BIG IoT – Bridging the Interoperability Gap of the Internet of Things

Deliverable 4.2.a

Semantic Model for the Application Domain - first release

Date: 31.12.2016

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688038.

Responsible Person and Affiliation	Dr. Achille Zappa (NUIG)
Due Date / Delivery Date	31 December 2016
Author(s):	Darko Anicic (Siemens), Aparna Saisree Thuluva (Siemens), Victor Charpenay (Siemens), Yong Wang (TU Clausthal), Arne Bröring (Siemens), Hoan Quoc (NUIG), Achille Zappa (NUIG),
Reviewer(s):	Jelena Mitic (Siemens) Luca Gioppo (CSI)
State	Final
Version	1.0
Confidentiality	Public

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Meaning
AC	Alternating Current (Here: Slow Charging Stations)
API	Application Programming Interface
APM	Advanced Parking Management
BCN	Barcelona (pilot)
BIG IoT	Bridging the Interoperability Gap of the Internet of Things
CoAP	Constrained Application Protocol
CSI	Project Partner Piedmont, Italy
CSP	Project Partner Piedmont, Italy
DC	Direct Current (Here: Fast Charging Stations)
DOA	Description of Action
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
ICE	Inter City Express (German High Speed Train)
IERC	IoT European Research Cluster
IoT	Internet of Things
IoT-A	Internet of Things - Architecture
ISO	International Organization for Standardization
M2M	Machine to Machine
MQTT	Message Queuing Telemetry Transport
NG	Northern Germany (pilot)
OD	Offering Description
OGC	Open Geospatial Consortium
PIE	Piedmont (pilot)
RDF	Resource Description Framework
RDFa	Resource Description Framework in Attributes
REST	Representational State Transfer
SDK	Software Development Kit
SSN	Semantic Sensor Network
TD	W3c Things Description
TIC	Traffic Information Center
TUC	TU Clausthal Partner Northern Germany
UI	User Interface
UPC	Project Partner Barcelona, Spain

URI	Uniform Resource Identifier
VMZ	Project Partner Berlin, Northern Germany
Wi-Fi	Wireless Local Area Network
WLAN	Wireless Local Area Network
WMS	Web Map Service
WP	Work Package

BIG IoT Terminologies and Definitions

Terms	Definitions
Accounting	Accounting collects data about each access to an Offering and relates it to the respective Subscription.
BIG IoT API	A set of specifications for Providers and Consumers to interact with the BIG IoT Marketplace to authenticate, register, discover and subscribe to Offerings; and perform accounting
Consumers to directly access the Resources offered by a Provider	The BIG IoT API defines the supported communication protocols, data formats, semantic descriptions, etc. In order to facilitate BIG IoT Applications, Services and Platforms to implement and use the BIG IoT API, dedicated Provider and Consumer Libs (SDKs) are provided for various platforms and programming languages, offering also programming interfaces to developers.
BIG IoT Application (or short Application)	An application software that uses the BIG IoT API to discover Offerings on the BIG IoT Marketplace, subscribe to Offerings and access the offered Resources. A BIG IoT Application acts merely as an Offering Consumer.
BIG IoT Application / Service / Platform Developer (or short BIG IoT Developer)	A software developer that implements or integrates a BIG IoT Service, Application or Platform.
BIG IoT Application / Service / Platform / Marketplace Provider or Operator	The organization that operates a BIG IoT Application, Service, Platform, Marketplace instance. It is hereby not relevant if a particular instance is hosted on the provider organization's own infrastructure or a 3rd party infrastructure.
BIG IoT Core Developer	A software developer that implements or extends BIG IoT Marketplace and/or BIG IoT Lib components.
BIG IoT enabled Plat-	An IoT Platform (or Smart Object Platform) that implements and

form (or short BIG IoT Platform or just Plat-form)	uses the BIG IoT API to register Offerings on the BIG IoT Market-place and provide access to the offered Resources. A BIG IoT Platform acts merely as an Offering Provider.
BIG IoT Marketplace	The BIG IoT Marketplace allows Providers to register their Offer-ings (based on semantic descriptions) and Consumers to discover relevant Offerings (based on semantic queries) at run-time. It also provides accounting support for Consumers and Providers to track the amount of resources accessed, as well as a web portal for de-velopers and administrators to support the implementation and management of their Applications, Services, and Platforms.
BIG IoT Service (or short Service)	A BIG IoT Service implements and uses the BIG IoT API to con-sume and/or provide Offerings via the BIG IoT Marketplace. A BIG IoT Service can act both as an Offering Consumer and Provider. It typically consumes basic Information or Function in order to offer "higher-value" Information or Functions on the BIG IoT Market-place.
BIG IoT User	A User of a BIG IoT Application. A BIG IoT User is typically an em-ployee of an Enterprise, SME or Organization (e.g. City Authority), but not limited to that.
Billing	Billing collects Charging data and creates invoices.
Charging	Charging is based on the collected Accounting data. The Charging Service multiplies the accounting data with the respective Price data of an Offering, and also takes into account special Consumer (group) accounting models, to compute the final amount to be charged.
Device-level BIG IoT enabled IoT Platform (= Device-level BIG IoT Platform or just Device-level Platform)	A BIG IoT enabled Platform that is implemented directly on a Smart Object, as opposed to on a backend or cloud infrastructure.
Endpoint	An Endpoint in the context of BIG IoT is a web based interface for consumers to access Offerings via a Provider. An Endpoint de-scription consists of properties like Endpoint type and URI.
Function	Functionality that can be invoked by Consumers and is provided by a task on an actuator (as part of an IoT Platform) a Service that provides some computational functions or higher level functionality delegating to one or more lower level Functions
Information	Data provided to Consumers by a sensor (as part of an IoT Platform) a Service that takes one or more Information sources and com-bines them to provide some added value
IoT Service (or short Service)	Software component enabling interaction with resources through a well-defined interface in order to access or manipulate information

	or to control entities. An IoT Service can be orchestrated together with non-IoT services (e.g., enterprise services). Interaction with the service is done via the network. (based on [IoT-A])
IoT Platform (= Smart Object Platform)	A computing and communication system that hosts software components enabling interaction with Smart Objects in order to access or manipulate information or to control them. An IoT Platform may be implemented on a backend or cloud infrastructure, or directly on a Smart Object. Interaction with the platform is done via the network.
License	The Provider of an Offering can choose the License terms for the provided Information.
Offering	BIG IoT enables Providers to offer or trade access to Information and Functions with Consumers via the Marketplace. An Offering is defined by an Offering description, which describes a set of Resources offered on the Marketplace. It typically encompasses a set of related Information or Functions. An Offering description provides a semantic description of the Resource(s) provided to a Consumer once the Offering is accessed. The description also entails context and meta information about the Offering, including information like the Region (e.g. a city or spatial extent) where the Resource(s) relate to, the Price for accessing the Resource(s), the License of the Information provided, input & output data fields, etc.
Offering Consumer (or short Consumer)	A BIG IoT Application or Service that is interested to discover and access IoT resources in order to provide a new service or function. A Consumer discovers and subscribes to relevant Offerings via the BIG IoT Marketplace, and accesses the offered resources via the BIG IoT API.
Offering Provider (or short Provider)	A BIG IoT Platform or Service that wants to offer or trade IoT resources via the BIG IoT Marketplace. A Provider registers its Offering(s) on the BIG IoT Marketplace, and provides access to the offered resources via the BIG IoT API.
Offering Query (or short Query)	Consumers are able to discover offerings of interest on the marketplace by providing an (Offering) Query. A Query describes the properties of Offerings a client is interested in (Offering type, input & output data fields, Price, License, ...)
Physical Entity	Any physical object that is relevant from a user or application perspective. [IoT-A]
Price	The Provider of an Offering can choose the accounting model (e.g. Free or Per Month or Per Access) and amount of money (if applicable) a Consumer has to pay when accessing a Resource.
Resource	Abstraction for either Information or Function.
Smart Object (= Thing)	A Device able to compute and communicate information about itself or related artifacts (Physical Entities) to other devices or

	computer applications; a Smart Object is typically attached to or embedded inside a Physical Entity. Smart Objects either monitor a Physical Entity (sensing) or interact with the physical world through actuators (actuation). Those functions can be either controlled autonomously by local computations or triggered from remote.
Subscription	Agreement to access the Resource(s) of a single Offering. This comprises: a Consumer's willingness to access the Offering (he checked License, service level, rating, description, ...) the Consumer's consent to pay for the access to the Resources (according to the specified Price), if applicable
Semantic Offerings Composition Recipe (or BIG IoT Recipe)	Semantic Offerings Composition Recipe or shorter Recipe provides description of the composition of offerings. It is a specification of requirements of an added value service, and hence it represents a template that can be fulfilled by multiple offerings.
BIG IoT Semantic Core Model	BIG IoT Semantic Core Model specifies all important domain concepts in BIG IoT project including all basic conceptual entities and their relationships. This semantic model is used as basis for (1) the Offering Description to define the capabilities of offerings provided by IoT platforms or services, and (2) the underlying data model of the BIG IoT Marketplace.
Mobility Domain Model	Mobility Domain Model defines a common terms used in the mobility domain. The model aims to improve information exchange and data interoperability between mobility systems, in particular used in Internet of Things applications. This model represents an extended schema.org vocabulary for the mobility domain (mobility.schema.org).
BIG IoT Semantic Application Domain Model	BIG IoT Semantic Application Domain Model defines an application-specific vocabulary that is built on both, BIG IoT Semantic Core Model and Mobility Domain Model, and can be used for annotating Offering Descriptions.
Semantic Recipe Model	Semantic Recipe Model defines a model for BIG IoT Recipes. A Recipe is a specification of requirements of an added value service, and hence it represents a template that can be fulfilled by multiple offerings. All terms and their relations, required for specifying Recipes, are defined in Semantic Recipe Model.

1. Introduction

1.1. Scope of this Document

In this deliverable, the semantic model for the application domain to be referenced in the descriptions of offerings of services and platforms will be defined. This BIG IoT Semantic Application Domain Model defines the terms of the vocabulary needed for the realization of use cases and hence there is a strong interaction with WP5. The task extends existing recommended standards, vocabularies, and ontologies. In particular, existing vocabularies in the application domains of mobility and smart cities are studied and reused if appropriate. Based on the reused works, a coherent and consolidated vocabulary will be promoted for the usage by BIG IoT services and platforms.

To be easy to use and to achieve wide adoption, this consolidated application domain model needs to be lightweight, i.e., it is based on a narrow vocabulary with simple interrelations. Together with the work in Task 3.2 on semantic descriptions of smart object platforms and services, this semantic model for the application domain will achieve semantic interoperability and provides the fundamental semantic-based contexts for service discovery and automated service orchestration (Task 4.3). The model enables a composer of offerings not only to understand the data formats and interfaces of the constituent services but also to aggregate them using semantic-based service composition rules. This can further ensure semantic correctness on the application level.

During this task, example service descriptions will be created, showing all major aspects of the semantic model of a service and will be used as starting point for use case services to be realized in WP5.

1.2. Executive Summary

Due to its diverse set of users (and use cases), BIG IoT project has to cover a wide area of domain models. It would have to specify how basic IoT sensors and sensor measurements are represented. It would have to define how to describe BIG IoT platforms and BIG IoT services. It has to identify data models for Smart City applications, e.g. for smart transport systems, smart parking. And it has to specify metadata for the BIG IoT system and its components, e.g. to model security and accounting information.

In this document we start from existing ontologies and domain models that can be used as the basis of the BIG IoT domain models and extend them as needed by requirements of BIG IoT project. We subdivide the work into different areas – Traffic, Parking, Environment, and BIG IoT systems and services. For each of these areas we discuss which vocabularies are used for modelling the required data, present necessary additional data items and how to model them, and show examples of the resulting descriptions and how to create them. The presented data models are not final and/or static. Instead, in the time until the second version of this deliverable (D4.2.b in M21) we will fine-tune the data models, incorporate more practical experiences and new requirements, and integrate new data items that we identified during the project lifetime.

The document is intended for: (a) users of the BIG IoT marketplace that want to learn about the used semantic models to develop applications, (b) BIG IoT platform providers that want to learn how to integrate their platform with BIG IoT, (c) the partners of the BIG IoT consortium, allowing them to develop their components in a way that ensures easy integration and interoperability, (d) external researchers and developers that want to design and/or use domain models in Smart City systems beyond BIG IoT, as well as (e) the project reviewers to better understand the work done in the project.

1.3. Relation to other Deliverables

The tasks of WP3 and WP4 are highly interrelated; however, their deliverables have each their own, clear responsibilities. The figure below follows the general BIG IoT architecture description in D2.4 and illustrates at hand of this architecture the scope of each of the different deliverables of the tasks 3.1, 3.2, 4.1, 4.2, and 4.3.

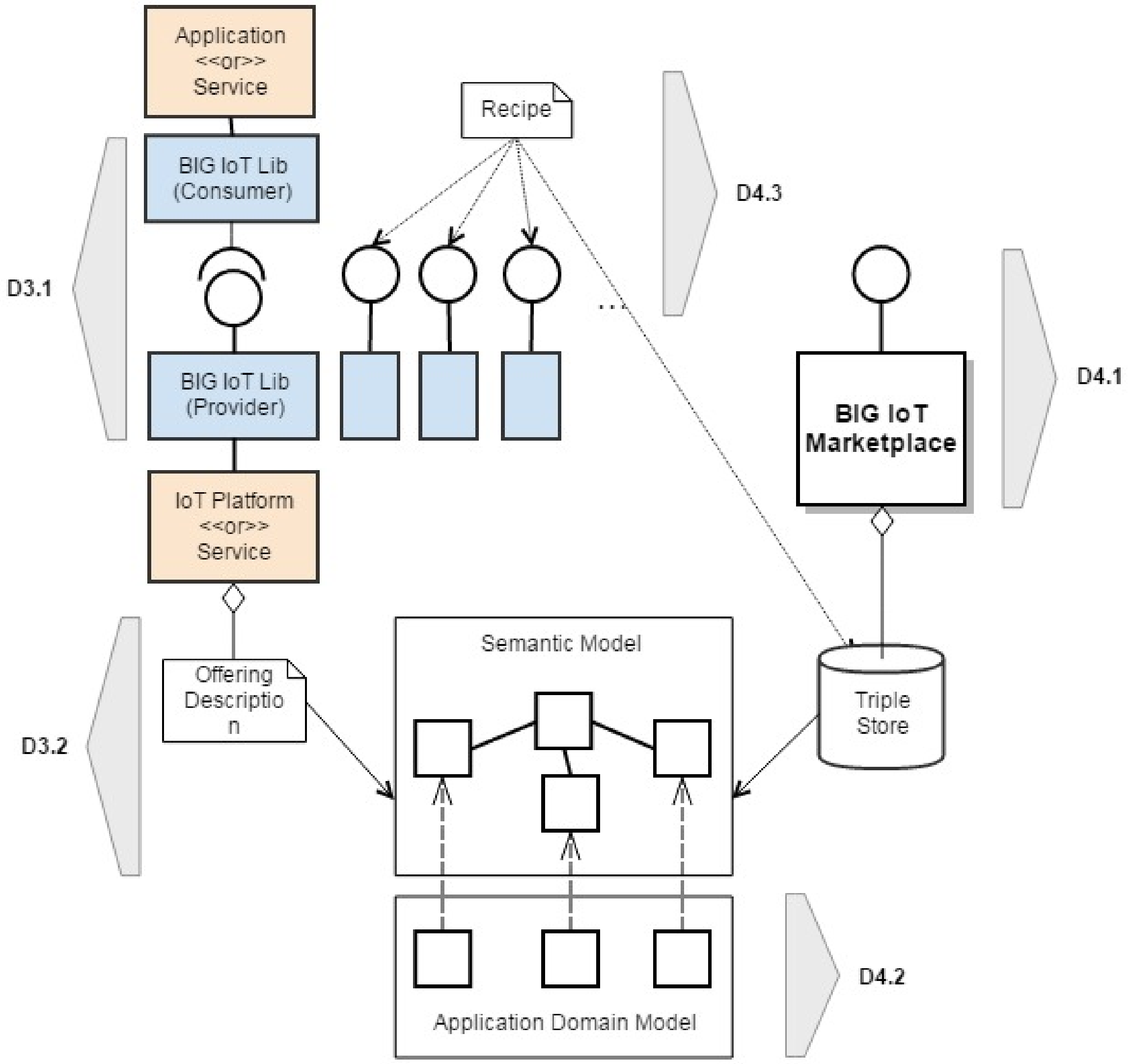

Figure 1: Relation of deliverables.

