Modelling with NGSI-LD: the VALLPASS project case study

Tiago Ribeiro*, João Paulo Coelho*, Luísa Jorge*†, Joaquim Sardão‡, José Gonçalves*, Higor Rosse§

* CeDRI - Research Center for Digitization and Intelligent Robotics
SusTEC - Associate Laboratory for Sustainability and Technology in Mountains Regions
Bragança, Portugal
Email: {tiagoribeiro, goncalves, jpcoelho, ljorge}@ipb.pt

† INESC Coimbra, University of Coimbra,
Department of Electrical and Computer Engineering, Polo 2, 3030-290 Coimbra, Portugal

‡ Valled Lda, Bragança, Portugal
Email: joaquim.sardao@valled.pt

§ Intelligent Technologies
Collaborative Laboratory Mountains of Research
Bragança, Portugal
Email: hrosse@morecolab.pt

2023 IEEE 21st International Conference on Industrial Informatics (INDIN) | 978-1-6654-9313-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/INDIN51400.2023.10218110

***Abstract*—The smart cities paradigm covers multiple domains which span from citizens' accessibility and mobility to general infrastructures and services. Hence, smart cities can be seen as an excellent showcase of heterogeneity, namely at the data level. For this reason, they are a perfect candidate for linked data and semantic web concept applications. This powerful combination leads to interoperability at the data level which is one of the ultimate goals of the Internet of Things (IoT). In this reference frame, NGSI-LD is an open framework for context information processing consisting of both a semantic information model and a RESTful Application Programming Interface (API). This paper proposes a methodology for creating semantic data models in the context of IoT, namely to represent and describe data associated with digital twins. The methodology is presented in a practical way, through the process of creating an NGSI-LD semantic data model for the VALLPASS project, inserted in the traffic domain, which is one of the most popular in smart cities.**

***Index Terms*—VALLPASS project, NGSI-LD, FIWARE, data models, smart cities.**

I. Introduction

Smart cities have gone beyond the realm of literature and have become a reality, with the market size reaching $457 billion in 2021 [1]. Despite its popularity and array of applications, even today the concept of smart cities is vague and framed inconsistently as a function of the domain in which it is used [2].

According to Hall's definition [3], a smart city is "*a city that monitors and integrates conditions of all of its critical infrastructures, including roads, bridges, tunnels, rail/subways, airports, seaports, communications, water, power, even major buildings, can better optimise its resources, plan its preventive maintenance activities, and monitor security aspects while maximising services to its citizens.*"

As pointed out in [4], generating large amounts of data does not necessarily translate into the creation of any value or services appreciated by citizens. It is necessary to integrate data from different domains and numerous sources and be able to extract meaningful information from them. Achieving this integration (or interoperability) is a complex task since two types of integration must be achieved: syntactic, related to content type, and semantic, associated with the contextual meaning.

In the Internet of Things (IoT), heterogeneity doesn't just apply to the data domain. There is also a myriad of hardware, protocols, platforms, and policies, so syntactic interoperability is the first challenge [5]. Once the data is encapsulated in a commonly accepted format, it is necessary to annotate it semantically. That is, to link it to ontological concepts so that it has significance outside the domain from which it originated.

Linked data (LD) and semantic web technologies (SWT) are a powerful combination that promotes the desired interoperability. On one hand, LD allows data to be integrated into a common and navigable conceptual graph while leaving data distributed and managed in different systems (loosely integrated). On the other, with SWT, meaningful, well-structured data models can be applied in the form of shared ontologies and Web vocabularies.

Developing semantic models for the IoT can be quite tricky, not only because of the practical novelty of SWT, but also because of the inevitable mismatches between well-known ontologies and the scenarios being modeled, which sometimes lead to decisions that are not in line with SWT principles.

This work advances a methodology for creating semantic data models within the Internet of Things (IoT) framework. Specifically, it focuses on representing and describing data related to digital twins. The methodology is demonstrated practically by creating an NGSI-LD semantic data model for the VALLPASS project, which is situated in the traffic domain, widely recognized as a prominent area in the context of smart cities [4]. The remainder of this paper is organised as follows: Section II presents the results concerning the review of related works, supported by SWT, already published in the scientific literature. Section III introduces the fundamental concepts associated with SWT while aiming at the NGSI-LD data model of the VALLPASS project. Details about this project will be put forward in Section IV where a special emphasis on its data model will be given. Section V demonstrates the systematic approach taken in developing the NGSI-LD semantic data model for the VALLPASS project. Section VI presents the verdict of the integration of the semantic data model into the VALLPASS system. This paper's final remarks and conclusions will be provided in Section VII.

II. Related work

Although the concept of the SWT was introduced to the world in 1999 by Berners-Lee [6], it is only in the last few years that it has begun to gain traction in the real world and has reached maturity.

As far as smart cities are concerned, researchers are studying and implementing models [7], [8], frameworks [9], [10] and architectures [11]–[13] supported by SWT. More specifically in the scope of this article, three case studies of projects supported by the NGSI-LD specification are highlighted.

