

# Normalizing Public Transport Accessibility Data in Real-Time

## A Systematic Transformation of Accessibility Data to RDF

Paloma Cáceres, Almudena Sierra-Alonso, Belén Vela, José María Cavero, Miguel A. Garrido, Carlos E. Cuesta

Escuela Técnica Superior de Ingeniería Informática

Universidad Rey Juan Carlos, URJC

Móstoles (Madrid), Spain

e-mail: {paloma.caceres, almudena.sierra,belen.vela,josemaria.cavero,miguel.garrido,carlos.cuesta}@urjc.es

**Abstract**—Smart mobility is a major component of smart city initiatives and is focused on improving citizens' mobility. One of the policies employed to increase mobility is that of promoting the use of public transport. But if this use is to be made extensive, it is necessary to make transport much more inclusive and accessible. Providing applications that guide citizens as they travel around the city on public transport is not something new, but it often occurs that these applications do not have information regarding accessibility that will allow users to establish routes according to their needs. One of the origins of this situation is that these public transport accessibility data are not easy to find or use in order to develop applications that address this issue. Moreover, in many cases the information is not updated in real time. In this work, we present a smart data framework that is capable of managing accessibility data related to public transport and gathering information about incidents in the transport network in real time. These data are annotated semantically and published as open data using automatic transformations defined with the RDF modelling language, thus making data immediately available for the calculation of new accessible routes.

**Keywords** - Public transport; accessibility; linked open datasets; normalization; RML.

### I. INTRODUCTION

As stated in our previous work, presented at SMART ACCESSIBILITY conference, urban mobility is one of the greatest challenges for modern cities [1]. It is a complex problem in which very different aspects, such as air and noise pollution, traffic congestion or medical and economic costs resulting from accidents, have to be considered [2]. Smart mobility as part of smart cities would appear to be one of the main approaches by which to solve this problem [2].

The source of all problems in urban mobility is the flow of people and vehicles attempting to use the same spaces at the same time. One improvement that could be made is that of stressing the importance of using the public transport, which would, however, be much more efficient if people were able to avoid colliding with other people when traversing the cities. This could be done if they were able to use their own smart devices to find specific routes within the urban space.

A software system that provides maps and calculates routes in public transport networks is no longer a novelty. However, most of the current routing systems still do not

consider accessibility. Even the most popular among them, Google Maps, does not provide accessible routes. However, institutions such as the European Union (EU) aim to provide initiatives that will improve the information concerning the accessibility of the urban transport network: “the special needs of user groups in different cities need to be taken into account” [3] and to provide those users with real-time information in order to improve their mobility and well-being [2].

But, why do route calculation applications not provide accessibility information? The main reason is that information of this type is scattered in different sources and formats. That is, there are no easily available data that can be used to provide accessible routes to users with special needs.

The intention of our previous research has been to use semantic information to fill this niche [1][4][5][6][7]. In [7], we worked with the Public Bus network of the city of Madrid. In our other work, we have gathered the data concerning the metro infrastructure in the city of Madrid (Spain) and the accessibility features of that infrastructure, which we have then used to develop an open data semantic repository [4][5]. These data are, therefore, now available for the applications that provide accessible routes.

Building upon this infrastructure, our platform is able to evolve beyond its original constraints, and thus consider higher accessibility levels. Our initial architecture had to deal with mostly non-volatile data, i.e., data with a small rate of change. When our system has to compute a route, it usually uses very stable data (the structure of the network, predefined timetables), and their alterations usually have a certain durability (planned work, changes in the infrastructure or the rolling stock). However, other sources of change are much less predictable: collapses, floods, accidents, or even the traffic flow. These are usually easy to perceive once they have occurred - but many citizens are unaware of these *incidents* until they encounter them.

Our system must, therefore, include incident management to be complete. This would allow our smart data to include live information and to provide an even more intelligent response in real-time, which would take the situation at any given moment into account. This is particularly important when considering accessibility and the relevant groups of interest. Owing to the need for real-time data, in [1] we, therefore, collected data provided by Metro Madrid users and related to incidents regarding accessibility elements, such as lifts, stairs, etc. This data

collection is carried out using a *crowdsourcing* strategy: it is the users themselves who provide information about incidents at the moment at which they encounter them. This information is sent using mobile devices is stored in our routing platform, and has probably been attained from many coincidental sources. These new data are collapsed and integrated with pre-existing accessibility information – signifying that semantic annotations are also used for their processing when an accessible route is calculated. In summary, their combination can be conceived as *real-time accessibility data*. This approach effectively provides an advantage over other approaches, such as those discussed in the Related Work Section.

