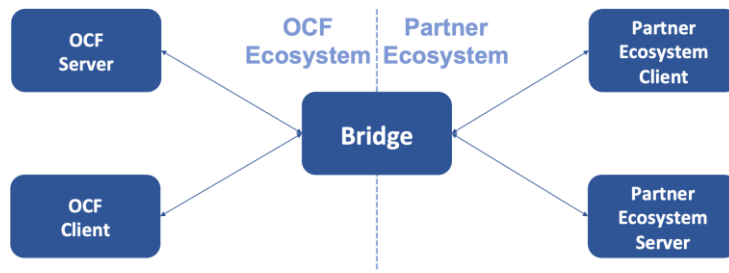


Figure II -2: OCF bridging concept – bidirectional operation [b-OCF IoTivity]

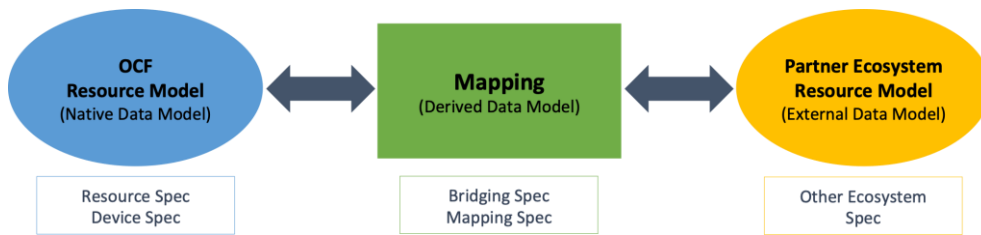


Figure II -3: OCF Bridging concept – data model [b-OCF oneIoTa]

II.2 OGC

The goal of Open Geospatial Consortium (OGC) is to support interoperable solutions that "geo-enable" the Web, wireless and location-based services and mainstream IT. OGC standards empower technology developers to make complex spatial information and services accessible and useful with all kinds of applications. OGC Sensor Web Enablement (SWE) is a suite of OGC standards designed specifically for sensors. The first version of SWE standards were published in 2005. SensorML and SensorThings API are two SWE standards relevant to IoT.

II.2.1 SensorML [b-Botts]

Sensor Model Language (SensorML) is an approved Open Geospatial Consortium standard, and its main objective is to enable interoperability by using ontologies and semantic mediation. It can be used to describe a wide range of sensors, and it provides standard models and a XML or JSON encoding for describing sensors and measurement processes. Processes described in SensorML are discoverable and executable. All processes define their inputs, outputs, parameters, and method as well as provide relevant metadata.

II.2.2 SensorThings API [b-Liang]

The OGC SensorThings API provides an open, geospatial-enabled and unified way to interconnect the Internet of Things (IoT) devices, data, and applications over the Web. At a high level, the OGC SensorThings API provides two main functions and each function is handled by the Sensing part or the Tasking part. The Sensing part provides a standard way to manage and retrieve observations and metadata from heterogeneous IoT sensor systems. The Tasking part provides a standard way for parameterizing - also called tasking - of taskable IoT devices. SensorThings API has been widely adopted and implemented around the world. For example, there are at least eight different server software implemented the SensorThings API standard, and more than 100 GitHub code repositories using SensorThings API.

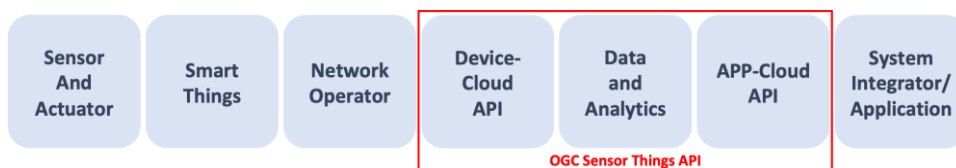


Figure II -4: Scope of OGC Sensor Things API [b-Sensorup]

II.3 W3C

World Wide Web Consortium (W3C)'s work on the Web of Data is closely coupled to the Web of Things. W3C's Web of Things Working Group has recently advanced specifications, Web of Things Architecture and Web of Things Thing Description, to Candidate Recommendation status and aims to advance them to W3C recommendations.