D3.1 covers the description of the BIG IoT API. This entails a specification of the Web API (e.g., interaction model, and encodings) as well as description of the programmatic API that is implemented as part of consumer and provider lib of the SDK.

D4.1 describes the architecture of the BIG IoT Marketplace. This includes the general design, the workflows of interactions, the GraphQL-based API of the market-place, as well as the user interface of the portal to utilize the marketplace. Also con-tained in this deliverable is a description on how to map between the GraphQL API of the marketplace frontend and its SPARQL based triple store backend.

D3.2 describes the model and schema of the core semantic model of BIG IoT. This semantic model is used as a basis for (1) the Offering Description to define the capabilities of offerings provided by IoT platforms or services, and (2) the underlying data model of the BIG IoT Marketplace.

D4.2 builds up on the core semantic model of D3.2 and defines the application domain model, which specifies a vocabulary of terms that can be used in Offering Descriptions, and in the triple store of the marketplace.

D4.3 addresses the orchestration and composition of the Offerings of BIG IoT Providers (i.e., platforms or services). A composition of offerings can be specified by defining a Recipe. The semantic model and examples of such recipes is the scope of this deliverable.

1.4. Structure of this Document

The document is structured as follows. After an introduction about the delivera-ble in this section the Section 2 introduces firstly the different ways to use vocabulary models (ontology) in the project, analyzes the existing domain independent models and domain dependent models and the models for non-functional properties of offer-ings. Section 3 describes from a high-level point of view the coherent and consolidat-ed application domain model. This model comprises the vocabularies usable by the realization of the use cases in WP5. Hence, it is highly influenced through the analysis of platforms and services to be contributed to the use cases. Section 4 describes the implementation of the models defined in Section 3. This implementation is done by aligning them with and extending schema.org. Section 5 presents the completed semantic description of BIG IoT offerings and services, based on the Offering Description model in D3.2.a and the application domain model defined in Section 4. Finally, in Section 6 we present our conclusions and indications for future work.

2. Background and Related Work

The main goals of BIG IoT is to realize the interoperability among the offering providers and BIG IoT services through the semantic representation of Offerings of Providers. To achieve this goal, we need some common vocabularies, standards, or ontologies for the relevant application domains of offering providers. Fortunately, there exist many such vocabularies, which could be used for defining BIG IoT Offerings. This section analyzes those existing vocabularies, standards, and ontologies which are relevant for the BIG IoT application domains, and gives advantages and shortcoming of each of analyzed domain dependent and independent ontologies.

We first introduce the different ways to use vocabulary models (ontologies) in the project. Secondly, we analyze the existing domain independent models and domain dependent models. Finally, we introduce the models for non-functional properties of offerings.

2.1. Usage of Ontologies

In general, it is possible to envision 5 different levels of implementation methods when proceeding with inclusion of vocabularies and ontologies in a project during the modelling of the different part of the domains in order to reach the semantic interoperability among different actors, see Figure 2.

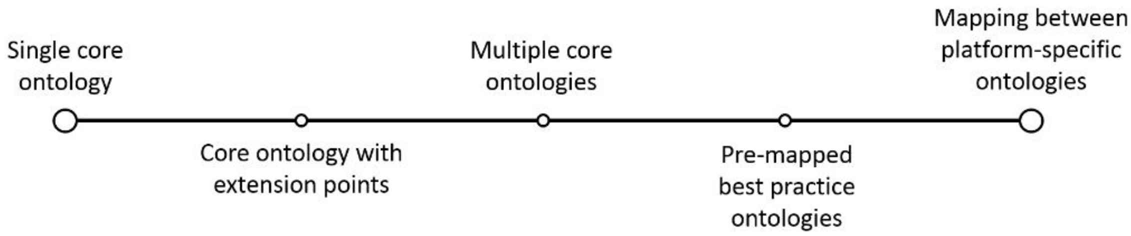

Figure 2: Different ways for combination of ontologies [1]

A Single core ontology: Only one information model that can be used. The information is clearly defined and easy to use. However, some properties of Offerings which are not modelled in the core ontology cannot be expressed.

A Core Ontology with extension points: A core ontology which is explicitly designed for extension. The basic information in core ontology can be easily understandable to all Offering Providers. But, the extension needs to be mapped.

Multiple core ontologies: provides multiple core ontologies for Offering and restrict the offering provider to use only one of them. It provides more opportunity for the offering provider to express his offering. But, maybe it is still not suitable for all offering providers.

Pre-mapped best practice ontologies: similar to multiple core Ontologies, but Offering Providers are not restricted to use one of the core ontologies, instead they can use their own ontologies. Pre-mapped best practice ontologies are easy to use for the offering provider. Interoperability among offerings is only possible with the mapping between different core ontologies

Mapping between platform-specific ontologies: Every platform provides its own information model/ontology. All information of the offering provider could be supported. However, the mapping between ontologies is necessary.

In our work, we started from the BIG IoT Semantic Core Model (see D3.2.a) and extended it with our special needs in specific domain model extensions, using a schema.org guidance. The process and models for these extensions are introduced in Section 3 in detail.

2.2. Domain Independent Models

Domain independent models are information models which do not dependent on a specific domain application, such as parking, smart city, or air quality. They have a wide agreement. Examples are schema.org or the W3C SSN Ontology.

2.2.1. schema.org

schema.org[1] is a joint initiative by Google, Bing, Yandex and Yahoo. schema.org was mainly designed (and is still primarily used) to annotate Web pages in order to improve search results. It provides a common vocabulary that Web developers can use to structure the information provided on their Website in order to improve search results. Beyond optimizing search results, BIG IoT could utilize schema.org to increase interoperability between platforms, for instance by describing the data provided by a platform with the vocabulary defined by schema.org. Schema. org is one of the best candidates for representation of information of offerings. It is introduced in Section 4 in detail.

Advantage: a community activity with a mission to create, maintain, and promote schemas for structured data on the Web. It is developed as a Core-Ontology-with-extension-points ontology (Figure 2) and can be easily extended. For example, schema.org is being extended for the domain Internet of Thing. This extension is called iot.schema.org.

Shortcoming: schema.org still lacks more basic information to be used in the context of Internet of Things applications.

[1] [1] https://schema.org

2.2.2. W3C SSN Ontology

Semantic Sensor Network Ontology (SSN) [3] is an ontology especially designed for the representation of sensors. It is developed by the W3C Semantic Sensor Network Incubator Group. It supports general descriptions of sensors and aspects related to them.

Advantage: SSN is a widespread ontology in the IoT domain.

Shortcoming: it is designed only for sensors, in order to represent more base information, it needs to be aligned with other domain independent ontologies.

2.3. Domain dependent Models

Domain dependent models are information models which can be used for specific domain applications. These models are candidates to be used for use cases which defined in D5.1a.

2.3.1. M3

Machine to Machine Measurement (M3)

M3 [3] nomenclature has been implemented in the M3 Ontology which references more than 30 sensors, measurements, units and about 10 domains. The M3 ontology is focused on the ssn:ObservationValue concept from the W3C SSN ontology which describes sensors and observations, and related concepts. M3 is also an extension of W3C SSN which was considered insufficient to unify sensor metadata such as describing units, sensor measurement types, applicative domains, and sensor types.

M3 ontology was mainly created to classify and reference the various domains, measurements and sensor types to unify the description of the M2M data. M3-lite taxonomy is a refinement of the M3 ontology (refactored, cleaned and aligned with the vision of the EU H2020 FIESTA-IoT project[2]).

Advantage: It has ability to cover not only the conventional sensor layer, extended application of SSN ontology, but also various domains, units and measurements.

Shortcoming: unfortunately, it doesn't cover the specific domains that are needed and has some limitations in various categories of classes and properties.

2.3.2. MobiVoc

MobiVoc [4]: The Open Mobility Vocabulary is being developed as an open vocabulary for future-oriented mobility solutions. Mobivoc is a project under the supervision of the ITA - Automotive Partnership Association. It provides an open, standardized vocabulary for forward-looking mobility solutions and data-based value-added services to support mobility. MobiVoc can present the mobility information with mobility related terms such as Battery, BikeSharing, ChargingPoints, FillingStation, Fuel, LowEmissionZone, MeansOfTransport, Parking, or Vehicle. The MobiVoc initiative intends to enable data communication among all available data sources by providing a powerful vocabulary for modeling the mobility data.

Advantage: It has ability to cover not only the conventional mobility, but also the mobility for the future.

Shortcoming: continuing promotion and support still need to be observed.

2.3.3. DATEX II

In the road sector, the DATEX standard [5] has been developed for information exchange between traffic management centres, traffic information centres, and ser-

[2] http://fiesta-iot.eu/

vice providers. It constitutes the reference for applications that have been developed in the last 10 years. The second generation, the DATEX II specification, also opens up for all actors in the traffic and travel information sector. DATEX II is of relevance for applications where dynamic information on the transport systems and notably the road system is concerned. DATEX II already covers a wide range of content in the road traffic and transport domain. It is one of its main achievements to establish a logical model for this domain that is widely supported by users all over Europe. DATEX uses an XML data format with an XSD schema. During the open summer of code 2016, a project was started to create a linked DATEX version using DATEX as a vocabulary for linked mobility data. The Linked Datex 2 - was thought to annotate existing DATEX II feeds into JSON-LD using a DATEX II vocabulary, however, this development is still in early stage status and the semantic vocabulary is not well structured.

Advantage: DATEX II is widely used in different EU projects.

Shortcoming: Alignment with other standards needs to be considered.

2.3.4. OCPI

The Open Content Provider Interface (OCPI) [6] is a comprehensive interface which allows for the quick and cost efficient integration of other systems. It is used by Siemens in context of the Sitraffic® family (Scala / Concert / Guide) to model traffic messages, as well as parking and environmental data, or cameras.

Advantage: it enables communication via other standards such as WanCom, TLS, DAV, DATEX II, or VDV.

2.4. Models for Non-Functional Properties in Offering

In the BIG IoT project, Non-Functional Properties (NFPs) plays an important role in both Offering Description (OD) and in Offering Composition Recipe or Service

Recipe (SR). Semantic model for NFPs supports semantic representation for time, location, Billing of offering, quality of offering, etc. Some domain independent Ontologies describe models which can be used for the NFPs such as schema.org and Web Service Modeling Ontology (WSMO) [7]. schema.org defines price specification, payment method, broker and accounted which can be used for representation of billing of offerings. Further, it defines spatial availability, temporal availability, rating, and license, which can be used for the representation of quality of service of offerings. WSMO also provides ontologies for modeling NFPs and includes concepts such as availability, currency, payments, price, or security. Representation of time and location of offering plays the key role for offering description as well as for offering discovery. Therefore, in this section, we focus on analyzing of existing ontologies which describe spatial and temporal aspects of NFPs.

2.4.1. Location

As can be easily inferred, every resource (Provider, Organization, or Offering) has some physical location, either gathered from a built-in GPS device attached to each device or through a manual annotation. In most cases, a "latitude/longitude" pair will be generated, but there are also more advanced geospatial notations. For certain kind of entities such as Organizations or Providers, there is the necessity to have more general high level location information related for example to city, country or street. There are various vocabularies we can consider for this:

- **WGS84 Geo Positioning**

The WGS84 Geo Positioning ontology [8] is a W3C vocabulary for representing latitude, longitude and altitude information in the WGS84 geodetic reference datum. This is a basic RDF vocabulary that provides the Semantic Web community with a namespace for representing latitude, longitude and other few information about spatially-located things.

- **The GeoNames Ontology**

The GeoNames Ontology [9] makes it possible to add geospatial semantic information on the Web. All over 11 million geonames toponyms have a unique URL with a corresponding RDF web service. Other services describe the relation between toponyms. The Features in the GeoNames Semantic Web are interlinked with each other. Depending on applicability the following documents are available for a Feature:

The children (countries for a continent, administrative subdivisions for a country, etc.). As an example, see the children of France:

http://sws.geonames.org/3017382/contains.rdf.

The neighbours (neighbouring countries): As an example the neighbours of France can be accessed from: http://sws.geonames.org/3017382/neighbours.rdf

Nearby features: Nearby to the Eiffel Tower are Champ de Mars, Trocadéro - Palais de Chaillot, etc.: http://sws.geonames.org/6254976/nearby.rdf

2.4.2. Time

Temporal aspects are essential for any system addressing real world phenomena, e.g. smart city IoT systems. Timestamps can be used in an offering to describe when a sensor reading was taken or when it was valid. Multiple readings can be ordered by the time of their occurrence.

- **Time Ontology**

An established ontology for this is Time Ontology in OWL [10]. OWL Time allows describing of temporal properties and relationships. It also supports time intervals as well as durations, which are useful for example, when describing imprecise measurement times as well as complex event specifications.