The first, by Jeong *et al.* [14], presented the City Data Hub platform, introduced in a Korean smart city project. This project was characterised by data-level interoperability achieved through the use of NGSI-LD interfaces and the definition of data models compliant with the NGSI-LD specification. However, in the paper, the data models used were not presented.

In the field of smart agriculture, López-Morales [15] has proposed an open and interoperable platform for irrigation community management. This platform was based on standard and open interfaces and protocols, where the NGSI-LD specification is included. Regarding the data model adopted for the platform, he pointed out that the existing data models were too specialised and, even if they covered the information to be handled, their implementation was time-consuming. For this reason, a new data model was developed, based on the harmonized data models proposed by FIWARE and supported by the requirements established by the stakeholders. The newly developed data model has been presented and validated, however, the underlying creation process has not been described.

In the context of the CityIoT project, which is a vendor-independent IoT platform for smart cities, the NGSI-LD "ThreePhaseACMeasurement" data model was used to represent electrical measurements from a system using a three-phase alternating current. This model, after being developed, was accepted as an official FIWARE data model [16], [17].

III. The FIWARE framework

FIWARE is an open-source initiative that works towards defining and implementing a set of standards to develop smart solutions for different domains, such as Smart Cities, Smart Agri-Food, Smart Industries, among others [18].

Intelligent applications feed themselves with data from different sources about events relevant to them. This context information, after being processed, visualised, and analysed, can drive intelligent behaviours [19].

It is in the context-data management activities that FIWARE intervenes, promoting a standard that describes how to collect, manage and publish this context information and, additionally, adding certain elements to exploit this collected data [19].

At the architectural level, FIWARE can be seen as a framework of software platform components that can be used in conjunction with, and even with third-party components, to build platforms that support the development of intelligent solutions [20]. These components are called "Generic Enablers" (GEs) [21].

The main component of the FIWARE framework is the "FIWARE Context Broker GE" and is the only mandatory component for an application to be a "Powered by FIWARE" solution [20]. It has the function of managing context information, which is a crucial and cross-cutting need for any intelligent solution [20]. This component exports an API called "FIWARE NGSI" which enables components integration within a "Powered by FIWARE" platform and provides a means for applications to interact with context information [20]. The specifications of this API are currently aligned with the NGSI-LD standard from ETSI (European Telecommunications Standards Institute) [20].

Wrapped around the "FIWARE Context Broker GE", the FIWARE framework provides other GEs that act in different domains from interfacing with IoT, robots and third-party systems to processing, analysing and visualising context information [20].

FIWARE prioritizes interoperability through a standardized data model, enabling effective utilization of data by all components [21], while its distributed data management capabilities facilitate integration with other platforms [18]. Additionally, FIWARE maintains retrocompatibility by unifying data models and offering data format conversion for compatibility with existing systems [21].

A. Smart Data Models Initiative

The Smart Data Models Initiative stems from a collaboration between the FIWARE Foundation, TM Forum, IUDX and the OASC, and aims to support the adoption of a common compatible reference architecture and data models that support interoperable and replicable smart solutions across multiple sectors, starting with Smart Cities [22].

An Smart Data Model consists of three elements: the schema, the specification, and the examples [22]. The schema is the technical representation of the model and is where its structure and data types are defined [22]. The specification consists of a written document oriented to human readers - documentation [22]. The examples are payloads illustrating the use of the models and are available for the NGSIv2 and NGSI-LD specifications. All data models are free to use, free to modify, and allow free sharing of modifications [22].

Regarding their structure, the data models are grouped into subjects, assigned to a GIT repository, and these can be associated with one or several domains, representing industry sectors [22].

Smart cities are a very important FIWARE domain, and there are already more than 250 cities using FIWARE technology in various use cases such as smart tourism and mobility [23]. It could even be said that FIWARE is becoming the *de facto* standard for smart cities. As a matter of fact, using common standard APIs and information models has several associated advantages in developing smart solutions for smart cities. For example, increases in interoperability, enhanced portability and reusability, faster time-to-market, integrated security and fosters collaboration and co-creation between cities and countries.

Sharing structured information between different stakeholders, and promoting open data exchange is key in any smart data model and nuclear to FIWARE. In this framework, NGSI-LD appears as an information model that supports all the above requirements.

B. The NGSI-LD information model

First of all, it is important to understand the concept of context. The IoT is a cluster of entities - physical and non-physical. Examples of entities are the various sensors and actuators employed in any IoT practical solution. The context comprises all the characteristics, states, and other dynamic properties of these entities cumulatively with the important relationships that represent the real and virtual links between them [24]. In short, any information that characterises the situation of an entity is its context.

NGSI-LD is an open framework for context information processing developed by the ETSI Industry Specification Group for cross-cutting Context Information Management (ISG CIM) [25]. The term NGSI is related to previous work developed by the Open Mobile Alliance and served as inspiration for the term LD which represents the strong influence of Linked Data concepts [25]. It consists of a semantic information model and a RESTful API.