This integration of data from *crowdsourcing* into the aforementioned semantic repository (data concerning infrastructures and accessibility features) implies several challenges: their specific meanings and vocabulary; the incident lifecycle management, or the handling of simultaneous sources, and even their scale. Our evolved architecture must deal with all of this, and this paper presents our solution.

One of these challenges, i.e., the handling of simultaneous sources and different formats, can be addressed by means of a mapping language: mapping rules are defined to convert the collected data (in this case, in a JSON file) into an RDF model. The use of the mapping language facilitates and speeds up the task of transforming and linking data.

With regard to the date scale, the use of a mapping language also facilitates this task, especially if the data set is very large, as an automatic transformation is performed by applying the rules defined.

The paper is structured as follows: Section II presents some related works, and Section III describes our CoMobility and Access@City projects, which constitute the context of this work. Section IV presents the smart data framework developed, describing our proposed client-server scaling architecture and the main semantic technologies used to model the semantic repository. Section V describes the semantic vocabulary employed to annotate events and the set of mapping rules that converts the incident data into RDF triples using that vocabulary. Section VI describes how we validated this set of rules by means of a case study with the MM4A4 App [1], and our conclusions and future works are presented in Section VII.

## II. RELATED WORK

There already are several software applications that provide information about specific aspects of the public transport domain. In some cases, they include accessible wayfinding information or/and accessibility features, or elements for people with special needs or disabilities. Other software provides data about public transport using a crowdsourcing approach. In this section, we discuss some of the most representative of them.

Of those related to accessibility, we should mention Landmark Ontology for Hiking [8], which is focused on elderly people and helps them to walk less by using wheelchairs. It formally represents landmarks for hiking.

Wheelmate [9] and/or Wheel Map [10], meanwhile, provide some information about accessible places for people in wheelchairs, and Access Map provides accessible routes to people with mobility needs [11]. All of these consider only mobility-related disabilities. Furthermore, “Ciudades Patrimonio de la Humanidad” has a web application that provides accessible routes to people with special mobility needs, including blind or hearing-impaired people [12]. However, none of these applications are customizable.

The EU is making a great effort to improve public transport [3]. Projects like ACCESS 2 ALL [13] or Mediate [14] have been set up with the objective of analysing how to answer all citizens’ accessibility needs. The goal of ACCESS 2 ALL is to define mobility schemes and guidelines in order to ensure the accessibility of public transport to all citizens, such as the elderly and the disabled. It, therefore, began by defining user needs. Moreover, it proposes customised services for route guidance coupled with localisation methods. Furthermore, the Mediate Project has identified a set of indicators with which to describe accessibility, has developed a tool to measure the accessibility of urban transport and has published a Good Practice Guide for accessibility. These projects seek to establish a theoretical framework covering all aspects of citizens’ mobility, but do not offer solutions in the form of user applications.

Of those that employ a crowdsourcing approach, some applications use crowdsourcing to improve the experience of using public transport and to provide real-time information about the status of public transport. For example, Tiramisu Transit provides data such as how full a bus is or whether any wheelchair space is left [15]. Moovit is able to plan routes and indicate when to get off, or the status of the service [16]. Swiftly works with transport agencies rather than with the general public [17]. It provides more accurate vehicle arrival data for these transport agencies, thus enabling them to provide their users with better information. The OneBusAway project consists of a set of tools to improve the user experience on public transport by ensuring that buses and other transit systems arrive on time, decreasing waiting times, increasing feelings of safety, or even increasing transit trips per week [18]. OneBusAway provides several feedback mechanisms that allow users to make comments about these tools.

Other initiatives that use crowdsourcing are the BUSUP project [19] and the CIVITAS initiative [20]. The former allows users to book crowdsourced buses on demand, while the intention of the latter is to achieve cleaner, better transport in Europe. A CIVITAS subproject is dedicated to mobility strategies for vulnerable groups.

OpenTripPlanner (OTP) is another project with which to provide services for passenger information and transport network analysis [21]. It computes routes by combining transit, pedestrian, bicycle and car segments traversing networks built from OpenStreetMap [22] and GTFS [23] data. OTP also takes (transport) accessibility into account.