3. Consolidated Application Domain Model for BIG IoT

This Section describes from a high-level point of view the coherent and consolidated BIG IoT Semantic Application Domain Model. This model comprises the vocabularies usable by the realization of the use cases in WP5. Hence, it is highly influenced through the analysis of platforms and services to be contributed to the use cases.

The purpose of these vocabularies is to ensure semantic interoperability that bridges applications and services within BIG IoT as well as across other Linked Data Platforms. We have started this work by defining semantics for offerings, i.e., Offering Description (OD) in Task 3.2 and documented in D3.2.a. In Task 4.2 and documented in this deliverable, we extend these semantics with domain dependent and domain independent vocabularies as far as we reach an application specific semantics for services / offerings.

The following figure depicts this aim:

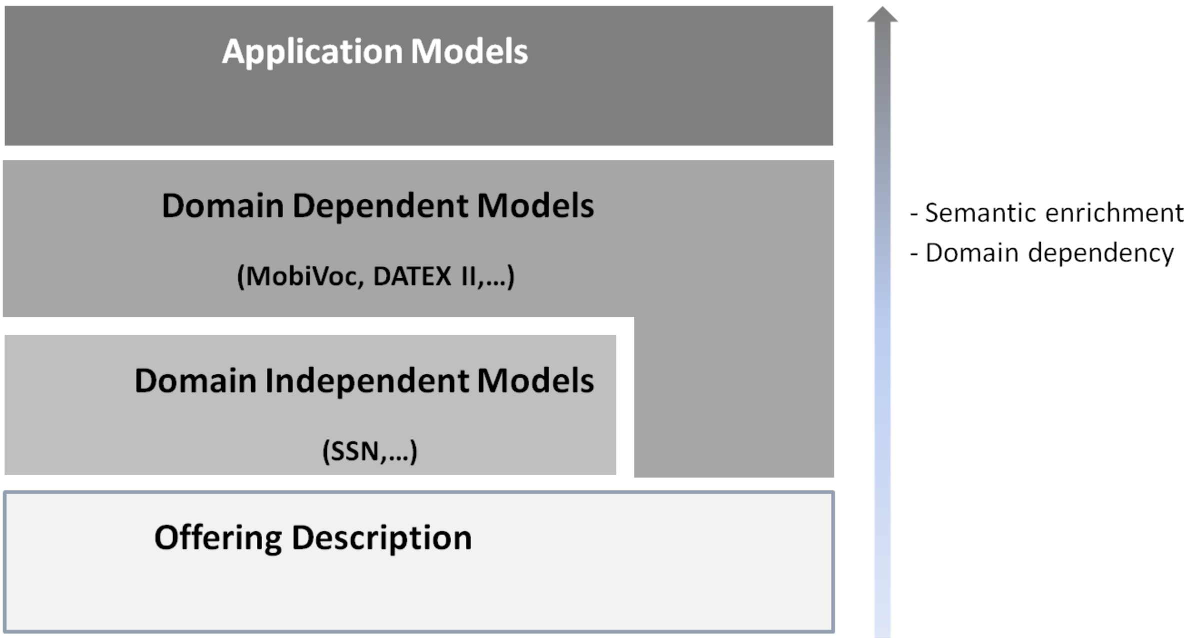

Figure 3: BIG IoT Semantic Modeling Layercake

In this deliverable it is also very important to explain our project-long aim related to Mobility Domain Models in BIG IoT:

1. identify a common set of vocabularies to build semantically interoperable solutions upon;
2. start the work within the consortium on integration of the identified set of mobility-related vocabularies under the common principles;
3. promote and extend the identified set of vocabularies as a community effort on realization of common semantics for mobility IoT applications.

Principles:

- **Re-usability** - in order to enhance interoperability, we use existing widely-used and standard vocabularies that are in scope of the BIG IoT project. We also define new vocabularies there where this is not the case or we have to address the project specific needs;
- **Simplicity** - we follow an approach where semantic vocabularies will be simple and light-weight. schema.org is a good example of such an approach. It is grounded on an RDF data model and uses RDF(S) to define semantics of its vocabularies. Therefore, we specify BIG IoT vocabularies with a schema.org-like approach;
- **Standard-based** - vocabularies are grounded on the W3C standards (RDF, RDF(S), JSON-LD etc.). Existing standard vocabularies in various non-RDF data formats will be translated;
- **Openness** - we specify BIG IoT vocabularies with schema.org-like approach and open them.

3.1. Mobility Domain Model for BIG IoT Applications:

This subsection introduces the Mobility Domain Model. The model aims to create the mobility domain vocabulary required to model the offerings and services of use cases defined in WP5. The model is developed adhering to the above principles.

1. The existing vocabularies such as DATEX II, MobiVoc, schema.org, and m3ontologies are reused as much as possible in the development of Mobility Domain Model. New vocabulary is defined in the model only if the existing vocabularies does not define the terms required to model the BIG IoT pilot use-case offerings and services.

   For Example:

```
mobility:Traffic
rdf:type owl:Class ;
rdfs:subClassOf [
rdf:type owl:Class ;
owl:oneOf (
mobility:traffic
mobility:trafficInformationCategory
mobility:trafficMonitoringCategory ) ; ] ; .
```

   In this example, the class "mobility:Traffic" is defined as a enumeration class with its instances "mobility:traffic", "mobility:trafficInformationCategory", "mobility:trafficMonitoringCategory" in Mobility Domain Model to categorize traffic related offerings. As there is no equivalent definition found in the existing vocabularies.

2. The model is developed iteratively. Current model consists of the vocabulary required to hierarchically categorize the offerings defined by WP5 in the first iteration. In addition, the model consists of vocabulary to model the input/output data and metadata of the offerings in the use-cases "Parking Spot Availability Service" and "Traffic Monitoring Service".

3. The model is created in Turtle format (.ttl) which is used for expressing data in the W3C standard RDF data model. The model has the namespace "mobility: http://big-iot.eu/mobility#".
4. Our aim is to specify Mobility Domain Model in schema.org like approach. To adhere to this, we reuse schema.org vocabulary as much as possible and we follow the guidelines of schema.org such as: while reusing the existing vocabulary, we replace the original namespace of the vocabulary and define new namespace called "mobility". We use schema.org approach to refer to the original vocabulary by using the annotation property "dct:source".

   For example:

```
mobility:ParkingSpace
rdf:type owl:Class ;
dct:source <http://vocab.datex.org/terms/ParkingSpace> ;
```

   The above example shows that our Mobility Domain Model reuses the "ParkingSpace" class defined in Datex. The annotation property "dct:source" is used to refer to the original term (a class or a property).

The model mainly defines the following:

- The vocabulary to categorize the BIG IoT Offerings hierarchically. E.g., an offering can be categorized as either "Parking" or "Traffic" or "Transportation" or its sub-categories depending on the category the offering belongs to.
- The vocabulary to annotate the semantics of input and output data of an offering.
- The vocabulary to annotate the mobility domain dependent metadata of an offering. For example, to annotate a Parking site or a Parking space, to semantically define a street section etc.

3.1.1. Model Namespaces

Table 1 below shows the namespaces in the mobility domain model. A new namespace "mobility" is defined in this model.

Table 1: Model Namespaces

Prefix	Ontology/Language	Namespace
mobility	Mobility Domain Model	http://big-iot.eu/mobility#
schema	schema.org ontology	http://schema.org/
xsd	XML schema Definition	http://www.w3.org/2001/XMLSchema#
rdf	RDF Concepts Vocabulary	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	RDF Schema ontology	http://www.w3.org/2000/01/rdf-schema#

3.1.2. BIG IoT Semantic Application Domain Models

The BIG IoT Core Model created in T3.2 (see deliverable: D3.2.a Semantic interoperability design for smart object platforms and services) defines the core vocabulary required to create an Offering Description. The semantics are further enriched by domain independent and domain dependent models. A domain dependent model is used to semantically annotate the metadata, OfferingCategory and input/output data of an Offering Description. BIG IoT Semantic Application Domain Model is created using the BIG IoT Semantic Core Model, domain independent and/or domain models. This model establishes the relationship between the core model and domain model. The BIG IoT Semantic Core Model defines among other things, the classes "bigiot:Offering", "bigiot:OfferingCategory" and "bigiot:Data". The class "bigiot:OfferingCateogry" is used to model the category to which an offering belongs to. The class "bigiot:Data" is used to model the input and output data of an offering. To create Offering Descriptions the BIG IoT Core Model and domain models are used. The Mobility Domain Model defines vocabulary to semantically annotate "bigiot:OfferingCategory", "bigiot:Data" and metadata specific to mobility related offerings and services.

Offering Category:

In the first iteration, the following categories are defined in this domain model

- **mobility:Parking**: All the offerings relating to parking such as "Parking space availability offering" belong to the parking category or a sub-category of parking such as "mobility:parkingSpaceMgmtCategory".
- **mobility:Traffic**: All the offerings relating to traffic such as "Monitor traffic alarms" or "Get traffic information" belong to the traffic or a sub-category of traffic.
- **mobility:Transportation**: All the transportation related offerings such as "People density estimation offerings" belong to transportation or a sub-category of transportation.
- **mobility:Environment**: All the offerings relating to environment can be categorized under the category environment.

Figure 4 shows the relationship between the "bigiot:OfferingCategory" class and the "mobility:Traffic", "mobility:Transportation", "mobility:Environment", "mobility:Parking" classes in the mobility model. The above mobility classes are modeled as sub-classes of "bigiot:OfferingCategory". The figure also shows the hierarchical classification of all the categories. Each category is defined as an enumeration class and the sub-categories are defined as enumeration values of the class. For example: The class "mobility:Parking" has enumeration values "mobility:parkingSiteMgmtCategory", "mobility:parkingSpaceMgmtCategory" and a default value "mobility:parking".

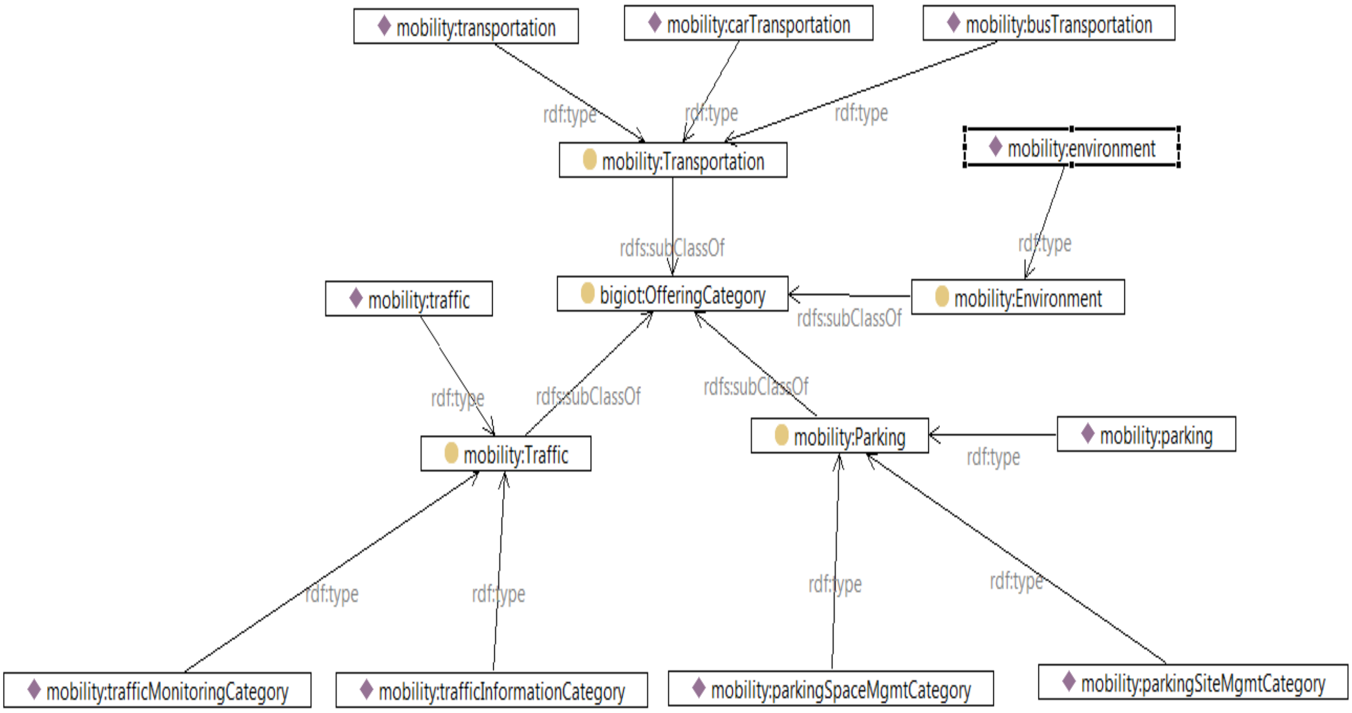

Figure 4: Hierarchical classification of offering categories

InputData/OuputData Modeling:

The Mobility Domain Model also defines the vocabulary to semantically annotate the input data and output data of an offering in the mobility domain. The class "mobility:Data" which is defined as subclass of "rdfs:Datatype" is used to model the input/output data of an offering. The class "bigiot:Data" is defined in the BIG IoT Core Model created in T3.2 (see deliverable: D3.2.a Semantic interoperability design for smart object platforms and services). The BIG IoT Application Domain Model creates the relationship between bigiot:Data and mobility:Data.

```
mobility:Data
rdf:type owl:Class ;
dct:source <http://vocab.datex.org/terms/BasicData> ;
rdfs:comment "Data that is either measured or calculated (elaborated) at
the same time or over the same time period."@en ;
rdfs:subClassOf <http://big-iot.eu/core#Data> ;
```

The Figure 5 shows the hierarchical classification of Data classes and the sub-class defined for "mobility:Data" and "mobility:DataValue". In the Application Domain Model "bigiot:Data" class has two subclasses "mobility:Data" and "mobility:DataValue". The subclasses of "mobility:Data" define the data that can be either measured or calculated by an offering at the same time or over the same time period. It defines the complex data types required to model the input and output data of the offerings. E.g., the data type "mobility:TrafficSpeed" can be used to model the input / output data in the offerings that measure the speed of the traffic in a certain location.

The class "mobility:DataValue" defines a data value of something that can be measured or calculated. E.g., the class "mobility:DurationValue" defines a measured or calculated value of a period of time expressed in seconds.