The NGSI-LD information model facilitates the modelling of real-world entities, relationships and properties, and can link and federate other information models through the use of JSON-LD [25]. It is also compatible with RDF [25]. In NGSI-LD, data takes the form of a graph of correspondence links between informational units that correspond to real-world entities - graph data model [24]. The semantic referencing used by NGSI-LD is based on standard RDF/RDS/OWL typing and public ontologies [24]. All nodes and edges of the graph are thus matched to the various classes of these ontologies that jointly characterise the features shared by all instances of these classes [24].

The main constructs defined in the NGSI-LD meta-model are:

a) **Entity**: Informational representative of something that is supposed to exist in the real world, physically or conceptually [24]. Any instance of an entity must be uniquely identified by a Uniform Resource Identifier (URI) and characterised by one or more Entity Type(s) such as classes in Web Ontology Language (OWL) [24].
b) **Property**: An description instance that associates a main characteristic - Value - with an Entity, Relationship or other Property [24].
c) **Value** It can take one of three forms: JSON value, JSON-LD typed value, or JSON-LD structured value [24].
d) **Relationship** It represents a direct link between a subject which can be an Entity, Property or other Relationship and an object, which must be an Entity [24].

Messages related to API operations are expressed using the JSON-LD format, with any entity being represented by a JSON-LD encoded object [25].

IV. THE VALLPASS PROJECT

Mobility is one of the foundation stones of human development, and improving its safety, efficiency, and sustainability is always on the agenda. Crosswalks play a key role in urban mobility, but they bring challenges, particularly related to safety. Between 2010 and 2018 more than 51,000 pedestrians died in Europe [26].

The VALLPASS project - Intelligent Active Surveillance with LoRa Support for Pedestrian Crossings - aims to develop a system used in crosswalks that ensures the safe crossing of pedestrians, and, to this end, is equipped with a set of features and technological innovations, presented below:

a) **Energetically self-sufficient**: The VALLPASS system is isolated from the electrical grid, generating its energy through solar panels and employing intelligent energy management techniques, thus ensuring that it can be installed virtually anywhere.
b) **Self-commissioning**: The VALLPASS system only has to be placed on the crosswalks, dispensing with any configuration by the installers - plug-and-play installation. Algorithms based on self-organisation techniques will coordinate their connection and operationalisation.
c) **Use of self-maintenance techniques**: The VALLPASS system employs self-maintenance techniques and algorithms to increase its resilience and reduce its operational cost.
d) **Cutting-edge road safety features**: VALLPASS crossings have a “luminous tunnel” for the safe crossing of pedestrians and an efficient warning system directed at vehicle drivers whenever pedestrians are crossing the road.

e) **Remote management platform "Powered by FIWARE"**: A remote management platform supported by the NGSI-LD specification enables customers to efficiently manage the system on a global basis, with emphasis on the presentation of relevant telemetry and statistics.

The VALLPASS system consists of a set of energy-autonomous lampposts distributed along urban regions at pedestrian crosswalks that are not controlled by traffic lights. Two lampposts, one at each end, are installed at each crosswalk. In addition to a high-efficiency LED luminaire and an energy management system, each pole includes a myriad of sensors. In particular, a camera with integrated artificial intelligence to detect pedestrians and a radar to detect vehicle speeds. Regarding the telemetry and data transmission infrastructure, a LoRaWAN network architecture is used with the poles.

In the end, the VALLPASS system will be commercialised by VALLED, a company located in the Northeast region of Portugal. Future clients, such as town halls and other governmental institutions, will be able to purchase the system for one or several crosswalks with each crosswalk having 2 poles. Each pole has a LED lighting fixture and a power supply unit composed of a battery, a solar panel and a charging controller. Additionally, a custom-made microcontroller-based device interfaces with a set of sensors and will be responsible for data transmission.

Regarding the sensors, they will be identified in the data model as follows:

a) **WeatherSensor** It monitors the temperature, relative humidity, atmospheric pressure, and brightness.
b) **PedestrianSensor** It monitors the flow of pedestrians in a given period.
c) **VehicleSensor** It monitors the flow of vehicles over a given period and tells the average, maximum and minimum speed associated with the flow.
d) **AccidentSensor** An accelerometer that monitors the structural integrity of the pole.

On the other hand, the LED luminaire data model is represented by two different entities:

a) **LuminaireModel**: It represents the static attributes of the luminaire, namely its technical characteristics.
b) **Luminaire**: Represents the operational status of the luminaire.

Similar to the luminaire, two entities are also used to represent the battery:

a) **BatteryModel**: Represents the technical characteristics of the battery (static attributes).
b) **BatteryMeasurement**: It represents the dynamic attributes of the battery, namely its temperature, its state of charge (SoC) and its state of health (SoH).