II.3.1 SSN [b-Compton]

Semantic Sensor Network (SSN) ontology can describe sensors in terms of the accuracy and capabilities of such sensors, measurement processes, observations and deployments. It is organised, conceptually but not physically, into ten modules, and the full ontology consists of 41 concepts and 39 object properties. Concepts and properties in the ontology are commented with rdfs.

II.3.2 TD [b-Kaebisch]

Things Description (TD) is a central building block in the W3C Web of Things and can be considered as the entry point of a Thing. A TD instance has four main components, textual metadata, a set of interaction affordances, schemas, and web links.

- Textual metadata: metadata about the Thing itself.
- Interaction affordances: information indicating how the Thing can be used.
- Schemas: schemas for the data exchanged with the Thing for machine-understandability.
- Web link: links to express any formal or informal relation to other Things or documents on the Web.

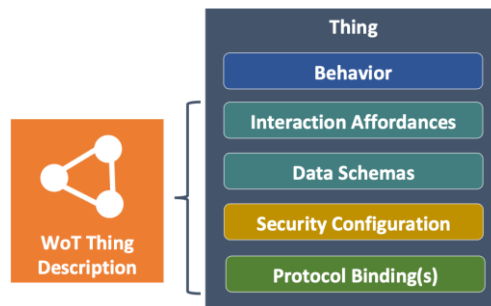


Figure II -5: Architectural aspects of a thing [b-Kaebisch]

The TD offers the possibility to add contextual definitions in some namespace. This mechanism can be used to integrate additional semantics to the content of the Thing Description instance. The contextual information can also help specify some configurations and behaviour of the underlying communication protocols.



Figure II -6: Consumer-Thing interaction [b-Kaebisch]

II.4 oneM2M

oneM2M aims to develop technical specifications addressing the need for a common M2M Service Layer that can be readily embedded within various hardware and software. An M2M services platform built upon devices, gateway, and servers. Producing standards for ICT-enabled systems, applications and services deployed across all sectors of industry and society, including smart appliances, smart metering, smart cities, smart grids, eHealth, etc.

II.4.1 oneM2M [b-oneM2M]

The different working groups in oneM2M produces specifications for a reference architecture, a messaging protocol, a data management, abstraction and semantics, and interoperability testing. oneM2M defines a horizontal architecture providing common services functions that enable applications in multiple domains, using a common framework and uniform APIs. Using these standardized APIs make it is able to abstract out the details of using underlying network technologies, underlying transport protocols and data serialization, which simplify the complex and heterogeneous connectivity choices.

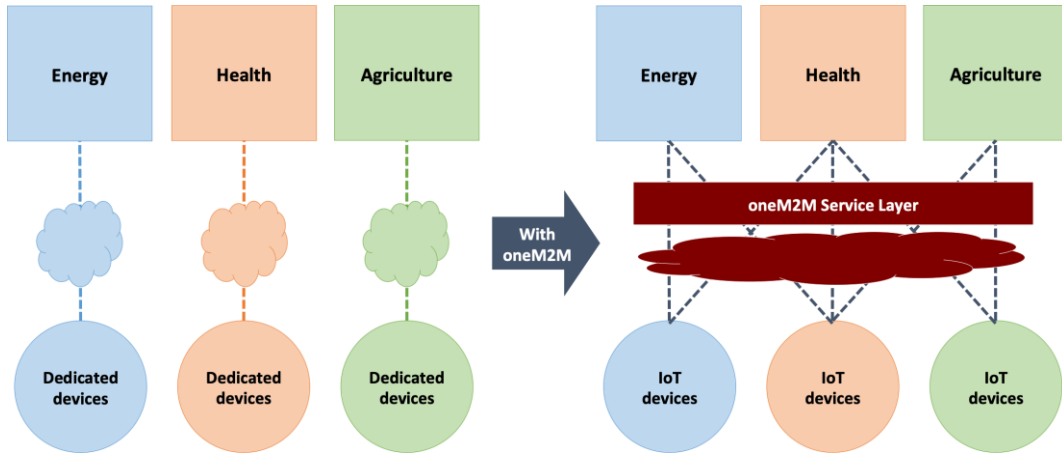


Figure II -7: IoT Cross-domain interoperability [b-oneM2M]