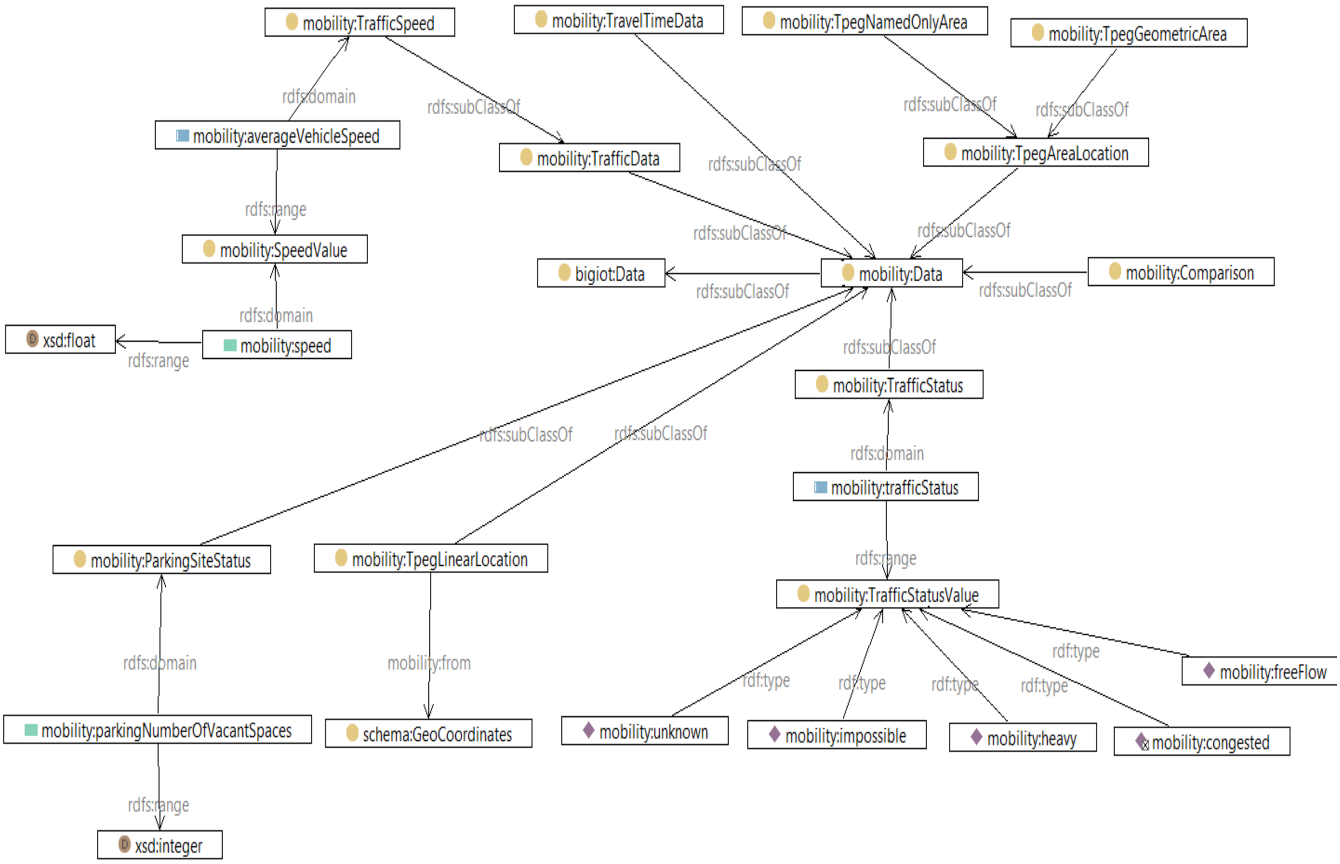

Figure 5: RDF Schema of BIG IoT Application Domain Model

3.1.3. Domain Dependent Metadata Modeling

Along with offering categorization and data modeling, the Mobility Domain Model also defines the vocabulary to semantically annotate the domain dependent features of an offering. For example, the class "mobility:ParkingSpace" can be used to annotate a parking space and its features such as parking space id, or parking space location.

The Figure 6 and Figure 7 show the examples of the classes ParkingSite and ParkingSpace.

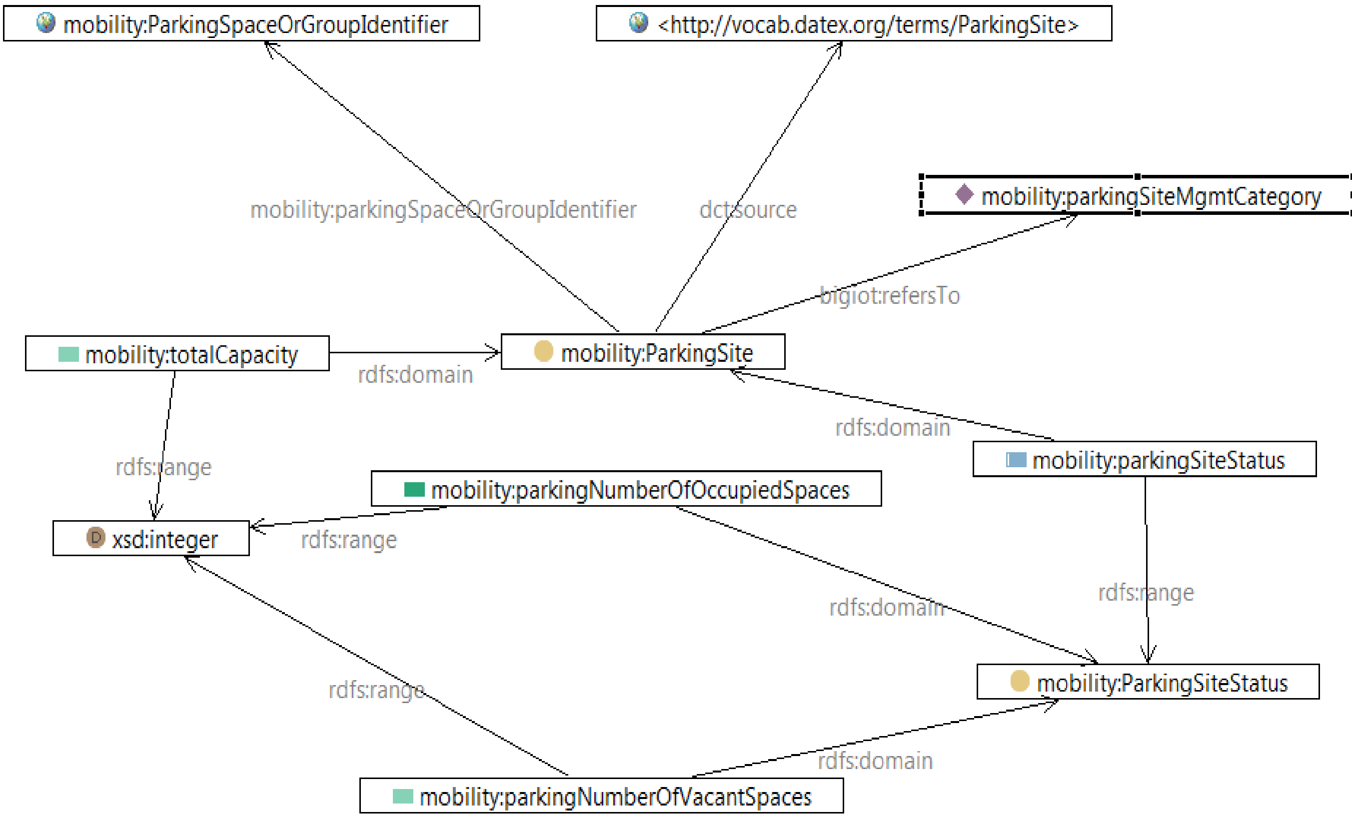

Figure 6: ParkingSite class

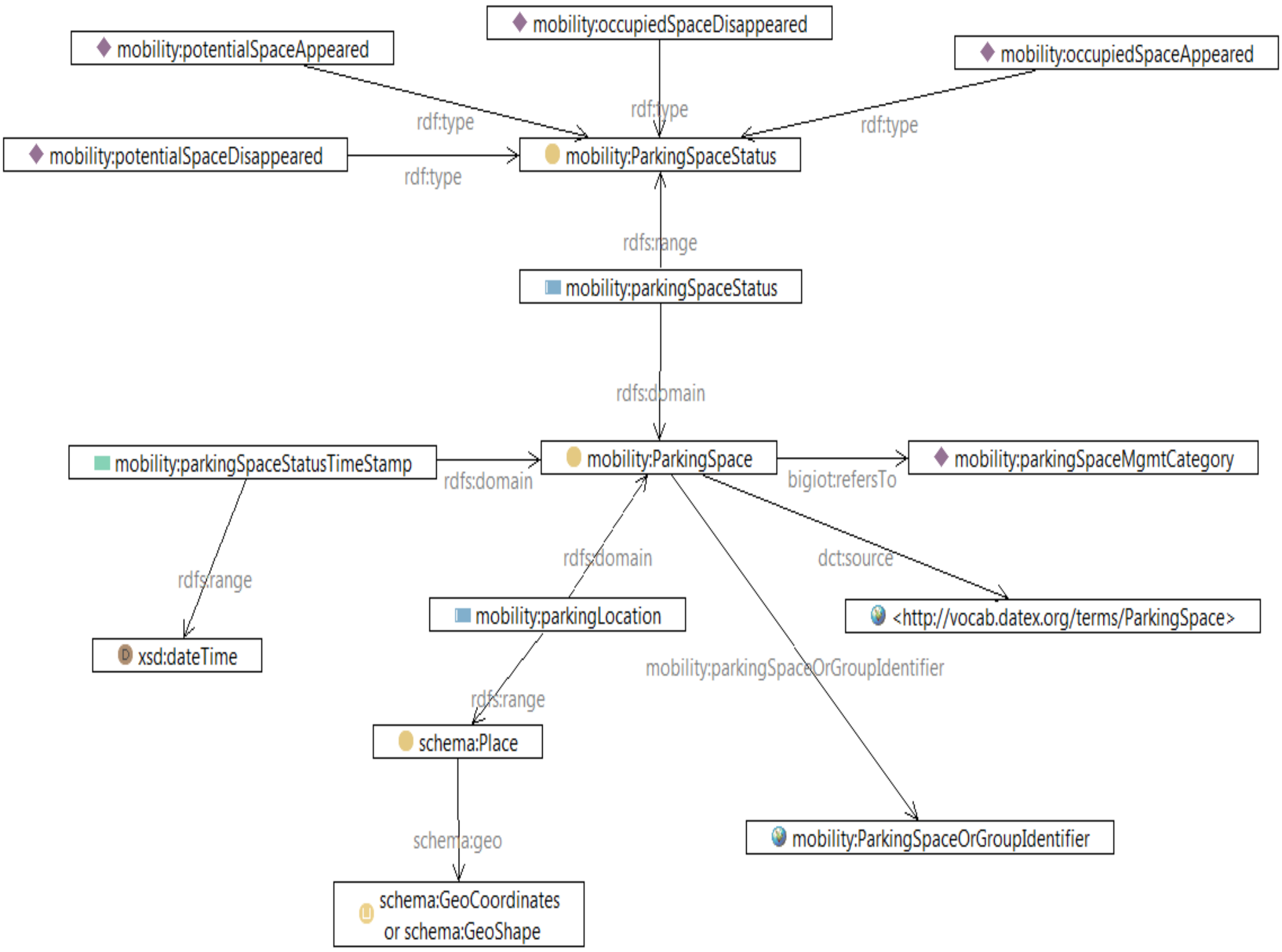

Figure 7: ParkingSpace class

The first version of the BIG IoT Semantic Application Domain Model in Turtle (.ttl) format is presented in the Annex. The model is based on the first version of the BIG IoT Semantic Core Model and the Mobility Domain Model. It defines all the concepts for offering categorization, input and output data modeling and domain dependent metadata required to create the offering descriptions for the BIG IoT pilot use cases.

4. BIG IoT schema.org Implementation

This Section describes the implementation of the models described in Section 3. This implementation is done by aligning them with and extending schema.org.

4.1. Introduction to schema.org

The schema.org project has started around 5 years ago with the aim to provide a vendor-neutral, community developed vocabulary for structured data on the Web. Nowadays, schema.org markup is widely used, with tens of millions of mainstream Web sites routinely including schema.org markup within their page content. schema.org approach has proved very beneficial to Web publishers because it allows to improve the presentation of their content within search engines.

The schema.org project now is looking to extend its vocabulary to IoT devices and services in the same vendor-neutral, community-developed fashion. For sure bringing schema.org to the IoT specific domain raises a number of considerations beyond simple modeling tasks. However, we can envisage some benefits of IoT applications using schema.org vocabulary for connected devices and services, e.g.(i)that they build on a large existing base, (ii) that they do not draw an unsustainable distinction between "IoT" and "non-IoT" scenarios/applications, and (iii) that they help decouple data structures from the tools, products and applications that use the data.

In IoT we have different layers, from single sensors, to devices, networks and application that exchange many kinds of data using Internet, Web and other network technology. Often these structured data describe real world entities, their properties and relationships, in a manner that could benefit from the use of widely adopted, application independent, non-proprietary schemas. This can be an interesting reason why adopting and improving schema.org's vocabularies for IoT can reduce needless duplication and focus attention on innovation rather than replication. As reported by their founders, the schema.org's approach is open, organic and community based, and is supported by a technical design that includes several extensibility mechanisms designed to support decentralization without encouraging fragmentation.

IoT-related applications are incredibly heterogeneous and are highly inter-related with other application domains such as health-lifescience, e-commerce, geo-spatial, datasets, or sensors. Many existing schema.org schemas are also highly relevant to IoT applications. Below is a short selection of situations where the schema.org project could contribute in an IoT context:

- User data portability
- Beacons or sensors and the description of the physical environment
- Smart Assistants
- On-device content
- Multi Domain

For a better exploitation and future expansion the official schema.org project has created W3c Community Groups to improve the evolution of schemas and for developer engagement and now various W3c Groups, e.g. Web of Things and Spatial Data on the Web are interacting with schema.org to discuss about some official integration of part of its schemas and future extensions in their specifications.

An ideal outcome would be for all schema.org's schemas to be usable (directly or via mappings) by a variety of platforms and standards in the IoT/WoT ecosystem. In addition to the specific IoT/WoT ecosystem it is also possible to (re)use the same principles for extending schemes and classes related to general domain of application e.g. Smart Cities, public transportation, parking and traffic management, or business and accounting and spatial extensions.

Supporting these purposes, within the schema.org initiative the community has developed mechanisms for the creation and development of community extension, as a way of adding more detailed specific vocabularies that builds on the schema.org core.

For example, as reported in schema.org “Organization of Schemas” page [https://schema.org/docs/schemas.html], there are:

- “Hosted extensions are managed and published as part of the schema.org project, with their design often led by one of more dedicated community groups.”
- “External extensions live elsewhere in the Web, typically managed by other organizations with their own processes and collaboration mechanisms. Please consult external documentation for full details of their vocabulary, versioning system and release history.”