It is important to notice that in the current discussion, the word "sensor" refers to virtual rather than real devices. For example, the *WeatherSensor* designates a virtual device that consists of two "real" sensors: one that provides temperature, humidity and atmospheric pressure, and the other that gives information on brightness.

The entity relationship associated with the VALLPASS data model is shown in Fig. 1. Notice that this is a simplified version intended just to develop intuition regarding the relationships between the above entities as well as to show their key attributes.

V. Development of the NGSI-LD data model

During this section, the systematic process used to create the NGSI-LD data model for the VALLPASS system and, by extension, the proposed methodology for creating data models in the IoT domain will be presented. The data model can be created using various tools, but it will be defined according to the OpenAPI specification Version 3.1 namely through its Schema objects [27]. In this OpenAPI version, the paths specification object is not required. For this reason, it is not necessary to define paths and operations for the API - they are already defined in the NGSI-LD API. To make the specification valid in versions before 3.1, where the Paths object is mandatory, it is possible to define a dummy path.

A. Baseline data models

When creating a semantic data model, such as an NGSI-LD information model, is not necessary, and even advisable, to start from scratch since the goal is to use common, shared ontologies. For this reason, and whenever possible, the data models from the Smart Data Models Initiative will be used as baseline data models to represent the entities related to our scenario.

As already referred, the sensors in the current VALLPASS scenario represent virtual rather than real devices. In these use cases, where a virtual sensor system provides multiple types of measurements, [28] recommends modelling them as a system of sensors where each sensor is attached to a concentrator which, in the case of the VALLPASS system, will be the lighting pole. Following this recommendation, the *WeatherSensor* will be divided into two sensors:

a) **TemperatureHumidityPressureSensor**: It monitors the temperature, relative humidity and atmospheric pressure.
b) **BrightnessSensor**: Responsible to monitor the ambient brightness.

For each of the VALLPASS system entities identified, a search is made to look for the most appropriate Smart Data Model (if any). There is no problem if the Smart Data Model found does not faithfully represent our entity, since it can be amended later.

For the current application, the Smart Data Models found that are suitable for representing the VALLPASS data architecture will serve as the baseline data models for the following entities:

a) **Company**: The "Organisation" Smart Data Model [29], assigned to the Cross-Sector domain, will be used. It is a generic data model that is intended to represent various types of organisations and is therefore quite suitable for this entity.

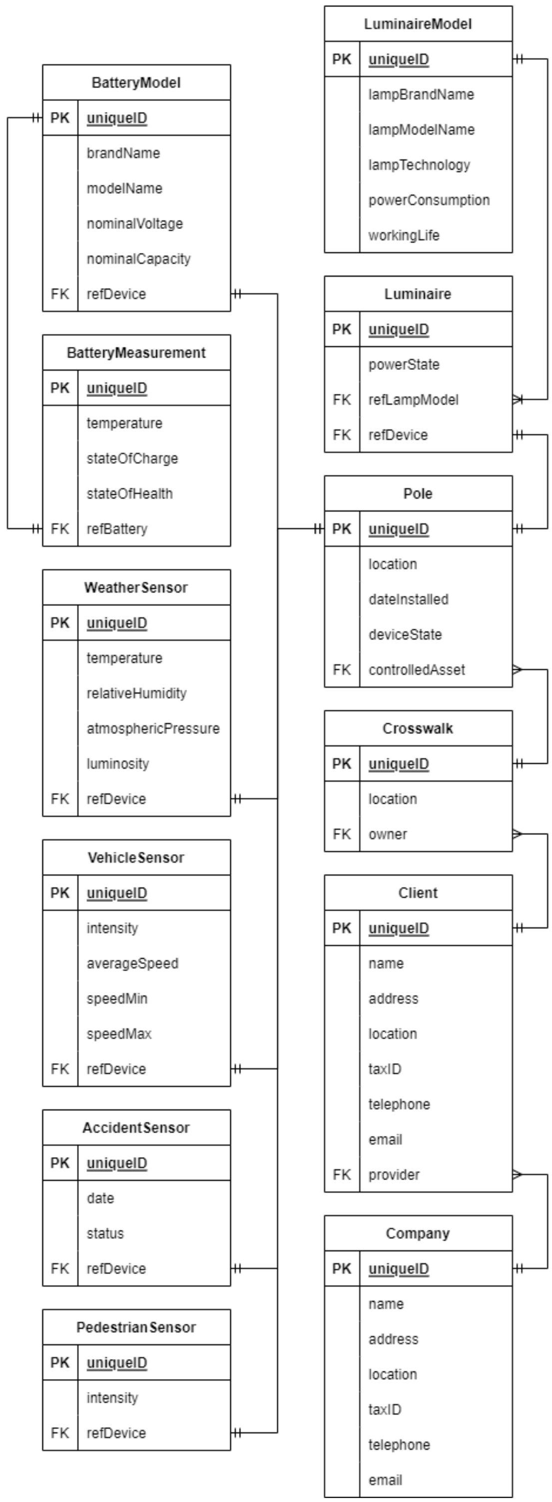

Fig. 1. Entity-relationship diagram associated with the data model of the VALLPASS system

b) **Client**: The target customers of the VALLPASS system will be organisations, so the same Smart Data Model of the Company entity will be used.