As will be noted, the number of systems whose purpose is to improve the user experience in the public transport are increasing. However, most of them (with some exceptions, such as Tiramisu Transit or CIVITAS) do not take the information about accessibility elements into account, despite the fact that these elements are necessary for users with special needs in order to also improve their experience on public transport. Even the aforementioned systems [15][19], although they provide accessibility features, do not allow the users to inform about the actual state of these elements: whether a lift is operative, if work is taking place that will prevent or hinder access for, i.e., blind people, etc.

To the best of our knowledge, there are currently no software applications that analyse the status of the public transport network in order to estimate the availability of accessibility features, while also using crowdsourcing to update their data.

With regard to the transformation of the collected data (in a certain format) into RDF for publication in a semantic repository, several works propose the automatic execution of mappings. Most existing solutions are based on a specific format [24]: for example, R2RML converts data from relational databases [25]. There are several approaches with which to convert data from csv and spreadsheets into the RDF data model, which are based on the idea that each row describes a resource and each column represents a property [24]. Some examples are the XLWrap's mapping language [26] or Tarql [27], both of which convert CSV into RDF using SPARQL. There are a variety of solutions with which to map from XML to RDF that rely on XSLT, XPATH or XQUERY [24]. Few tools support mappings from different source formats to RDF. Furthermore, those tools cannot integrate data from different sources. OpenRefine [28] and Virtuoso Sponger [29] are examples of this type of tools.

### III. THE CONTEXT: COMOBILITY AND ACCESS@CITY PROJECTS

This work is being developed in the context of two research projects. The first, called CoMobility [30], defines a multimodal architecture based on linked open data for sustainable mobility. Its main goals are to improve citizens' mobility and to optimize their trips by combining public transport and car sharing. The second, called Access@City, is a coordinated project that defines a technological framework in which to process, manage and use open data concerning public transport with the goal of promoting its accessibility [31]. One of its subprojects is Multiply@City, which focuses on processing and harmonizing public transport accessibility data in a semantic manner by means of an ontology, taking into account that data are provided by different sources and have different formats [32]. The accessibility data are obtained from open data by means of Web scraping and they can also be updated here via crowdsourcing techniques. Fig. 1 provides a general depiction of the latter project.

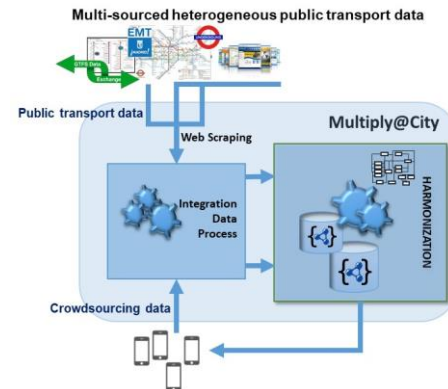


Figure 1. Multiply@City project architecture.

The Regional Consortium for Public Transport in Madrid (CRTM) [33], the Madrid public bus company (EMT Madrid) [34], and the Spanish National Society for the Blind (ONCE) [35] have all expressed their interest in the results of our CoMobility and Access@City projects.

### IV. BACKGROUND

In this section, we introduce our smart data framework, which is based on the Resource Description Framework (RDF) and the RDF Mapping language (RML) semantic technologies. The framework describes the architecture that supports our work. The semantic technologies will be used in order to define our current proposal.

#### A. The Smart Data Framework

Our Smart Data Framework (SDF) was defined in full in Cáceres et al. [1]. In this subsection, we introduce the most relevant aspects.

The SDF is supported by a client-server architecture (see Fig. 3), which allows communication between different Android apps and a server. The app related to this work, denominated as MMA4A, provides two different alternatives to users: (a) requesting an accessible route by considering their specific needs as regards getting around using the Metro, and (b) collecting events or incidents related to unavailable accessibility features (lifts, escalator, stairs, etc).

In order to compute the route (a), it is first necessary to obtain the infrastructure data concerning the public transport network, and to request the current state of that infrastructure – provided by the server. It is then necessary to establish a correspondence between the accessibility elements (their current state) and the user's needs. Transmodel [36] and IFOPT [37] standards have been analysed in order to identify the specific accessibility characteristics that each special need requires. TABLE I summarizes the accessibility elements and their correspondence with the users' needs considered in this work.