By using this extension mechanism, the core vocabulary is already being extended by the following hosted extensions:

- auto.schema.org
- bib.schema.org
- health-lifesci.schema.org

The BIG IoT project aims to introduce and promote some extensions to the schema core and the new iot.schema domain and introduce new domain specific extensions to cover the needs for vocabularies that cover areas like Transportation, Mobility (e.g., traffic, and parking), Environment, and other Smart Cities related fields as results of our research in our use-case application domain modelling. In this regard we have started the work on a new extension called **mobility.schema.org**.

4.2. BIG IoT mobility.schema.org

Mobility.schema.org extends the schema.org vocabulary for the mobility domain. Further on, it also implements the BIG IoT Mobility Domain Model following the schema.org guidance. schema.org offers software for a community-driven work on creating, publishing, and evolving vocabularies. In the BIG IoT project, we use this software for creating the extension mobility.schema.org. Therefore, in this section we provide a tutorial on how to install the schema.org software, and further how to create an extension vocabulary and its term definitions.

4.2.1. Installation Instructions

Step 1: In order to run the schema.org software one needs to clone the schema.org repository from GitHub. This can be done from the schema.org repository by forking the project under a certain account name (e.g., myAccount):

```
git clone https://github.com/myAccount/schemaorg.git
```

Step 2: Download the Standard Development Kit (SDK) of Google App Engine (e.g., google_appengine_1.9.40) from here. Unzip the code, and in the same folder where the code is, run the following command:

```
dev_appserver.py schemaorg
```

If your copy of schema.org repository is not in the same folder as the SDK of Google App Engine, provide the path to it. After executing the above command, the successful output should look like the following:

```
darko@darko-VirtualBox:~/software/google_appengine$ ./dev_appserver.py ../../repos/schemaorg
INFO     2016-12-02 15:08:52,006 sdk_update_checker.py:229] Checking for updates to the SDK.
INFO     2016-12-02 15:08:52,008 __init__.py:124] Connecting through tunnel to: appengine.google.com:443
INFO     2016-12-02 15:08:52,269 sdk_update_checker.py:273] This SDK release is newer than the advertised release.
INFO     2016-12-02 15:08:52,365 api_server.py:205] Starting API server at: http://localhost:58906
INFO     2016-12-02 15:08:52,386 dispatcher.py:197] Starting module "default" running at: http://localhost:8080
INFO     2016-12-02 15:08:52,396 admin_server.py:116] Starting admin server at: http://localhost:8000
```

Now a local schema.org web site can be accessed in a Web browser from:

```
localhost:8080/
```

The provided instructions apply in a Linux-based environment. It is worth noting that similar procedure can be used to run the schema.org software on Windows-based machines too.

4.2.2. Instructions to Create a Hosted Extension on schema.org

As schema.org expands to the IoT domain, we believe that there will be a need to create a number of hosted extensions. IoT itself involves various domains such as mobility, manufacturing, buildings, energy and so on. Therefore, we can expect various forms of domain vocabularies available on the Web for IoT applications. In this respect we have started the work on a hosted extension mobility.schema.org. This section briefs how a hosted extension can be created in schema.org.

If you inspect the schema.org repository you will notice a directory named “data”. This folder contains various files and directories, but most notably it contains the core schema.org vocabulary. This vocabulary is provided as an RDFa file called schema.rdfa. Hence each class or a property definition in schema.org is specified in this file. Further on, schema.org users know that many examples, accompanied with term definitions, are provided on the Web site. These examples are stored in an examples.txt file, which is also in the data folder. If you want to propose a change in the core schema.org vocabulary, you are supposed to provide changes in schema.rdfa and examples.txt of your local forked version of schema.org. You may then create a so-called “Pull Request” on the GitHub repository and discuss your proposal within the community in order to get the proposal eventually accepted by schema.org.

Another important directory in the “data” directory is “ext”. It contains vocabularies for all hosted extensions and corresponding examples. One can find there existing extensions such as auto, bib, and health-lifescience. In this directory we have created a directory called mobility. This is home for the BIG IoT mobility vocabulary. The directory contains two files, mobility.rdfa with current term definitions, and examples.txt with examples we have provided so far.

4.2.3. Creating Vocabulary in BIG IoT mobility.schema.org

In this section our goal is to show how to specify a term definition. Let us assume we want to define a type (class) ParkingSpace in mobility.schema.org. The following code snippet provides an example specification of the term in RDFa. The specification is supposed to be a part of the mobility.rdfa file.

Table 2 ParkingSpace in RDFa

```
<div typeof="rdfs:Class" resource="http://schema.org/ParkingSpace">
    <link proper-
ty="http://schema.org/isPartOf" href="http://mobility.schema.org" />
    <span class="h" property="rdfs:label">Parking Space</span>
    <span property="rdfs:comment">A single parking space. It is pos-
sible to define the same parking space more than once with differ-
ent
    properties, e.g. when there is a different parking assign-
ment for different times.</span>
    <span>Subclass of: <a proper-
ty="rdfs:subClassOf" href="http://schema.org/CivicStructure">CivicStr
ucture</a></span>
</div>
```

The first line of the snippet indicates a type definition (typeof="rdfs:Class") and its canonical URI: http://schema.org/ParkingSpace. The second line tells us that the term ParkingSpace is not in the schema.org core vocabulary, but is defined in an extension. The URI of the extension is http://mobility.schema.org. The third line provides a human readable label (rdfs:label) for the term, together with a descriptive comment (rdfs:comment). Finally, the snippet specifies the subclass relations between the defined term and a class CivicStructure (with its URI: http://schema.org/CivicStructure). If we restart the server with the schema.org repository (see Section 4.2.1), then we can find our newly defined term ParkingSpace. The following figure shows the term.

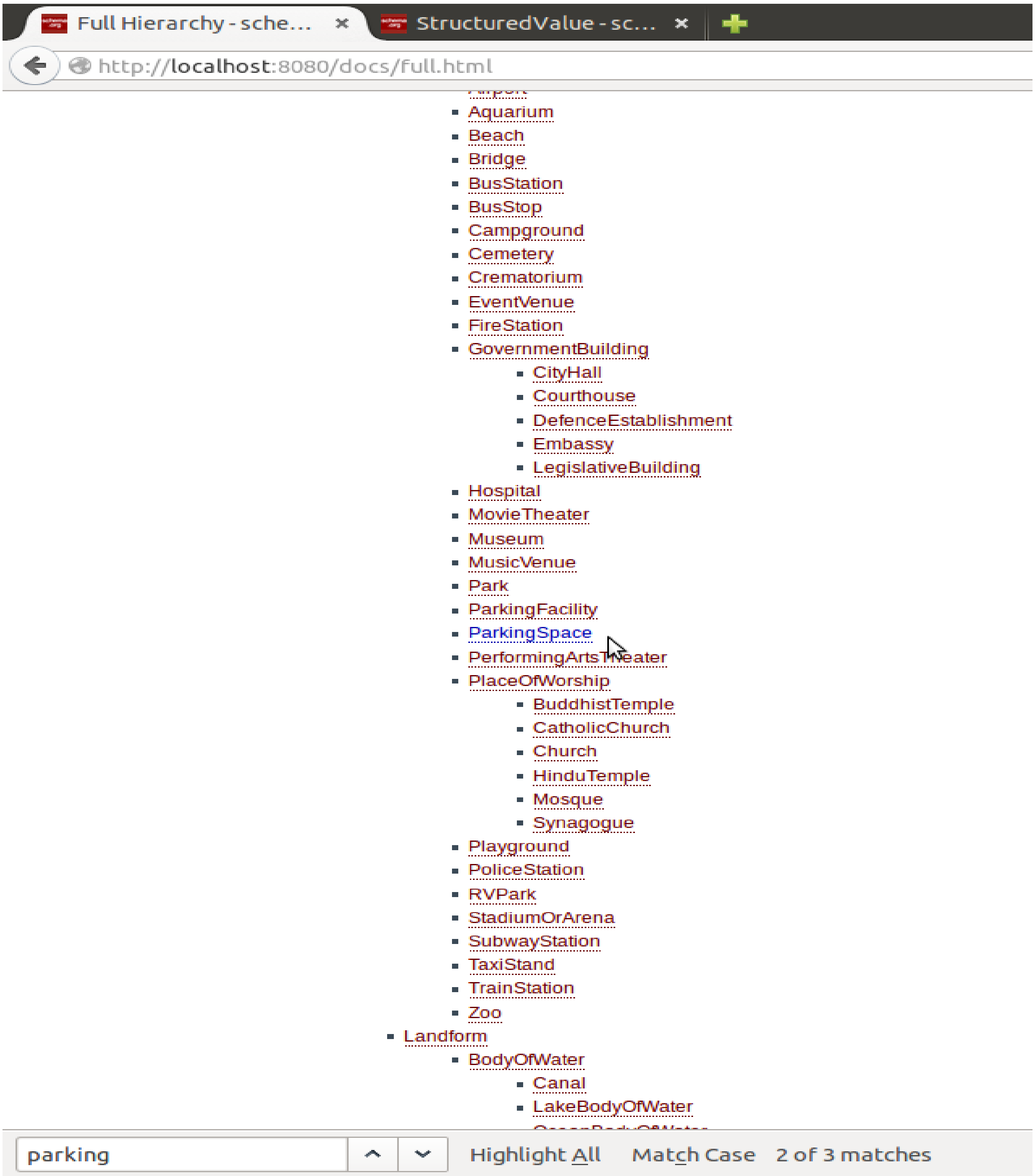

Figure 8: Defined term ParkingSpace

In a similar way we may proceed with the specification of our example vocabulary. The following code snipped shows the definition of few types and properties (i.e., **ParkingSpaceOrGroupIdentifier, hasParkingSapceOrGroupIdentifier, ParkingSpaceStatus, hasParkingSapceStatus, Parking, ParkingSiteMgmtCategory**). These terms have been created here to reflect our examples from Section 3. Thus for an explanation of the terms, an interested reader is referred to this section. In the following our intention is to show how the example terms need to be implemented in schema.org.

Table 3 Sample of types and properties

```
<div
typeof="rdfs:Class" resource="http://schema.org/ParkingSpaceOrGroupId
entifier">
    <link proper-
ty="http://schema.org/isPartOf" href="http://mobility.schema.org" />
    <span class="h" property="rdfs:label">ParkingSpaceOrGroupIdentifi
er</span>
    <span property="rdfs:comment">A parking space or group Identifi-
er.</span>
    <span>Subclass of: <a proper-
ty="rdfs:subClassOf" href="http://schema.org/DataType">DataType</a></
span>
</div>
<div
typeof="rdf:Property" resource="http://schema.org/hasParkingSapceOrGr
oupIdentifier">
    <link proper-
ty="http://schema.org/isPartOf" href="http://mobility.schema.org" />
    <span class="h" property="rdfs:label">parkingSapceOrGroupIdentifi
er</span>
    <span property="rdfs:comment">A property that relates parking
space with parking space identifier</span>
    <span>Domain: <a proper-
ty="http://schema.org/domainIncludes" href="http://schema.org/Parking
Space">ParkingSpace</a></span>
    <span>Range: <a proper-
ty="http://schema.org/rangeIncludes" href="http://schema.org/ParkingS
paceOrGroupIdentifier">ParkingSpaceOrGroupIdentifier</a></span>
</div>
<div
typeof="rdfs:Class" resource="http://schema.org/ParkingSpaceStatus">
    <link proper-
ty="http://schema.org/isPartOf" href="http://mobility.schema.org" />
    <span class="h" property="rdfs:label">ParkingSpaceStatus</span>
```

```
    <span property="rdfs:comment">Current status of a parking
space.</span>
    <span>Subclass of: <a proper-
ty="rdfs:subClassOf" href="http://schema.org/DataType">DataType</a></
span>
</div>
<div
typeof="rdf:Property" resource="http://schema.org/hasParkingSapceStat
us">
    <link proper-
ty="http://schema.org/isPartOf" href="http://mobility.schema.org" />
    <span class="h" property="rdfs:label">hasParkingSapceStatus</span
>
    <span property="rdfs:comment">A property that relates parking
space with its current status</span>
    <span>Domain: <a proper-
ty="http://schema.org/domainIncludes" href="http://schema.org/Parking
Space">ParkingSpace</a></span>
    <span>Range: <a proper-
ty="http://schema.org/rangeIncludes" href="http://schema.org/ParkingS
paceStatus">ParkingSpaceStatus</a></span>
</div>
<div typeof="rdfs:Class" resource="http://schema.org/Parking">
  <link proper-
ty="http://schema.org/isPartOf" href="http://mobility.schema.org" />
  <span class="h" property="rdfs:label">Parking</span>
  <span property="rdfs:comment">Parking category.</span>
   <span>Subclass of: <a proper-
ty="rdfs:subClassOf" href="http://schema.org/Enumeration">Enumeration
</a></span>
</div>
<div
typeof="http://schema.org/Parking" resource="http://schema.org/Parkin
gSiteMgmtCategory">
  <span class="h" property="rdfs:label">Parking Site Management Cate-
gory</span>
  <span property="rdfs:comment">A category for the offerings relating
to parking site management.</span>
</div>
```

Finally, once a term definition has been done, it is a good practice in schema.org to show an example demonstrating how to use it. For the terms specified above the following code snippet shows the content we have provided in the example.txt file. Essentially, it is an offering ("name": "**GetParkingSpotInfoOffering**") with annotations specified in our example vocabulary. For example, the offering category is a term that can be found in our mobility.schema.org (i.e., **ParkingSiteMgmtCategory**), and output data of the offering is annotated with vocabulary terms **ParkingSpaceOrGroupIdentifier** and **ParkingSpaceStatus**.

Table 4 "GetParkingSpotInfoOffering" with annotations

```
    TYPES:  ParkingSite Management,ParkingSpace
    JSON:
    <script type="application/ld+json">
    {
    "@context" : ["https://gitlab.com/BIG-IoT/specifications/tree/master/WP3/T3.2/bigiot_core_model.jsonld"],
    "schema": "http://schema.org",
    "providerId": "SP-NG-01",
    "name": "GetParkingSpotInfoOffering",
    "category": "schema:ParkingSiteMgmtCategory",
    "inputData":[{ "name": "destinationArea", "rdfType": "schema:place" }],
    "outputData":[{ "name": "parkingSpotInfo",
                    "members":[{ "name": "parkingSpaceId", "rdfType": "mobility:ParkingSpaceOrGroupIdentifier", "valueType" : "xsd:string"},
                               { "name": "parkingSpaceStatus", "rdfType": "schema:ParkingSpaceStatus", "valueType" : "xsd:string"}]
    }],
    "endpoints": {"schema:url": "http://localhost:8080/bigiot/NG-01/GetParkingSpaceInfoOffering", "schema:type": "HTTP", "method":"GET", "mediaType":"application/json", "schema:accessInterface": "BIGIOT_LIB"},
    "schema:license": "OPEN_DATA_LICENSE",
    "schema:priceSpecification": {"schema:price": "0.002000", "schema:priceCurrency": "EUR", "accountingModel":"PER_ACCESS"},
    "schema:location": {"city": "Berlin"},
    }
    }
</script>
```

5. BIG IoT Semantic Offering Specifications

This section presents the completed semantic description of BIG IoT offerings and services, based on the Offering Description model in D3.2.a and the application domain model defined in Section 4. The technical specifications in context of the BIG IoT pilot use cases regarding these offerings and services can be found in the D5.1.