II.5 ETSI

European Telecommunications Standards Institute (ETSI) is a standardization organization which produces globally applicable standards for ICT-enabled systems, applications, and services deployed across all sectors of industry and society. A wide range of fields of technology, home and office, networks, transportation, connecting Thing, interoperability, public safety, security, etc., are standardized by ETSI or touched by its standardization [b-ETSI]. As it deals with diverse domains, interoperability in data, systems, and applications are one of their issues importantly studied.

II.5.1 SAREF [b-Villaon]

Smart Appliance Reference (SAREF) ontology that has been standardized by ETSI Smart M2M (Smart Appliance Common Ontology and oneM2M mapping). SAREF is a shared model of consensus that facilitates the matching of existing assets in the smart appliances domain. SAREF provides building blocks that allow separation and recombination of different parts of the ontology depending on specific needs.

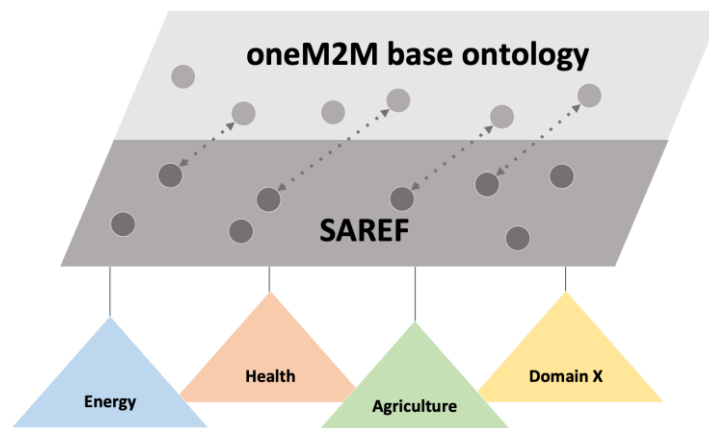


Figure II -8: SAREF ontology context [b-Villaon]

II 5.2 NGSI-LD API and C3IM [b-ETSI CIM UC], [b-ETSI NGSI-LD]

NGSI Linked Data (NGSI-LD) API is intended to not define a specific architecture and minimize architectural assumptions of API, so that it can be used in different architectural settings. Cross-

cutting Context Information Management (C3IM) can be provided by a C3IM platform, which collects data from IoT devices, other platforms, data base, and other sources and provides context information to applications with a NGS-LD API. The main categories of context information stakeholders are producer, consumer, and broker, and the context information can be managed in centralized, distributed, and federated architecture. Enabling the links between disparate but related information, it supports IoT applications to access a rich set of cross-cutting context information so that IoT services can be enriched. As shown in a smart light usecase provided by ETSI, on a CIM-compliant platform different system can share, exchange, and trade their context information through NGS-LD to offer more enhanced, comprehensive and diverse services.