c) **Crosswalk**: For this entity, no specific Smart Data Model was found. The Smart Data Model "RoadSegment" from the Smart Cities domain will be used. It is intended to represent road segments, so although it is not specific, it has some degree of compatibility [30].

d) **Pole**: For this entity, no specific Smart Data Model was found, however, the Pole can be seen as an IoT device that besides being an actuator itself, will also contain other sensors and another actuator - the luminaire. For this reason, the Smart Data Model "Device" will be used, which is associated with the Smart Sensoring domain, and is intended to represent an "apparatus (hardware + software + firmware) intended to accomplish a particular task (sensing the environment, actuating, etc.)." [31] .

e) **TemperatureHumidityPressureSensor**: Like the "Pole" entity, this entity can also be seen as an IoT device (sensor, in this case) by itself, but we consider it rather as an entity that only describes certain measurements coming from the Pole. Under this view, the Smart Data Model "WeatherObserved", associated with the Smart Environment domain, fits perfectly, since it is intended to describe weather observations and defines all the necessary properties for this entity [32].

f) **BrightnessSensor**: Like the "TemperatureHumidityPressureSensor" entity, it is not considered an IoT device *per se*, but rather as an entity that describes certain measurements coming from the Pole. So, the same Smart Data Model as the "TemperatureHumidityPressureSensor " entity is chosen, which also defines all the necessary properties for this entity.

g) **VehicleSensor**: For this entity, no specific Smart Data Model was found. The Smart Data Model "ItemFlowObserved" will be used, which belongs to the Smart Cities domain and is intended to describe "an observation linked to the movement of an item at a certain location and over a given period" [33]. It can be seen that, despite being a generic Smart Data Model, it is compatible, since the sensor associated with this entity monitors the flow of an item - vehicles - over a certain period.

h) **PedestrianSensor** As with the sensor related to the "VehicleSensor" entity, the sensor associated with this entity also monitors the flow of an item - pedestrians - over a certain period, so the same Smart Data Model used for the "VehicleSensor" entity can be used.

i) **AccidentSensor**: For this entity, the Smart Data Model "RoadAccident" will be used. This model belongs to the Smart Cities domain and is intended to describe a road accident with its causes and consequences [34].

j) **LuminaireModel** The Smart Data Model "Streetlight-Model", associated with the Smart Cities domain, was chosen because it is intended to describe the technical characteristics of a lamppost including, its luminaire (which is what we are interested in) [35].

k) **Luminaire**: The Smart Data Model "Streetlight", which is associated with the Smart Cities domain and allows to represent the operational state of a luminaire, will be used [36].

l) **BatteryModel**: For this entity, the Smart Data Model "StorageBatteryDevice" was chosen, belonging to the Cross-Sector domain, which is intended, precisely, to describe the technical characteristics of a battery [37].

m) **BatteryMeasurement**: The Smart Data Model "StorageBatteryMeasurement", which belongs to the Cross-Sector domain, allows describing the current state of a battery (namely its remaining energy capacity) [38].

The schemas of the chosen Smart Data Models are added to our data model by reference, using the `$ref` keyword.

Before proceeding, it may be useful to clarify how relationships are defined in the current NGSI-LD semantic data model. Actually, it is not defined, at least not in the same way that a relational database (using foreign keys) is defined. The relationships will only be defined when entities are created or modified. Namely, through CRUD operations featured in the NGSI-LD API. What is possible to do is to define, in the entities that will constitute relationships, the properties that will promote (but will not "enforce") the relationships. For example, the mandatory attribute `controlledAsset` of the Pole entity will get the Uniform Resource Name (URN) of the Crosswalk it will control (or be a part of) - one-to-many relationship. In some way, this attribute behaves like a foreign key. However, it does not restrict the type of entity whose URN it will receive and can get the URN of any entity.

B. Amending the baseline data models

The Smart Data Models which have been chosen earlier (as baseline data models) are a good starting point. However, it may be necessary to tinker with the definition of the entities to create a digital twin that faithfully represents the associated physical "thing". The changes made to each entity in the VALLPASS system (if applicable), may involve adding new schemas and new properties.

Similar to the schemas of the Smart Data Models chosen earlier, new schemas are also added by reference and are combined with existing ones (composition) by using the "allOf" property of the Schema object.

Whenever new properties need to be added two options are at hand: one would be to add them to the entity "manually", using the Schema object's "properties" keyword. In this case, each of these properties would therefore be a Schema object and, since a semantic data model is being developed, it is required that, whenever possible, reference to existing definitions, such as those of similar properties already defined in a Smart Data Model, must be endorsed. Another option would be to add a new schema that defines the properties to add. The choice of one or the other option will essentially take into account two factors: the number of properties needed to add and the relationship between those properties. For example, if many entities must be added, which are all defined in the same schema, it might be a good idea to add that schema to the entity instead of adding the properties to the entity one by one, thus making our data model more compact.