5.1. Semantic Offering Specifications of Use Case Services

5.1.1. Parking Spot Availability Service

Service description

This service will provide information about available parking spots in an area based on the user location or a targeted route. Following the service description in D5.1, this service is a BIG IoT provider with the main offering described as in the following table:

Table 5 Parking Spot Availability Service overview

Offering properties	Description
Name	ParkingSpotAvailability
Serviced Area	Barcelona city
Price	The price will be based on PER_ACCES accounting model. The cost of one access is 0.02 euro
Input Data	Coordinates: X; Y: (Double): center position, around which free parking spots shall be provided. Radius_Meter (int): Radius in Meter around center position (Buffer for Search)

Output Data	List<ParkingSpot>: List of Parking Spots with their occupancy status (available or occupied). Each parking spot will contain the parking spot id, the timestamp of last status update, the location of the parking spot.

Semantic description:

In the following we give a semantic specification of information provided by the service description.

Semantic Organization Description

A service provider can be described via its organization. The following semantic description provides an example of organization description. The terms used for this purpose are mix of BIG IoT Semantic Core Model terms and schema.org vocabulary.

Organization Semantic Description (JSON-LD)

```
{
  "@context": "http://big-iot.eu/core#",
  "name": "UPC",
  "organizationId": "orgaA",
  "schema:sourceOrganisation": "Barcelona_City",
}
```

Semantic Provider (Service) Description

Similarly, the service provider itself can be described as follows.

Provider Semantic Description (JSON-LD)

```
{
  "@context": "http://big-iot.eu/core#",
  "name": "Parking information service",
  "providerId": "145sfwr",
  "schema:sourceOrganisation": "orgaA",
}
```


Semantic Offering Description

Finally, we can provide a complete semantic Offering Description for the Parking Spot Availability Service. Apart from BIG IoT Semantic Core Model, for this purpose we use Mobility Domain Model too (see the context of the offering). In this way we can enrich basic BIG IoT concepts (e.g., `name`, `inputData`, `outputData` etc.) with the domain specific annotations (e.g.,`parkingSiteMgmtCategory`, `parkingSpaceOrGroupIdentifier`, `ParkingSpaceStatus` etc.), see Table 6.

Table 6 Parking Spot Availability Offering (JSON-LD)

Parking Spot Availability Offering (JSON-LD)


```
{
  "@context": ["http://big-iot.eu/ctx",
  {"mobility" : "http://big-iot.eu/mobility#"}],

  "name": "Parking Spot Availability Service",
  "@type": "offering",
  "category": "mobility:parkingSiteMgmtCategory",
  "providerId": "1452fwr",
  "inputData": [
    { "name": "areaSpecification", "rdfType": "schema:GeoCircle",
       "members": [{"name": "geoCoordinates", "rdfType":
        "schema:GeoCoordinates",
        "members":[{ "name": "latitude", "rdfType": "sche-
ma:latitude",
         "valueType" : "number"}, { "name": "longitude",
         "rdfType": "schema:longitude", "valueType" : "number"}]},
         { "name": "radius", "valueType": "number",
         "rdfType":"schema:geoRadius" }]}
  ],
  "outputData": [
    {"name": "geoCoordinates", "rdfType": "schema:GeoCoordinates",
```

```
      "members":[{ "name": "latitude", "rdfType": "schema:latitude",
       "valueType" : "number"},{ "name": "longitude",
       "rdfType": "schema:longitude", "valueType" : "number"}]},
    { "name": "parkingSpotID", "valueType": "string",
       "rdfType":"mobility:parkingSpaceOrGroupIdentifier" },
       { "name": "timestamp", "valueType": "xsd:dateTime",
       "rdfType":"mobility:parkingSpaceStatusTimeStamp"},
    { "name": "status", "valueType": "string",
       "rdfType":"mobility:ParkingSpaceStatus" }
  ],
  "endpoint": {
    "url": "http://example.org/pspot",
    "method": "HTTP_GET",
    "accessInterfaceType": "BIGIOT_LIB"
  },
  "license": { "type": "OPEN_DATA_LICENSE" },
  "price": {
    "amount": 0.002,
    "currency": "EUR",
    "accounting": "PER_ACCESS"
  },
  "region": "http://sws.geonames.org/3128760/"
}
```


5.1.2. Traffic Monitoring Service

Service Description

Traffic Monitoring Service will implement data filtering, map matching (for connected cars data) and estimate traffic variables (times and speeds).

Table 7 Traffic Monitoring Service overview

SERVICE NAME	Traffic Monitoring Service
Description	Traffic Monitoring Service will implement data filtering, map matching (for connected cars data) and estimate traffic variables (times & speeds).
Consumer - Resources consumed by this service	

Inputs	Traffic sensors - Bluetooth/WIFI: hash, timestamp, signal intensity, type (WIFI/Bluetooth); travel times (through WS API) Traffic sensors - magnetometers, loop detectors: vehicle count, instantaneous speed Traffic sensors - cameras: vehicle count Smart Cars data: geo position, speed, acceleration
Provider - Offerings provided by this service	
[STM-BCN-01] [STM-PIE-01] Get Traffic Information	Input: Area specification (i.e. coordinates + radius) Outputs: Set of Travel Time, Speed and other information at link level (where available)
[STM-BCN-02] [STM-PIE-02] Monitor Traffic Alarms	Input: Set of links Warning KPI's threshold Alarm Monitoring Interval Outputs: Push Notification of threshold reached for specific KPI
[STM-BCN-03] Get Travel Times between 2 points	Input: Predefined Origin-Destination pair Output: Estimation of Travel Time
[GRP-BCN-01] [GRP-PIE-01] Find Green Routes	Input: Area specification or set of links Output: Set of Travel Time, Speed and other information at link level (where available)

This service provides multiple offerings that can be describes as follows:

Offering 1:

Table 8 Traffic Monitoring Service Offering 1

Offering properties	Description
Name	Get Traffic Information
Serviced Area	Barcelona city
Price	The price will be based on PER_ACCES accounting model. The cost of one access is 0.02 euro
Input Data	Coordinates: X; Y: (Double): center position, around which free parking spots shall be provided. Radius_meter (int): Radius in meter around center position (Buffer for Search)
Output Data	Set of Travel Time, Speed and other information at link level (where available)

Offering 2:

Table 9 Traffic Monitoring Service Offering 2

Offering properties	Description
Name	Monitor Traffic Alarms
Serviced Area	Barcelona city
Price	The price will be based on PER_ACCES accounting model. The cost of one access is 0.02 euro
Input Data	Area Specification (coordinates + radius) Warning KPI's threshold Alarm Monitoring Time Interval
Output Data	Push Notification of threshold reached for specific KPI

Offering 3:

Table 10 Traffic Monitoring Service Offering 3

Offering properties	Description
Name	Monitor Travel Time Alarms
Serviced Area	Barcelona city
Price	The price will be based on PER_ACCES accounting model. The cost of one access is 0.02 euro
Input Data	Area Specification (coordinates + radius) Warning KPI's threshold Alarm Monitoring Time Interval
Output Data	Push Notification of threshold reached for specific KPI

Semantic description

Similarly, as in previous section we provide example semantic description for the Traffic Monitoring Service.

Semantic Organization Description

Organization Semantic Description (JSON-LD)


```
{
  "@context": "http://big-iot.eu/core#",
  "name": "UPC",
  "organizationId": "orgaA",
  "schema:sourceOrganisation": "Barcelona_City",
}
```

Semantic Provider (Service) Description

Provider Semantic Description (JSON-LD)

```
{
  "@context": "http://big-iot.eu/core#",
  "name": "Traffic monitoring service",
  "providerId": "1555sfwr",
  "schema:sourceOrganisation": "orgaA",
}
```

Semantic Offering Descriptions

The following semantic Offering Descriptions provide semantic information for:

- **Get traffic information offering**

Table 11 Get traffic information Offering (JSON-LD)

Get traffic information Offering (JSON-LD)

```
{
  "@context": ["http://big-iot.eu/ctx",
  {"mobility" : "http://big-iot.eu/mobility#"}],

  "name": "Get Traffic Information Offering",
  "category": "mobility:trafficInfoCategory",
  "providerId": "1452fwr",
  "inputData":[
      { "name": "areaSpecification",
         "rdfType": "mobility:TpegGeometricArea",
         "members":[{ "name": "areaName", "rdfType":
         "mobility:TpegAreaDescriptor", "valueType" : "xstring"},
      { "name": "radius", "rdfType": "schema:geoRadius",
         "valueType" : "number"},
      { "name": "midPoint", "rdfType": "schema:GeoCoordinates",
         "members":[{ "name": "latitude", "rdfType": "sche-
ma:latitude",
         "valueType" : "number"},
      { "name": "longitude", "rdfType": "schema:longitude",
         "valueType" : "number"}]}
```

```
      ]}
  ],
  "outputData":[
      { "name": "streetSection", "rdfType": "mobili-
ty:TpegLinearLocation",
         "members":[{ "name": "nodeStart", "rdfType": "mobili-
ty:from",
         "members":[{ "name": "latitude", "rdfType": "sche-
ma:latitude",
          "valueType" : "number"},
      { "name": "longitude", "rdfType": "schema:longitude",
        "valueType" : "number"}]},
     { "name": "nodeEnd", "rdfType": "mobility:to",
        "members":[{ "name": "latitude", "rdfType": "sche-
ma:latitude",
        "valueType" : "number"},
     { "name": "longitude", "rdfType": "schema:longitude",
        "valueType" : "number"}]}]},
     { "name": "speed", "rdfType": "mobility:speed",
        "valueType" : "number"},
     { "name": "durationTime", "rdfType": "mobility:duration",
        "valueType" : "xsd:time"}
  ],
  "endpoints": {
      "url": "http://localhost:8080/bigiot/BCN-
             01/GetTrafficInformationOffering",
      "method": "HTTP_GET",
      "mediaType":"application/json",
      "accessInterface": "BIGIOT_LIB"
  },
  "license": "OPEN_DATA_LICENSE",
  "price": {
     "amount": "0.002000",
     "currency": "EUR",
     "accounting":"PER_ACCESS"
  },
  "region": "http://sws.geonames.org/3128760/"
}
```

- **Monitor Traffic Alarm Offering**

Table 12 Monitor Traffic Alarm Offering (JSON_LD)

Monitor Traffic Alarm Offering (JSON_LD)

```
{
  "@context": ["http://big-iot.eu/ctx",
  {"mobility" : "http://big-iot.eu/mobility#"}],

  "name": "Monitor Traffic Speed Offering",
  "category": "mobility:trafficNotificationsCategory",
  "providerId": "1452fwr",
  "inputData":[
      { "name": "trafficSpeed", "rdfType": "mobility:speed",
         "valueType" : "number"},
      { "name": "travelTime", "rdfType": "mobility:duration",
         "valueType" : "xsd:time"},
      { "name": "comparison", "rdfType": "mobility:Comparison",
         "valueType": "string"},
      { "name": "speedThreshold", "rdfType": "mobility:speed",
         "valueType" : "number"},
      { "name": "travelTimeThreshold", "rdfType": "mobili-
ty:duration",
         "valueType" : "xsd:time"},
      { "name": "timeIntervalFrom", "valueType": "xsd:dateTime"},
      { "name": "timeIntervalTo", "valueType": "xsd:dateTime"}],
  "outputData":[{"name": "trafficNotification",
    "rdfType": "mobility:TrafficStatusValue", "valueType" :
"string"}],
  "endpoints": {
  "url": "http://localhost:8080/bigiot/BCN-
         01/MonitorTrafficAlarmsOffering",
  "schema:type": "HTTP_POST",
  "mediaType":"application/json",
  "accessInterface": "BIGIOT_LIB"},
  "license": "OPEN_DATA_LICENSE",
  "price": {
     "amount": "0.002000",
     "currency": "EUR",
     "accounting":"PER_ACCESS"
  },
  "region": "http://sws.geonames.org/3128760/"
}
```

5.2. Semantic Offering Specifications of Platforms

5.2.1. Area-level Parking Space Information

Platform description

Different to the above examples, where the consumer has to access the data through the provided services, in the following the Worldsensing platform is taken as an example of a platform, which provides the direct access to its parking data. This offering gives the user the information of the number of available parking spots as well as the number of occupied parking spots. The offering is described as follows:

Table 13 Worldsensing Platform Offering overview

Offering properties	Description
Name	Area-level parking space information
Serviced Area	Barcelona city
Price	The price will be based on PER_ACCES accounting model. The cost of one access is 0.05 euro
Input Data	Coordinates: X; Y: (Double): center position, around which free parking spots shall be provided. Radius_meter (int): Radius in meter around center position (Buffer for Search)
Output Data	The parking space's name, number of available and number of occupied parking slot

Offering semantic description

Semantic Organization Description

Organization Semantic Description (JSON-LD)

```
{
  "@context": "http://big-iot.eu/core#",
  "name": "UPC",
  "organizationId": "orgaA",
  "schema:sourceOrganisation": "Barcelona_City",
}
```


Semantic Provider (Platform) Description

Provider Semantic Description (JSON-LD)


```
{
  "@context": "http://big-iot.eu/core#",
  "name": "World sensing platform",
  "providerId": "145sfwr",
  "schema:sourceOrganisation": "orgaA",
}
```


Semantic Offering Description

Table 14 Area-level Parking Space Information Offering (JSON-LD)

Area-level Parking Space Information Offering (JSON-LD)


```
{
  "@context": "http://big-iot.eu/ctx",
  "name": "Area-level parking space information",
  "category": "bigiot:Parking",
  "providerId": "1452fwr",
  "inputData": [
    { "name": "longitude", "valueType":
       "number","rdfType":"schema:latitude" },
    { "name": "latitude", "valueType": "number",
       "rdfType":"schema:longitude" },
    { "name": "radius", "valueType": "num-
ber","rdfType":"schema:geoRadius"
    }
  ],
  "outputData": [
    {name: "name", "rdfType": "bigiot:parkingName"},
    {name: "available", "rdfType":
"bigiot:parkingNumberOfVacantSpaces"},
    {name: "occupied", "rdfType":
"bigiot:parkingNumberOfOccupiedSpaces"}
  ],
  "endpoint": {
    url: "http://example.org/pspot",
```

```
    method: "HTTP_GET",
    accessInterfaceType: "BIGIOT_LIB"
  },
  "license": { "type": "OPEN_DATA_LICENSE" },
  "price": {
    "amount": 0.005,
    "currency": "EUR",
    "accounting": "PER_ACCESS"
  },
  "region": "http://sws.geonames.org/3128760/"
}
```

6. Conclusions and Future Work

In this deliverable we present the first iteration of our definition of the BIG IoT Application Domain Model that reflects the semantic terminologies needed to annotate the offerings for the realization of use cases implemented in BIG IoT pilots. This paves the road for the semantic description of platforms and services to be realized in WP5 and added to the BIG IoT ecosystem.