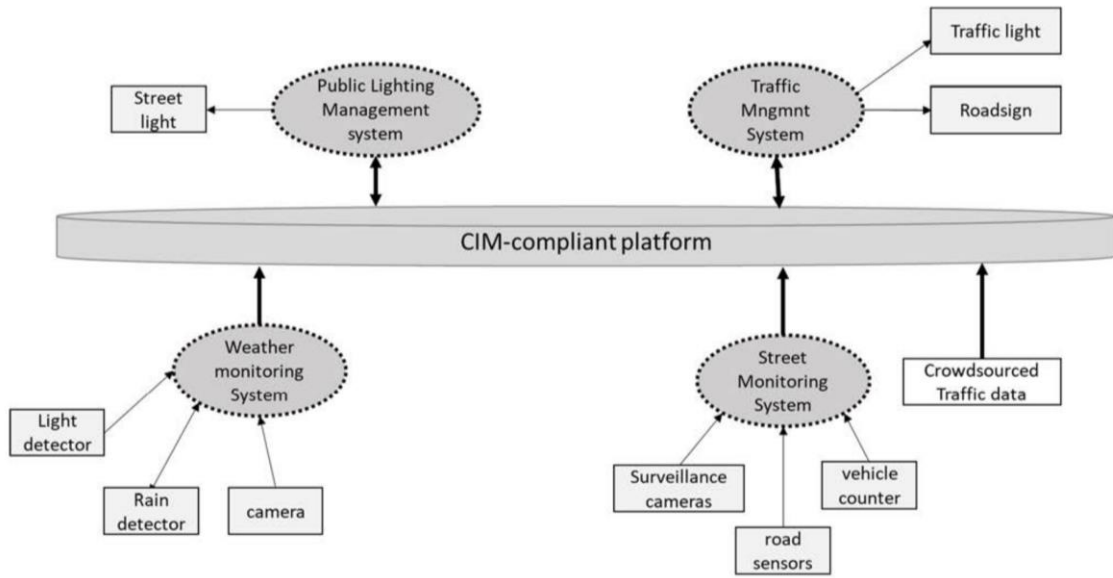


Figure II -9: Context information flow diagram in smart street lighting use case [b-ETSI CIM UC]

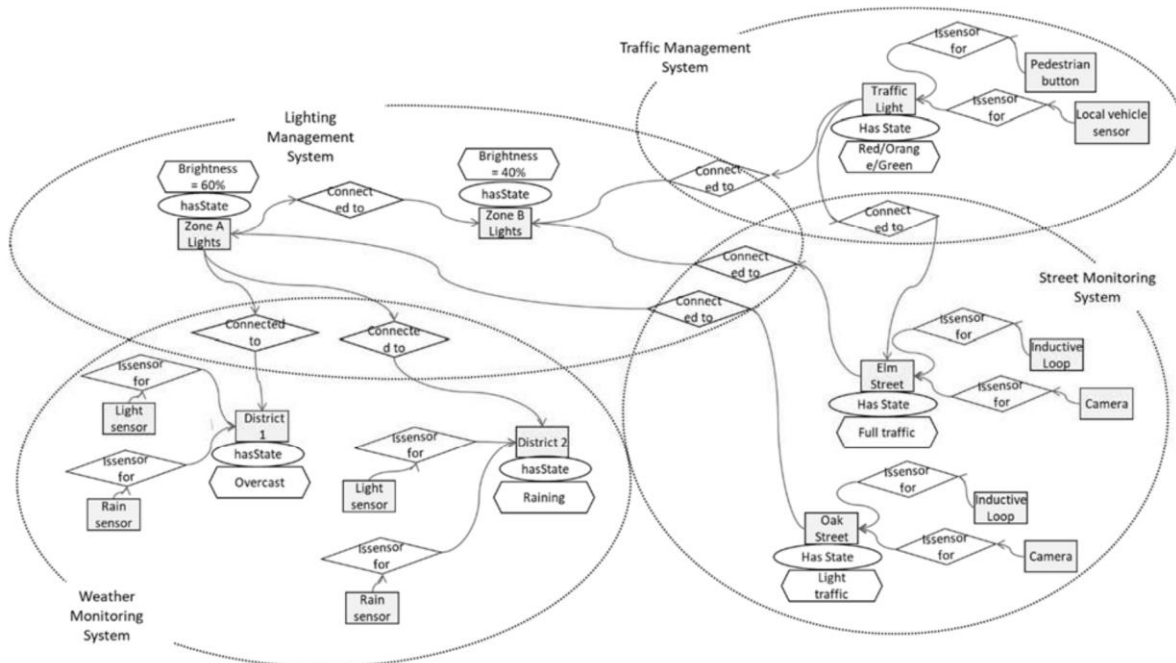


Figure II -10: Smart street lighting use case instance graph example [b-ETSI CIM UC]