Both models (IFOPT and Transmodel) have been used as a basis on which to develop a specific *vocabulary* as an RDF Schema [38], which comprises the information required to semantically annotate the events regarding

accessibility that can be produced, which are obtained via crowdsourcing. Fig. 2 shows the RDF Schema employed to describe this terminology and its relationships. This schema includes terms with which to describe the incidents.

In order to annotate the events regarding the accessibility elements with the concepts defined in the RDF Schema shown in Fig. 2, it was necessary to define a vocabulary denominated as MAnTo. TABLE II provides a summary of the MAnTo vocabulary.

The use of this vocabulary makes it possible to register an incident, e.g., a lift does not work (*hasLift FALSE*) at a specific *StopPlace* (i.e., station) on a specific *ofLine* (transport line). This event also has an associated opening date (*openDate*). When the incident is solved, we can close it by indicating the final date (*closeDate*) and then assigning a TRUE value to both *hasLift* (the original feature) and *closedEvent*.

The event data concerning accessibility features, which is processed from many sources (*crowd*), is semantically annotated on the basis of this vocabulary and is then stored in the semantic repository, in the events collection.

Once the events have been collected by the users (b), it is also necessary to send these incidents to the server (as a JSON file), in order to update the current accessibility features in the network. It is, therefore, necessary to develop a specific server architecture that stores the infrastructure data and its accessibility status and is updated using the crowdsourcing approach.

TABLE I RELATIONSHIP BETWEEN FEATURES AND USERS' NEEDS

Accessibility features of public transport (based on IFOPT)	Users' accessibility needs (based on IFOPT)			
	Auditory and visual	Mobility	Phobia to lifts	Phobia to escalators
Lift	✓	✓	✗	✓
Escalator	✓	✗	✓	✗
Ramp	✓	✓	✓	✓
Stairs	✓	✗	✓	✓
Travelator	-	-	✓	✗

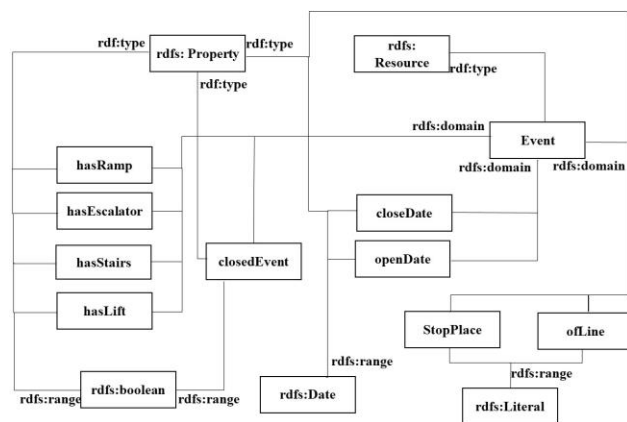


Figure 2. Events vocabulary in RDF Schema.

In the server, which is denominated as Coruscant, we have implemented Spring Boot [39] as the application server (AppServer), an Apache Kafka [40] server and an Apache Jena Semantic Repository (SR) [41]. The server should simultaneously 'listen to' the (potentially many) notifications of incidents or events concerning the accessibility features that are not available in the public transport network at that moment. It is for this reason that we have implemented the Apache Kafka server as a Queuing Manager (QM). The QM gathers the different events that users have notified by employing their smart devices, i.e., the QM gathers the JSON files generated by the apps. The JSON files are transformed into RDF data by the AppServer and the RDF data are then stored in the SR.

The SR maintains two separate data collections: one of them provides the (mostly static) data concerning stations, lines and stops, denominated as the *infrastructure collection*, and the other provides the (dynamic) data concerning the state of the network as regards its accessibility features, denominated as the *events collection*.

TABLE II EVENTS MANTO VOCABULARY

MAnTo term	Description
<code>mao:event</code>	Opens an event
<code>mao:closedEvent</code>	Closes the event
<code>mao:openDate</code>	Registers the date of the incident
<code>mao:closeDate</code>	Registers the closing date of the incident
<code>mao:hasLift</code>	Registers whether the lift works
<code>mao:hasStairs</code>	Registers whether the stairs are accessible
<code>mao:hasEscalator</code>	Registers whether the escalator works
<code>mao:hasRamp</code>	Registers whether the stairs have a ramp
<code>mao:hasTravelator</code>	Registers whether the travelator works
<code>mao:StopPlace</code>	Assigns the incident to a station
<code>mao:ofLine</code>	Assigns the incident to a transport line

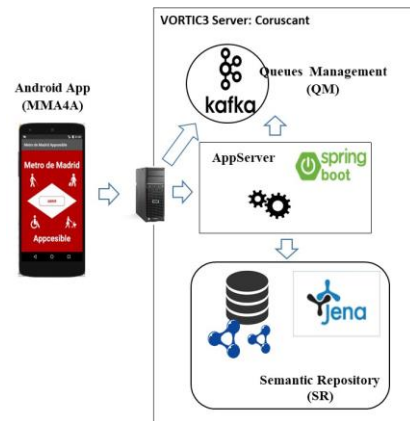


Figure 3. Architecture framework.