There is a growing understanding that shared approaches to semantic interoperability are the key to unlocking the full potential of the IoT in the various domains. Realizing semantic interoperability at scale will require collaboration and coordination across standard organizations, consortia, alliances, and open source projects. The need for a shared roadmap and commitment to work together seems self-evident.

An initial focus may be created around lightweight models of semantics sufficient to manage interoperability of common domain independent and domain specific terms. There is increasing interest in providing agile processes for standardizing such terms and defining accessible, usable schemes for discovery and reuse. Such schemes will need to address different stages of lifecycle and maturity, e.g. from experimental, to commercial implementations, to (eventual) deployment on a global scale.

An interesting precedent is provided by schema.org which defines a widely used lightweight RDF compatible vocabulary for websites to describe themselves to search engines. We propose to implement the same approach in defining the domain models that will be part of the BIG IoT Ecosystem.

What are the actual challenges?

- The models provide the basic description frameworks, but alignment between different models and frameworks would be required.
- Semantics are the starting point, reasoning and interpretation of data is required for automated processes.

- Real interoperability happens when data/services from different frameworks and providers can be interchanged and used with minimised intervention.

What are possible practical steps?

- Linked data approach is a promising way of integrating data from different sources and interlinking semantic descriptions.
- Study the alignment between different description models for IoT Services/Resources/Entities in different application domains.
- Proposing reference models for semantic descriptions in IoT and its domains of application.

This deliverable provides the basis for the semantic (meta-) data models used in the BIG IoT system. We build upon Linked Data principles and technologies to provide interoperable and platform agnostic data models that are based on existing ontologies and extend them when it is necessary. This allows BIG IoT applications to integrate other data sources in the Marketplace, resulting in a large and varied set of usable data items. Although we have analyzed a large number of ontologies during the design of the BIG IoT data models, the work is not finished. Firstly, for some system components like BIG IoT use case services, the actual data needed were not fully clear during the time of this deliverable's first version (D4.2.a). While these services have been developed, new data items are defined in this version. In addition, for other areas, e.g., trust and accounting, new data items and ontologies will be included. Secondly, there are other areas still not clear in some domains of application that will be integrated. Thus, it is envisaged that more domain models will be added as the project progresses.

This deliverable presents the results of the 1st iteration of the work related to the specification of domain models in the BIG IoT ecosystem, which has been carried out from M5 to M12 of the project. The work has been carried out by Task 4.2, but also involved interactions and coordination with representatives from other project activities, in particular Task 3.2 (Semantic interoperability for smart object platforms and services), with inputs from Task 2.3 (Requirements analysis and specification) and Task 5.1 (Pilot specification), which will be reflected in the work of Task 5.2 (Implementation of use case specific services and applications).

The 2nd and 3rd evolution of these application domain models will be reported in next deliverables D4.2.b and D4.2.c.

6.1. Future Work

The schema.org Community Group has been launched as part of W3C hosted Community Groups at https://www.w3.org/community/schemaorg/. This Community Group provides a forum for discussing all changes, additions and extensions to schema.org. In addition to providing a public setting for the day to day operation of the project, it serves as the mechanism for reviewing extensions and as a liaison point for all parties developing independent extensions to the schema.org core.

We are planning to take part in the schema.org process and groups to introduce BIG IoT inputs and extensions derived from our development and implementation into schema.org. In this way, not only we would be able to reach a large audience, but also engage with a wider community at international level that will be able to collaborate in the improvement of the vocabularies but also a wide audience of users and developers that would implement those solutions.

7. References

[1] M.Jacoby, A.Antonic, K.Kreiner, R.LŠapacz and J.Pielorz, "Semantic Interoperability as Key to IoT Platform Federation," InterOSS-IoT Workshop 2016

[2] M. Compton et al., "The SSN ontology of the W3C semantic sensor network incubator group," JWS, 2012.