Bibliography

- [b-FG-DPM TS D0.1] ITU FGDPM (2019), ‘Data Processing and Management for IoT and Smart Cities and Communities: Vocabulary’ (D0.1)
- [b-ISO 9000:2015] ISO (2015), ISO:9000 ‘Quality management systems – Fundamentals and vocabulary’
- [b-Forbes 1] P. Rotella, Forbes (2012). *Is Data The New Oil?*. Available: <https://www.forbes.com/sites/perryrotella/2012/04/02/is-data-the-new-oil/#4eb6c2d97db3>
- [b-HBR] James Wilder, Harvard Business Review (HBR) (2016). *Breaking Down Data Silos*. Available : <https://hbr.org/2016/12/breaking-down-data-silos>
- [b-Forbes 2] Gil Press, Forbes (2016). *Cleaning Big Data : Most Time-Consuming, Least Enjoyable Data Science Task, Survey says*. Available : <https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/>
- [b-Ahlgren] B. Ahlgren et al., ‘Internet of Things for Smart Cities: Interoperability and Open Data,’ IEEE Internet Computing, vol. 20, no. 6, pp. 52-56, Dec. 2016.
- [b-Corici] A. Corici et al., "Towards programmable and scalable IoT infrastructures for smart cities," 2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), March 2016.
- [b-Broring] Arne Broring et al., ‘Enabling IoT Ecosystems through Platform Interoperability,’ IEEE Software, vol.34, no. 1, pp. 54-61, Jan. 2017.
- [b-ITU-T TP] ITU-T Technical Paper (2016), ‘Analysis of Digital Data Technologies Toward Future Data Eco-Society.’
- [b-Riley] J. Riley (2017). Understanding Metadata: What is Metadata, and What is it For?: A Primer. National Information Standards Organization (NISO). Jan. 2017.
- [b-Raggett] D. Raggett, (2015). “The Web of Things: Challenges and Opportunities. Computer,” vol. 48, no. 5, pp. 26-32, May 2015.
- [b-Kajimoto] K. Kajimoto, M. Kovastsch, U. Davuluru, (2017). *Web of Things (WoT) Architecture*. W3C. <https://www.w3.org/TR/2017/WD-wot-architecture-20170914/>
- [b-Kamilaris] A. Kamilaris, S. Yunusak, M. I. Ali, (2016). “WOTS2E: A Search Engine for a Semantic Web of Things. World Forum on Internet of Things (WF-IoT),” 12-14 Dec. 2016. ^[1]_[SEP]
- [b-Scioscia] F. Scioscia, M. Ruta (2009). Building a Semantic Web of Things: issues and perspectives in information compression. International Conference on Semantic Computing (ICSC). 14-16 Sept. 2009. ^[1]_[SEP]
- [b-Miorandi] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, (2012). “Internet of Things: Vision, applications and research challenges. Ad Hoc Networks,” vol. 10, no. 7, pp. 1497–1516, 2012.
- [b-Dataversirty 1] Donna Burbank and Charles Roe, Dataversity (2016). *Emerging Trends in Metadata Management : A DATAVERISTY 2016 Report on the Top Business & Technical Drivers for Metadata*