**B. The Resource Description Framework (RDF)**

The Resource Description Framework (RDF) provides a very simple model for data interchange on the Web [42]. In RDF, the data that are distributed by the Web (also called resources) are described as a set of triples. A triple is formed of three components: a subject, a predicate and an object. The subject is a resource that is related to an object by means a predicate. The object can be another resource, which has been defined previously, or a literal. The predicate can describe a relationship between the subject and object or a property of the subject [43].

In order for the data of which the triples are formed to be referenced, RDF identifies each component of the triple with a URI (Uniform Resource Identifier) [44]. For example, a car with a set of features (make: BMW, license plate: "5644BHZ" and colour: red) is modelled in RDF as Fig. 4 shows.

The URI "http://www.vortic3.com/5644BHZ" would identify a car with the license plate number "5644BHZ" that has been manufactured by BMW and is red.

The set of triples can be represented graphically as a directed labelled graph (RDF Graph) on which subjects and objects are nodes and predicates and are represented with edges. The triples of the previous example form the graph shown in Fig. 5. The edge labels preserve the semantics of the relationship, signifying that the use of standard vocabularies that can always be interpreted in the same way by those who use the data is highly recommended. Sometimes, depending on the data domain, no vocabulary has been defined and it is necessary to formalize a new one. In Fig. 5, the label "http://purl.org/vso/ns#color" would identify where the semantics of that predicate is formalized, thus allowing all the users of that data to interpret the predicate in the same way. This condition facilitates the integration of several datasets by merging several RDF graphs. The integration takes place at the most basic level of triples, that is, connecting two datasets can be as easy as establishing a predicate between resources from different datasets or determining a common resource between those datasets.

Number	Subject	Predicate	Object
1	<a href="http://www.vortic3.com/5644BHZ">http://www.vortic3.com/5644BHZ</a>	<a href="http://purl.org/goodrelations/v1#hasManufacturer">http://purl.org/goodrelations/v1#hasManufacturer</a>	"BMW"
2	<a href="http://www.vortic3.com/5644BHZ">http://www.vortic3.com/5644BHZ</a>	<a href="http://purl.org/vso/ns#color">http://purl.org/vso/ns#color</a>	"Red"

Figure 4. RDF Triples.

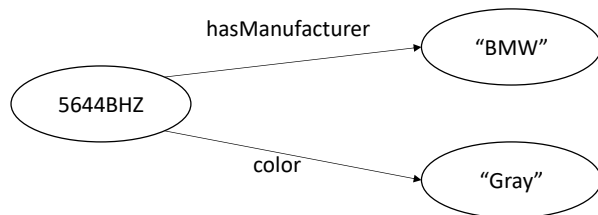


Figure 5. RDF Graph.

The representation as a graph allows a better understanding of the relationships among data. This representation must be transformed such that computers can handle the data. One of the most frequently used notations is RDF-XML [45]. Fig. 6 shows the serialization in RDF-XML notation for the graph in Fig. 5.

Data can be extracted from this RDF file by means of SPARQL queries [46] or by parsing the XML file.

In summary, by using RDF to model Web data it is possible: a) to integrate data from different sources; b) to publish data in order to reuse and share; c) to modify or increase data without affecting the customers who use them; and d) for computers to interpret the information modelled with RDF using a standard model that preserves its meaning.

**C. The RDF Mapping language (RML)**

RML (the RDF Mapping language) is a general language that permits the definition of rules with which to map heterogeneous data sources onto RDF graphs [24].

The transformation from each data source into RDF is usually carried out independently, after which data from different sources are interlinked. This causes the previous definitions to be disregarded or the same concepts to appear in different datasets and even with different names. In order to avoid this problem, the data must be incorporated and interlinked with the published data at the same time as they are converted to RDF triples. RML solves this problem by converting the data to RDF at the same time as it interlinks them with existing datasets.