[3] M3 ontology, http://sensormeasurement.appspot.com/

[4] Open Mobility Vocabulary — MobiVoc, www.mobivoc.org/

[5] DATEX II, www.datex2.eu

[6] The Open Content Provider Interface (OCPI), http://www.mobility.siemens.com/

[7] http://www.wsmo.org/

[8] WGS84 Geo Positioning, http://www.w3.org/2003/01/geo/wgs84_pos#

[9] The GeoNames Ontology, http://www.geonames.org/

[10] Time Ontology in OWL, https://www.w3.org/TR/owl-time/

8. ANNEX

8.1. BIG IoT Application Model in Turtle Format

Table 15 BIG IoT Application Model in the Turtle format

BIG IoT Application Model

```
# baseURI: http://big-iot.eu/mobility
# prefix: mobility
@prefix bigiot: <http://big-iot.eu/ns#> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix mobility: <http://big-iot.eu/mobility#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix p1: <http://example.org/file1#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://big-iot.eu/mobility>
rdf:type owl:Ontology ;
owl:versionInfo "Created with TopBraid Composer"^^xsd:string ;
.
mobility:AlertCArea
rdf:type owl:Class ;
dct:source <http://vocab.datex.org/terms/AlertCArea> ;
rdfs:subClassOf owl:Thing ;
.
mobility:Comparison
rdf:type owl:Class ;
rdfs:subClassOf mobility:Data ;
rdfs:subClassOf [
rdf:type owl:Class ;
owl:oneOf (
mobility:lessThanOrEqualTo
mobility:lessThan
mobility:greaterThanOrEqualTo
mobility:greaterThan
mobility:notEqualTo
mobility:equalTo
) ;
] ;
.
mobility:Data
rdf:type owl:Class ;
dct:source <http://vocab.datex.org/terms/BasicData> ;
rdfs:comment "Data that is either measured or calculated (elaborated)
```

```
at the same time or over the same time period."@en ;
     rdfs:subClassOf rdfs:Datatype ;
     rdfs:subClassOf bigiot:Data ;
     .
     mobility:DataValue
     rdf:type owl:Class ;
     dct:source <http://vocab.datex.org/terms/DataValue> ;
     rdfs:comment "A data value of something that can be measured or calcu-
lated. Any provided meta-data values specified in the attributes override
any specified generic characteristics such as defined for a specific meas-
urement in the MeasurementSiteTable. \"@en"^^xsd:string ;
     rdfs:subClassOf rdfs:Datatype ;
     rdfs:subClassOf bigiot:Data ;
     .
     mobility:Destination
     rdf:type owl:Class ;
     dct:source <http://vocab.datex.org/terms/Destination> ;
     rdfs:comment "The specification a destination. This may be either a
point location or an area location."@en ;
     rdfs:subClassOf owl:Thing ;
     .
     mobility:DurationValue
     rdf:type owl:Class ;
     dct:source <http://vocab.datex.org/terms/DurationValue> ;
     rdfs:subClassOf mobility:DataValue ;
     .
     mobility:Environment
     rdf:type owl:Class ;
     rdfs:subClassOf <http://big-iot.eu/core#OfferingCategory> ;
     rdfs:subClassOf [
     rdf:type owl:Class ;
     owl:oneOf (
     mobility:environment
     ) ;
     ] ;
     .
     mobility:GroupOfLocations
     rdf:type owl:Class ;
     dct:source <http://vocab.datex.org/terms/GroupOfLocations> ;
     rdfs:comment "\"One or more physically separate locations. Multiple
locations may be related, as in an itinerary (or route), or may be unrelat-
ed. It is not for identifying the same physical location using different
Location objects for different referencing systems\". "^^xsd:string ;
     rdfs:subClassOf owl:Thing ;
     .
     mobility:Itinerary
     rdf:type owl:Class ;
     dct:source <http://vocab.datex.org/terms/Itinerary> ;
     rdfs:comment "\"Multiple (i.e. more than one) physically separate lo-
cations arranged as an ordered set that defines an itinerary or route.\"@en
;"^^xsd:string ;
     rdfs:subClassOf mobility:GroupOfLocations ;
     .
     mobility:ItineraryByIndexedLocations
```

```
     rdf:type owl:Class ;
     dct:source <http://vocab.datex.org/terms/ItineraryByIndexedLocations>
;
     rdfs:subClassOf mobility:Itinerary ;
     .
     mobility:Location
     rdf:type owl:Class ;
     rdfs:comment "The specification of a location either on a network (as
a point or a linear location) or as an area. This may be provided in one or
more referencing systems."@en ;
     rdfs:subClassOf owl:Thing ;
     .
     mobility:LocationContainedInItinerary
     rdf:type owl:Class ;
     dct:source <http://vocab.datex.org/terms/LocationContainedInItinerary>
;
     rdfs:subClassOf owl:Thing ;
     .
     mobility:NetworkLocation
     rdf:type owl:Class ;
     rdfs:subClassOf mobility:Location ;
     .
     mobility:Parking
     rdf:type owl:Class ;
     rdfs:subClassOf <http://big-iot.eu/core#OfferingCategory> ;
     rdfs:subClassOf [
     rdf:type owl:Class ;
     owl:oneOf (
     mobility:parking
     mobility:parkingSiteMgmtCategory
     mobility:parkingSpaceMgmtCategory
     ) ;
     ] ;
     .
     mobility:ParkingSite
     rdf:type owl:Class ;
     mobility:parkingSpaceOrGroupIdentifier mobili-
ty:ParkingSpaceOrGroupIdentifier ;
     dct:source <http://vocab.datex.org/terms/ParkingSite> ;
     rdfs:comment "A record containing static details of a parking site.
Must be specialised as an 'Urban-' or 'InterUrbanParkingSite' or a
'SpecialLocationParkingSite'."@en ;
      bigiot:refersTo <http://big-iot.eu/mobility#parkingSiteMgmtCategory>
;
     rdfs:subClassOf owl:Thing ;
     .
     mobility:ParkingSiteStatus
     rdf:type owl:Class ;
     dct:source <http://vocab.datex.org/terms/ParkingSiteStatus> ;
     rdfs:subClassOf mobility:Data ;
     .
     mobility:ParkingSpace
     rdf:type owl:Class ;
     mobility:parkingSpaceOrGroupIdentifier mobili-
```

```
ty:ParkingSpaceOrGroupIdentifier ;
     dct:source <http://vocab.datex.org/terms/ParkingSpace> ;
     rdfs:comment "A single parking space. It is possible to define the
same parking space more than once with different properties, e.g. when
there is a different parking assignment for different times."@en ;
     bigiot:refersTo <http://big-iot.eu/mobility#parkingSpaceMgmtCategory>
;
     rdfs:subClassOf owl:Thing ;
     .
     mobility:ParkingSpaceOrGroupIdentifier
     rdf:type owl:Class ;
     rdfs:subClassOf mobility:Data ;
     .
     mobility:ParkingSpaceStatus
     rdf:type owl:Class ;
     dct:source <http://vocab.datex.org/terms/ParkingSpaceStatus> ;
     rdfs:comment "Status (occupied or closed) for a single parking space
which was defined in the static part of the model."@en ;
     rdfs:subClassOf mobility:DataValue ;
     rdfs:subClassOf [
     rdf:type owl:Class ;
     owl:oneOf (
     mobility:occupiedSpaceAppeared
     mobility:occupiedSpaceDisappeared
     mobility:potentialSpaceAppeared
     mobility:potentialSpaceDisappeared
     ) ;
     ] ;
     .
     mobility:PaymentMethodForParking
     rdf:type owl:Class ;
     dct:source <http://vocab.datex.org/terms/PaymentMethodForParking> ;
     rdfs:subClassOf mobility:Data ;
     .
     mobility:ReservationStatus
     rdf:type owl:Class ;
     rdfs:comment "Gives the status \"reserved\" or \"not reserved\", after
reserving a parking space."@en ;
     rdfs:subClassOf mobility:Data ;
     .
     mobility:SpeedValue
     rdf:type owl:Class ;
     dct:source <http://vocab.datex.org/terms/SpeedValue> ;
     rdfs:subClassOf mobility:DataValue ;
     .
     mobility:TpegAreaDescriptor
     rdf:type owl:Class ;
     dct:source <http://vocab.datex.org/terms/TpegAreaDescriptor> ;
     rdfs:subClassOf owl:Thing ;
     .
     mobility:TpegAreaLocation
     rdf:type owl:Class ;
     dct:source <http://vocab.datex.org/terms/TpegAreaLocation> ;
     rdfs:subClassOf mobility:Data ;
```

```
     .
     mobility:TpegGeometricArea
     rdf:type owl:Class ;
     mobility:name mobility:TpegAreaDescriptor ;
     dct:source <http://vocab.datex.org/terms/TpegGeometricArea> ;
     schema:geoMidpoint schema:GeoCoordinates ;
     rdfs:subClassOf mobility:TpegAreaLocation ;
     .
     mobility:TpegLinearLocation
     rdf:type owl:Class ;
     mobility:from schema:GeoCoordinates ;
     mobility:to schema:GeoCoordinates ;
     rdfs:subClassOf mobility:Data ;
     .
     mobility:TpegNamedOnlyArea
     rdf:type owl:Class ;
     dct:source <http://vocab.datex.org/terms/TpegNamedOnlyArea> ;
     rdfs:subClassOf mobility:TpegAreaLocation ;
     .
     mobility:Traffic
     rdf:type owl:Class ;
     rdfs:subClassOf <http://big-iot.eu/core#OfferingCategory> ;
     rdfs:subClassOf [
     rdf:type owl:Class ;
     owl:oneOf (
     mobility:traffic
     mobility:trafficInformationCategory
     mobility:trafficMonitoringCategory
     ) ;
     ] ;
     .
     mobility:TrafficData
     rdf:type owl:Class ;
     dct:source <http://vocab.datex.org/terms/TrafficData> ;
     rdfs:comment "Measured or derived values relating to traffic or indi-
vidual vehicle movements on a specific section or at a specific point on
the road network."@en ;
     rdfs:subClassOf mobility:Data ;
     .
     mobility:TrafficElement
     rdf:type owl:Class ;
     dct:source <http://vocab.datex.org/terms/TrafficElement> ;
     rdfs:comment "An event which is not planned by the traffic operator,
which is affecting, or has the potential to affect traffic flow."@en ;
     rdfs:subClassOf owl:Thing ;
     .
     mobility:TrafficSpeed
     rdf:type owl:Class ;
     dct:source <http://vocab.datex.org/terms/TrafficSpeed> ;
     rdfs:comment "Averaged measurements or calculations of traffic
speed."@en ;
     rdfs:subClassOf mobility:TrafficData ;
     .
     mobility:TrafficStatus
```

```
    rdf:type owl:Class ;
    dct:source <http://vocab.datex.org/terms/TrafficStatus> ;
    rdfs:subClassOf mobility:Data ;
    .
    mobility:TrafficStatusValue
    rdf:type owl:Class ;
    dct:source <http://vocab.datex.org/terms/TrafficStatusValue> ;
    rdfs:subClassOf mobility:DataValue ;
    rdfs:subClassOf [
    rdf:type owl:Class ;
    owl:oneOf (
    mobility:congested
    mobility:freeFlow
    mobility:heavy
    mobility:impossible
    mobility:unknown
    ) ;
    ] ;
    .
    mobility:Transportation
    rdf:type owl:Class ;
    rdfs:subClassOf <http://big-iot.eu/core#OfferingCategory> ;
    rdfs:subClassOf [
    rdf:type owl:Class ;
    owl:oneOf (
    mobility:transportation
    mobility:busTransportation
    mobility:carTransportation
    ) ;
    ] ;
    .
    mobility:TravelTimeData
    rdf:type owl:Class ;
    dct:source <http://vocab.datex.org/terms/TravelTimeData> ;
    rdfs:comment "Derived/computed travel time information relating to a
linear section of the road network; forecast = true means a forecast for a
vehicle at the start of the specified location, forecast = false means cal-
culation/measurement at the end."@en ;
    rdfs:subClassOf mobility:Data ;
    .
    mobility:Vehicle
    rdf:type owl:Class ;
    dct:source <http://vocab.datex.org/terms/Vehicle> ;
    rdfs:comment "Details of an individual vehicle."@en ;
    rdfs:subClassOf owl:Thing ;
    .
    mobility:alertCArea
    rdf:type owl:ObjectProperty ;
    dct:source <http://vocab.datex.org/terms/alertCArea> ;
    rdfs:domain mobility:Location ;
    rdfs:range mobility:AlertCArea ;
    .
    mobility:averageVehicleSpeed
    rdf:type owl:ObjectProperty ;
```

```
    dct:source <http://vocab.datex.org/terms/averageVehicleSpeed> ;
    rdfs:domain mobility:TrafficSpeed ;
    rdfs:range mobility:SpeedValue ;
    .
    mobility:busTransportation
    rdf:type mobility:Transportation ;
    .
    mobility:carTransportation
    rdf:type mobility:Transportation ;
    .
    mobility:congested
    rdf:type mobility:TrafficStatusValue ;
    dct:source <http://vocab.datex.org/terms/congested> ;
    .
    mobility:duration
    rdf:type owl:DatatypeProperty ;
    dct:source <http://vocab.datex.org/terms/duration> ;
    rdfs:domain mobility:DurationValue ;
    rdfs:range xsd:time ;
    .
    mobility:environment
    rdf:type mobility:Environment ;
    .
    mobility:equalTo
    rdf:type mobility:Comparison ;
    .
    mobility:freeFlow
    rdf:type mobility:TrafficStatusValue ;
    dct:source <http://vocab.datex.org/terms/freeFlow> ;
    .
    mobility:from
    rdf:type owl:ObjectProperty ;
    rdfs:range schema:GeoCoordinates ;
    .
    mobility:greaterThan
    rdf:type mobility:Comparison ;
    .
    mobility:greaterThanOrEqualTo
    rdf:type mobility:Comparison ;
    .
    mobility:groupOfLocations
    rdf:type owl:ObjectProperty ;
    dct:source <http://vocab.datex.org/terms/groupOfLocations> ;
    rdfs:domain mobility:GroupOfLocations ;
    .
    mobility:hasParkingIdentifier
    rdf:type owl:DatatypeProperty ;
    rdfs:domain mobility:ParkingSpaceOrGroupIdentifier ;
    rdfs:range xsd:string ;
    .
    mobility:heavy
    rdf:type mobility:TrafficStatusValue ;
    dct:source <http://vocab.datex.org/terms/heavy> ;
    .
```

```
    mobility:impossible
    rdf:type mobility:TrafficStatusValue ;
    dct:source <http://vocab.datex.org/terms/impossible> ;
    .
    mobility:lessThan
    rdf:type mobility:Comparison ;
    .
    mobility:lessThanOrEqualTo
    rdf:type mobility:Comparison ;
    .
    mobility:location
    rdf:type owl:ObjectProperty ;
    rdfs:domain mobility:LocationContainedInItinerary ;
    rdfs:range mobility:Location ;
    .
    mobility:locationContainedInItinerary
    rdf:type owl:ObjectProperty ;
    dct:source <http://vocab.datex.org/terms/locationContainedInItinerary>
;
    rdfs:domain mobility:ItineraryByIndexedLocations ;
    rdfs:range mobility:LocationContainedInItinerary ;
    .
    mobility:methodOfPaymentForParking
    rdf:type owl:ObjectProperty ;
    dct:source <http://vocab.datex.org/terms/methodOfPaymentForParking> ;
    rdfs:range mobility:PaymentMethodForParking ;
    .
    mobility:name
    rdf:type owl:ObjectProperty ;
    dct:source <http://vocab.datex.org/terms/name> ;
    rdfs:domain mobility:TpegNamedOnlyArea ;
    rdfs:range mobility:TpegAreaDescriptor ;
    .
    mobility:notEqualTo
    rdf:type mobility:Comparison ;
    .
    mobility:notReserved
    rdf:type mobility:ReservationStatus ;
    .
    mobility:occupiedSpaceAppeared
    rdf:type mobility:ParkingSpaceStatus ;
    .
    mobility:occupiedSpaceDisappeared
    rdf:type mobility:ParkingSpaceStatus ;
    .
    mobility:parking
    rdf:type mobility:Parking ;
    .
    mobility:parkingDescription
    rdf:type owl:ObjectProperty ;
    .
    mobility:parkingLocation
    rdf:type owl:ObjectProperty ;
    rdfs:domain mobility:ParkingSpace ;
```

```
    rdfs:range schema:Place ;
    .
    mobility:parkingNumberOfVacantSpaces
    rdf:type owl:DatatypeProperty ;
    dct:source <http://vocab.datex.org/terms/parkingNumberOfVacantSpaces>
;
    rdfs:domain mobility:ParkingSiteStatus ;
    rdfs:label "parking number of vacant spaces"^^xsd:string ;
    rdfs:range xsd:integer ;
    .
    mobility:parkingSiteMgmtCategory
    rdf:type mobility:Parking ;
    .
    mobility:parkingSiteStatus
    rdf:type owl:ObjectProperty ;
    dct:source <http://vocab.datex.org/terms/parkingSiteStatus> ;
    rdfs:domain mobility:ParkingSite ;
    rdfs:range mobility:ParkingSiteStatus ;
    .
    mobility:parkingSpaceMgmtCategory
    rdf:type mobility:Parking ;
    .
    mobility:parkingSpaceOrGroupIdentifier
    rdf:type owl:ObjectProperty ;
    dct:source
<http://vocab.datex.org/terms/parkingSpaceOrGroupIdentifier> ;
    .
    mobility:parkingSpaceStatus
    rdf:type owl:ObjectProperty ;
    dct:source <http://vocab.datex.org/terms/parkingSpaceStatus> ;
    rdfs:domain mobility:ParkingSpace ;
    rdfs:range mobility:ParkingSpaceStatus ;
    .
    mobility:parkingSpaceStatusTimeStamp
    rdf:type owl:DatatypeProperty ;
    rdfs:domain mobility:ParkingSpace ;
    rdfs:range xsd:dateTime ;
    .
    mobility:payAndDisplay
    rdf:type mobility:PaymentMethodForParking ;
    dct:source <http://vocab.datex.org/terms/payAndDisplay> ;
    rdfs:comment "Pay at machine and display ticket inside vehi-
cle."^^xsd:string ;
    .
    mobility:payAtExitBoothManualCollection
    rdf:type mobility:PaymentMethodForParking ;
    dct:source
<http://vocab.datex.org/terms/payAtExitBoothManualCollection> ;
    rdfs:comment "Pay at the manned exit booth of the parking facili-
ty."^^xsd:string ;
    .
    mobility:payAtMachineOnFootPriorToExit
    rdf:type mobility:PaymentMethodForParking ;
    dct:source
```

```
<http://vocab.datex.org/terms/payAtMachineOnFootPriorToExit> ;
     rdfs:comment "Pay at machine on foot prior to returning to vehicle and
use payment ticket to exit."^^xsd:string ;
     .
     mobility:payByPrepaidToken
     rdf:type mobility:PaymentMethodForParking ;
     dct:source <http://vocab.datex.org/terms/payByPrepaidToken> ;
     rdfs:comment "Pay by prepaid token which is used at exit."^^xsd:string
;
     .
     mobility:potentialSpaceAppeared
     rdf:type mobility:ParkingSpaceStatus ;
     dct:source <http://vocab.datex.org/terms/potentialSpaceAppeared> ;
     .
     mobility:potentialSpaceDisappeared
     rdf:type mobility:ParkingSpaceStatus ;
     dct:source <http://vocab.datex.org/terms/potentialSpaceDisappeared> ;
     .
     mobility:publicationTime
     rdf:type owl:DatatypeProperty ;
     dct:source <http://vocab.datex.org/terms/publicationTime> ;
     rdfs:range xsd:dateTime ;
     .
     mobility:reservationFee
     rdf:type owl:DatatypeProperty ;
     rdfs:range xsd:float ;
     .
     mobility:reserved
     rdf:type mobility:ReservationStatus ;
     .
     mobility:speed
     rdf:type owl:DatatypeProperty ;
     dct:source <http://vocab.datex.org/terms/speed> ;
     rdfs:domain mobility:SpeedValue ;
     rdfs:range xsd:float ;
     .
     mobility:to
     rdf:type owl:ObjectProperty ;
     rdfs:range schema:GeoCoordinates ;
     .
     mobility:totalCapacity
     rdf:type owl:DatatypeProperty ;
     dct:source <http://vocab.datex.org/terms/totalCapacity> ;
     rdfs:domain mobility:ParkingSite ;
     rdfs:range xsd:integer ;
     .
     mobility:tpegAreaDescriptorType
     rdf:type owl:DatatypeProperty ;
     dct:source <http://vocab.datex.org/terms/tpegAreaDescriptorType> ;
     rdfs:domain mobility:TpegAreaDescriptor ;
     rdfs:range xsd:string ;
     .
     mobility:tpegAreaLocation
     rdf:type owl:ObjectProperty ;
```

```
     dct:source <http://vocab.datex.org/terms/tpegAreaLocation> ;
     rdfs:domain mobility:Location ;
     rdfs:range mobility:TpegAreaLocation ;
     .
     mobility:tpegLinearLocation
     rdf:type owl:ObjectProperty ;
     rdfs:domain mobility:NetworkLocation ;
     .
     mobility:traffic
     rdf:type mobility:Traffic ;
     .
     mobility:trafficInformationCategory
     rdf:type mobility:Traffic ;
     .
     mobility:trafficMonitoringCategory
     rdf:type mobility:Traffic ;
     .
     mobility:trafficSpeedofTpegAreaLocation
     rdf:type owl:ObjectProperty ;
     dct:source
<http://vocab.datex.org/terms/trafficSpeedofTpegAreaLocation> ;
     rdfs:domain mobility:TpegAreaLocation ;
     rdfs:range mobility:TrafficSpeed ;
     .
     mobility:trafficStatus
     rdf:type owl:ObjectProperty ;
     dct:source <http://vocab.datex.org/terms/trafficStatus> ;
     rdfs:domain mobility:TrafficStatus ;
     rdfs:range mobility:TrafficStatusValue ;
     .
     mobility:transportation
     rdf:type mobility:Transportation ;
     .
     mobility:travelTime
     rdf:type owl:ObjectProperty ;
     dct:source <http://vocab.datex.org/terms/travelTime> ;
     rdfs:domain mobility:TravelTimeData ;
     rdfs:range mobility:DurationValue ;
     .
     mobility:travelTimeofTpegAreaLocation
     rdf:type owl:ObjectProperty ;
     dct:source <http://vocab.datex.org/terms/travelTimeofTpegAreaLocation>
;
     rdfs:domain mobility:TpegAreaLocation ;
     rdfs:range mobility:TravelTimeData ;
     .
     mobility:unknown
     rdf:type mobility:TrafficStatusValue ;
     dct:source <http://vocab.datex.org/terms/unknown> ;
     .
     mobility:vehicleType
     rdf:type owl:ObjectProperty ;
     dct:source <http://purl.org/net/mobivoc/vehicleType> ;
     rdfs:range mobility:Vehicle ;
```

```
.
dct:source
rdf:type owl:DatatypeProperty ;
rdfs:range xsd:anyURI ;
.
schema:GeoCircle
rdf:type owl:Class ;
rdfs:subClassOf schema:GeoShape ;
.
schema:GeoCoordinates
rdf:type owl:Class ;
rdfs:subClassOf mobility:DataValue ;
.
schema:GeoShape
rdf:type owl:Class ;
rdfs:subClassOf mobility:DataValue ;
.
schema:Place
rdf:type owl:Class ;
schema:geo [
rdf:type owl:Class ;
owl:unionOf (
schema:GeoCoordinates
schema:GeoShape
) ;
] ;
rdfs:subClassOf owl:Thing ;
.
schema:geo
rdf:type owl:ObjectProperty ;
.
schema:geoMidpoint
rdf:type owl:ObjectProperty ;
rdfs:domain schema:GeoCircle ;
rdfs:range schema:GeoCoordinates ;
.
schema:geoRadius
rdf:type owl:DatatypeProperty ;
dct:source <http://vocab.datex.org/terms/radius> ;
rdfs:range xsd:double ;
.
schema:latitude
rdf:type owl:DatatypeProperty ;
rdfs:domain schema:GeoCoordinates ;
rdfs:range xsd:double ;
.
schema:longitude
rdf:type owl:DatatypeProperty ;
rdfs:domain schema:GeoCoordinates ;
.
schema:supersededBy
rdf:type owl:ObjectProperty ;
.
```