Although not shown, the current data model specifies which properties are mandatory, that is, the ones that must be set when creating new entities. Notice that, although not required, this is a good practice. In this follow-up, the amendments made to the base data model are enumerated below:

a) **Company**: The telephone and e-mail of the Company are properties that are not present in the baseline schema chosen earlier. To add these properties, we chose to add a new schema to the entity - an OpenAPI definition of Schema.org type "ContactPoint" - which also defines other properties related to a contact point that may be useful to us in the future [39], [40].

b) **Client**: As with the Company entity, we also need to add the telephone and email properties to this entity, so for this problem, we applied the same solution as for the Company entity. This entity also represents the many ends of the relationship with the Company entity (one-to-many relationship), so it will have to have defined the property that will receive the URN of the Company thus signifying the relationship. Such a property is not defined in the baseline schema. For this reason, the property presented in Listing V-B was developed.
In the current definition, the "EntityIdentifierType" schema is used [41] which is part of a set of schemas that underpin all Smart Data Models and is intended to unambiguously identify any given NGSI entity. The `x-ngsi` field is a specification extension [27] and, as the name implies, serves to extend our OpenAPI specification. The fields associated with these properties must begin with `x-`. It is not mandatory to define these properties [42], but it is a good idea, especially if you plan to use the data model outside of NGSI-LD applications, namely in "generic" JSON-LD applications.

c) **AccidentSensor**: Unfortunately, the baseline schema chosen for this entity does not define a property that describes the relationship between this entity and the "pole" (one-to-one relationship). That is, the one that indicates which "Pole" is associated with an accident. For this reason, such a property was added. Regarding this property definition, referring to the "refDevice" property, present in the Smart Data Model "ItemsFlowObserved" (used in our "PedestrianSensor" and "VehicleSensor" entities) and intended to uniquely identify "the device or devices used to obtain the data expressed by this record" [33] which, although generic, perfectly fits the actual entity.

```
{
  refProvider:
    description: "Organization from which the system was
    acquired."
    anyOf:
      - description: Property. Identifier format of any
        NGSI entity
        type: string
        minLength: 1
        maxLength: 256
        pattern: ^[\w\-\.\{\}\$\+\*\[\]`|~^@!,:\\]+$
      - description: Property. Identifier format of any
        NGSI entity
        type: string
        format: uri
    x-ngsi:
      type: Relationship
```

```
}
```

C. NGSI-LD @context file

Now that the semantic data model has been created, there is only one step left to use it in the VALLPASS remote management platform, which is to generate the NGSI-LD @context file that will be used to ”expand” the terms, from shorthand strings, into concepts, specified by the URIs, and vice versa. That is, “compact” the URIs into terms. This way, applications that interact with the VALLPASS NGSI-LD broker will be able to programmatically “understand” the data stored in it. More specifically, the NGSI-LD @context file will specify the URIs for entity types, properties, and properties of properties (metadata).

VI. Results and discussion

The semantic data model developed is being used in the VALLPASS remote management platform successfully.

Only the entities defined in the data model that represent actuators - Pole and Luminaire - raised some difficulties in the integration, which were overcome, due to the way commands are processed by Orion-LD Context Broker (NGSI-LD broker) and IoT Agent (FIWARE GE used to interact with IoT devices). For example, a Pole can be turned on and off remotely and its operational state is represented by the attribute ”deviceState”, however, when defining commands we cannot simply declare this attribute as a command, firstly because it is not recommended (commands should be atomic) and secondly because it would always be ”split” into two meaningless attributes in the data model: ”deviceState_info” (actual result of the command) and ”deviceState_status” (state of the command). Thus, two commands - ”on” and ”off” - were defined and, to overcome this difficulty, a small backend application was developed which, using the subscription service provided by Orion-LD, updates the value of the ”deviceState” attribute according to the result of the execution of the commands - ”on_info” and ”off_info” attributes.

VII. Conclusions

IoT is strongly related to data heterogeneity. For this reason, achieving interoperability is one of the biggest, and most important, challenges in the IoT world today. Fortunately, when it comes to interoperability at the data level, LD and SWT are powerful enablers. Indeed, it is clear that semantic data models are an important step forward in IoT interoperability.

From the current research, it is possible to conclude that, at the present, a large effort is being done by different players in the data model scenario to provide the basis for a uniform semantic kernel. FIWARE is an example of an organisation that provides data models for a large number of real-world applications. Moreover, it also provides the tools, such as context brokers and IoT agents, that enable the implementation of those data models into a practical solution.