- [b-Dataversirty 2] Donna Burbank and Charles Roe, Dataversity (2017). *Trends in Data Architecture : A DATAVERISTY 2016 Report*
- [b-Fowler] J. Lewis, M. Fowler (2014), *Microservices*, Available: <https://martinfowler.com/articles/microservices.html#footnote-etymology>
- [b-EPISTLE] Matthew West and Julian Fowler, The European Process Industries STEP Technical Liaison Executive (EPISTLE) (1999). *Developing High Quality Data Models*. Available : <https://sites.google.com/site/drmatthewwest/publications/princ03.pdf?attredirects=0>
- [b-ANSI] American National Standards Institute (ANSI) (1975). *ANSI/X3/SPARC Study Group on Data Base Management Systems; Interim Report*. FDT (Bulletin of ACM SIGMOD) 7:2.
- [b-DataWarehouse] Data Warehouse. *Data Warehousing Concepts*. Available : <https://www.1keydata.com/datawarehousing/concepts.html>
- [b-Google] Google, (2017, July). *Introduction to structured data*, Available: <https://developers.google.com/search/docs/guides/intro-structured-data>
- [b-Wetherill] B. Wetherill, “*RDFa and Microdata*,” *Library Philosophy and Practice* (e-journal), 2014
- [b-Bizer] C. Bizer, R. Meusel, and A. Primpeli, (2017, Jan.). *Web Data Commons – RDFa, Microdata, and Microformat Data Sets*, Available: <http://webdatacommons.org/structureddata>
- [b-ITU journal] N. Kim, H. Oh, J. Choi, (2018). “Procedural Metadata: Structured Guide for Data Interoperability in Support of Web of Things,” *ITU Journal: ICT Discoveries*, Special Issue No,2, Nov. 2018
- [b-Charpenay] V. Charpenay, J. Hund, S. Kabisch, T. Kamiya, M. Koster, (2018). *WoT Current Practices*. W3C. Available: <http://w3c.github.io/wot/current-practices/wot-practices.html>
- [b-Kaebisch] S. Kaebisch, T. Kamiya, M. McCool, V. Charpenay (2019). *Web of Things (WoT) Thing Description*. W3C. Available : <https://www.w3.org/TR/wot-thing-description/>
- [b-OGC SensorML] OGC (2014), ‘SensorML: Model and XML Encoding Standard’
- [b-ETSI CIM UC] ETSI GR CIM 002 V1.1.1 (2018), Context Information Management (CIM); Use Cases (UC)
- [b-OCF] OCF (2017), OCF Specification Introduction and Overview, Nov. 2017, Available:https://openconnectivity.org/specs/OCF_2.0_Specification_Overview.pdf
- [b-OCF IoTivity] IoTivity, *IoTivity About*, Available: <https://iotivity.org/about>
- [b-OCF oneIoTa] OCF, *ONEIOTA DATA MODEL TOOL*, Available: <https://openconnectivity.org/developer/oneiota-data-model-tool>
- [b-Botts] Botts, M., Robin, A., Greenwood, J., & Wesloh, D. (2014). ‘OGC SensorML v2.0: Model and XML Encoding Standard. OGC Implementation Standard (ogc 12-000).’
- [b-Liang] Liang, Steve, Chih-Yuan Huang, and Tania Khalafbeigi. (2016) ‘OGC SensorThings API-Part 1: Sensing. OGC Implementation Standard (ogc 15-078r6)’

- [b-Sensorup] S. Liang (2016), 'Open IoT Made Easy OGC SensorThings API,' Available: <https://www.slideshare.net/steve.liang/open-iot-made-easy-introduction-to-ogc-sensorthings-api>
- [b-Compton] M. Compton et al, "The SSN ontology of the W3C semantic sensor network incubator group, Journal of Web Semantics," Dec. 2012
- [b-Kovatsch] M. Kovatch et al (2019), *Web of Things (WoT) Architecture*, May 2019, Available: <https://www.w3.org/TR/wot-architecture/>
- [oneM2M] oneM2M, 'TR-0057 Service Layer,' Available: <http://www.onem2m.org/getting-started/onem2m-overview>
- [b-ETSI] ETSI, *Technologies*, Available: <https://www.etsi.org/technologies>
- [b-Villaon] M. Villalon et. al (2018). 'Extending the SAREF ontology for building devices and topology,' June 2018, Available: http://linkedbuildingdata.net/ldac2018/files/Presentations/20180620_mpove_da_SAREF.pdf
- [b-ETSI NGSI-LD] ETSI GS CIM 009 V1.1.1 (2019), 'Context Information Management (CIM); NGSI-LD API'
-