In describing the systematic process used in the development of the NGSI-LD semantic data model for the VALLPASS project, this paper has presented a useful methodology that can be adopted in the development of semantic data models to represent and describe digital twins, where common difficulties related to the selection of well-known data models and how to proceed whenever there are mismatches between the chosen models and the scenario we are modeling, arise.

The VALLPASS project deals with the construction of smart crosswalks and the developed solution must be able to be easily integrated into the smart cities ecosystem. The data model presented in this paper is, at the moment, living inside FIWARE’s Orion context broker where entities are populated with real-time data that comes from smart crosswalk prototypes.

Acknowledgment

This work was supported by NORTE - FEDER funds and by Horizon 2020, within the project “VALLPASS”, NORTE-01-0247-FEDER-113439. The authors are also grateful to the Foundation for Science and Technology (FCT, Portugal) for financial support through national funds FCT/MCTES (PIDDAC) to CeDRI (UIDB/05757/2020 and UIDP/05757/2020), SusTEC (LA/P/0007/2021) and INESC Coimbra (UIDB/00308/2020).

References

[1] “Smart Cities Market Size to Reach USD 1,427.84 Billion in 2030 | Emergen Research,” accessed on January 11, 2023. [Online]. Available: https://finance.yahoo.com/news/smart-cities-market-size-reach-163000941.html

[2] V. Albino, U. Berardi, and R. M. Dangelico, “Smart cities: Definitions, dimensions, performance, and initiatives,” *Journal of urban technology*, vol. 22, no. 1, pp. 3–21, 2015.

[3] R. E. Hall, B. Bowerman, J. Braverman, J. Taylor, H. Todosow, and U. Von Wimmersperg, “The vision of a smart city,” Brookhaven National Lab.(BNL), Upton, NY (United States), Tech. Rep., 2000.

[4] A. Abella, M. Ortiz-de Urbina-Criado, and C. De-Pablos-Heredero, “A model for the analysis of data-driven innovation and value generation in smart cities’ ecosystems,” *Cities*, vol. 64, pp. 47–53, 2017.

[5] S.-R. Oh and Y.-G. Kim, “Security requirements analysis for the iot,” in *2017 International Conference on Platform Technology and Service (PlatCon)*. IEEE, 2017, pp. 1–6.

[6] T. Berners-Lee and M. Fischetti, *Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor*. Harper San Francisco, 1999.

[7] R. Uceda-Sosa, B. Srivastava, and R. J. Schloss, “Building a highly consumable semantic model for smarter cities,” in *Proceedings of the AI for an Intelligent Planet*, 2011, pp. 1–8.

[8] B. Rocha, E. Cavalcante, T. Batista, and J. Silva, “A linked data-based semantic information model for smart cities,” in *2019 IX Brazilian Symposium on Computing Systems Engineering (SBESC)*. IEEE, 2019, pp. 1–8.

[9] N. Zhang, H. Chen, X. Chen, and J. Chen, “Semantic framework of internet of things for smart cities: Case studies,” *Sensors*, vol. 16, no. 9, p. 1501, 2016.

[10] J. Lee, S. Jeong, S. K. Yoo, and J. Song, “Ssf: Smart city semantics framework for reusability of semantic data,” in *2021 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2021, pp. 1625–1627.

[11] B. Villalón, Dalia M, S. Jiménez, Laura, M. de la Iglesia, M. E. Campos, and D. Fernández, Tatiana, “An iot architecture for smart cities based on the fiware platform,” *Revista de Ciencia y Tecnología*, no. 38, pp. 21–30, 2022.

[12] L.-G. Cretu, “Smart cities design using event-driven paradigm and semantic web,” *Informatica Economica*, vol. 16, no. 4, p. 57, 2012.

[13] P. Gonzalez-Gil, J. A. Martinez, and A. Skarmeta, “A prosumer-oriented, interoperable, modular and secure smart home energy management system architecture,” *Smart Cities*, vol. 5, no. 3, pp. 1054–1078, 2022.

[14] S. Jeong, S. Kim, and J. Kim, "City data hub: Implementation of standard-based smart city data platform for interoperability," *Sensors*, vol. 20, no. 23, p. 7000, 2020.

[15] J. A. López-Morales, J. A. Martínez, and A. F. Skarmeta, "Digital transformation of agriculture through the use of an interoperable platform," *Sensors*, vol. 20, no. 4, p. 1153, 2020.

[16] O. Hylli, V. Heikkilä, and K. Systä, "Collecting data to fiware," Tech. Rep., 2020, accessed on June 05, 2023. [Online]. Available: https://trepo.tuni.fi/bitstream/handle/10024/137915/Collecting_data_to_FIWARE.pdf?sequence=1

[17] O. Hylli, "Ngsi-ld," Tech. Rep., 2020, accessed on June 05, 2023. [Online]. Available: https://trepo.tuni.fi/bitstream/handle/10024/137920/NGSI_LD.pdf?sequence=1

[18] "About FIWARE – FIWARE," Oct. 2021, accessed on February 09, 2023. [Online]. Available: https://www.fiware.org/about-us/

[19] "FIWARE, the standard that the IoT needs," accessed on February 09, 2023. [Online]. Available: https://www.tmforum.org/press-and-news/fiware-standard-iot-needs/

[20] "Developers Catalogue – FIWARE," May 2018, accessed on February 09, 2023. [Online]. Available: https://www.fiware.org/catalogue/

[21] "FIWARE, Information Platform for Implementing Data Utilization Based City Management: NEC Technical Journal | NEC," accessed on February 09, 2023. [Online]. Available: https://www.nec.com/en/global/techrep/journal/g18/n01/180109.html

[22] "Smart Data Models – FIWARE," Mar. 2020, accessed on February 09, 2023. [Online]. Available: https://www.fiware.org/smart-data-models/

[23] "FIWARE – Cities Directory," Nov. 2021, accessed on February 09, 2023. [Online]. Available: https://www.fiware.org/about-us/smart-cities/cities-directory/

[24] "ETSI GS CIM 006 V1.1.1," ETSI, Tech. Rep., July 2019, accessed on June 05, 2023. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/CIM/001_099/006/01.01.01_60/gs_CIM006v010101p.pdf

[25] D. Bees, L. Frost, M. Bauer, M. Fisher, and W. Li, "Ngsi-ld api: for context information management," ETSI, Tech. Rep., January 2019, accessed on June 05, 2023. [Online]. Available: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp31_NGSI_API.pdf

[26] "EU: pedestrian traffic fatalities," accessed on February 21, 2023. [Online]. Available: https://www.statista.com/statistics/1197292/pedestrian-traffic-fatalities-in-the-eu/

[27] "OpenAPI Specification v3.1.0 | Introduction, Definitions, & More," accessed on March 29, 2023. [Online]. Available: https://spec.openapis.org/oas/v3.1.0

[28] G. Privat, "Guidelines for modelling with ngsi-ld," ETSI, Tech. Rep., March 2021, accessed on June 05, 2023. [Online]. Available: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp_42_NGSI_LD.pdf

[29] "Organization," accessed on March 29, 2023. [Online]. Available: https://swagger.lab.fiware.org/?url=https://smart-data-models.github.io/dataModel.Organization/Organization/swagger.yaml

[30] "Roadsegment," accessed on March 29, 2023. [Online]. Available: https://swagger.lab.fiware.org/?url=https://smart-data-models.github.io/dataModel.Transportation/RoadSegment/swagger.yaml

[31] "Device," accessed on March 29, 2023. [Online]. Available: https://swagger.lab.fiware.org/?url=https://smart-data-models.github.io/dataModel.Device/Device/swagger.yaml

[32] "Weatherobserved," accessed on March 27, 2023. [Online]. Available: https://swagger.lab.fiware.org/?url=https://smart-data-models.github.io/dataModel.Weather/WeatherObserved/swagger.yaml

[33] "Itemflowobserved," accessed on March 27, 2023. [Online]. Available: https://swagger.lab.fiware.org/?url=https://smart-data-models.github.io/dataModel.Transportation/ItemFlowObserved/swagger.yaml

[34] "Roadaccident," accessed on March 27, 2023. [Online]. Available: https://swagger.lab.fiware.org/?url=https://smart-data-models.github.io/dataModel.Transportation/RoadAccident/swagger.yaml

[35] "Streetlightmodel," accessed on March 29, 2023. [Online]. Available: https://swagger.lab.fiware.org/?url=https://smart-data-models.github.io/dataModel.Streetlighting/StreetlightModel/swagger.yaml

[36] "Streetlight," accessed on March 29, 2023. [Online]. Available: https://swagger.lab.fiware.org/?url=https://smart-data-models.github.io/dataModel.Streetlighting/Streetlight/swagger.yaml

[37] "Storagebatterydevice," accessed on March 27, 2023. [Online]. Available: https://swagger.lab.fiware.org/?url=https://smart-data-models.github.io/dataModel.Battery/StorageBatteryDevice/swagger.yaml

[38] "Storagebatterymeasurement," accessed on March 27, 2023. [Online]. Available: https://swagger.lab.fiware.org/?url=https://smart-data-models.github.io/dataModel.Battery/StorageBatteryMeasurement/swagger.yaml

[39] S. D. M. Initiative, "Schema.org," https://github.com/smart-data-models/data-models/blob/master/schema.org.yaml, 2022, accessed on March 29, 2023.

[40] "ContactPoint - Schema.org Type," accessed on March 26, 2023. [Online]. Available: https://schema.org/ContactPoint

[41] S. D. Models, "common-schema.json (source code) [github repository]," https://github.com/smart-data-models/data-models/blob/master/common-schema.json, 2020, accessed on March 27, 2023.

[42] F. Foundation, "Understanding @context - step-by-step for ngsi-ld," https://ngsi-ld-tutorials.readthedocs.io/en/latest/understanding-@context.html, 2023, accessed on March 29, 2023.