RML is an extension of R2RML. R2RML permits express rules to transform data in a relational database into an RDF graph [25], while RML makes it possible to define rules that map relational and semi-structured data (e.g., XML, JSON...) onto RDF graphs. Fig. 7 shows the RML structure.

In RML, the data source is defined with a Triples Map rule that consists of a Logical Source, a Subject Map and zero or more Predicate-Object Maps [47]:

```

<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:vso="http://purl.org/vso/ns#"
  xmlns:gr="http://purl.org/goodrelations/v1#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description
rdf:about="http://www.vortic3.com/5644BHZ">
    <gr:hasManufacturer>BMW</gr:hasManufacturer>
    <vso:color>Red</vso:color>
  </rdf:Description>
</rdf:RDF>
  
```

Figure 6. RDF Graph in RDF-XML notation.



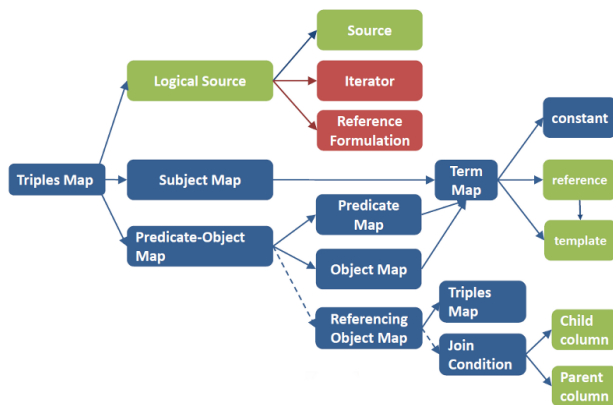


Figure 7. RML structure (from <http://rml.io/RMLmappingLanguage.html>).

- The Logical Source establishes the data source and where it is (Source). The source can be a relational database, JSON or XML documents, etc. Data from that source are accessed by defining an iterator pattern (Iterator) on the input data. The iterator pattern describes how the input data should be retrieved. For example, if the data source is an XML document, an XPath expression can be used to define how to extract the data. If the input data are contained in a JSON document, a JSONPath expression can define the iterator on data, and an SQL query can define the iterator on a relational database. The type of expression that defines the iterator is specified in the Reference Formulation. RML has the following predefined Reference Formulations: XPath, JSONPath, SQL2008, CSV, CSS3 and JSONPath (see <http://rml.io/RMLmappingLanguage.html>).
- The Subject Map defines the rule required to generate the URIs that identify the resources to be mapped and is used as the subject of all the triples that this Triples Map rule generates.
- Finally, the Predicate-Object Map rules define how triples are formed. These rules have two elements: The Predicate Map, which specifies how the predicate is defined, and the Object Map, which determines how the object is generated.

The Subject Map, the Predicate Map and the Object Map are Term Maps, that is, rules that produce an IRI, a blank node or a literal. A Term Map can be mapped onto a constant value, a data value of a referenced data fragment in a given Logical Source, or a template that is a valid string template, which can contain the referenced data fragments of a given Logical Source [47].

In order to transform a complete data source, we can define as many Triple Map rules as we need.

RML solves the above mentioned problem of generating the triples and then linking the data. The solution is attained by specifying resources as URIs when the Subject Map and the Object Map are defined. When

data are incorporated into a dataset, the same schemes are, therefore, used to define the resources that are linked at the same time as the triples are created [24].

## V. NORMALIZING SMART ACCESSIBILITY DATA

As mentioned in the Background Section, the MMA4A app stores the users' notifications regarding the state of accessibility features of the Metro in real-time as JSON files and then sends them to the Coruscant server. Once in the server, the QM gathers the different JSON files generated by different users, after which the AppServer transforms the JSON files into RDF data and then stores these data in the SR. At this point, we wish to underline that this transformation from JSON files to RDF data was carried out by means of a custom-made Java code specifically developed for that purpose.

This work focuses on the transformation of the JSON files, which are the events or incidents generated by the MMA4A users in real-time, into RDF data using RML. As mentioned previously, RML is a generic mapping language that can express rules that map data with heterogeneous structures and serializations onto RDF graphs. The "RML in details" web page ([http://rml.io/RML\\_details.html](http://rml.io/RML_details.html)) states that most current methods support only mappings from a certain source format, employing source-centric approaches for each of the formats they support. It also states that RML supports the definition of the mapping rules independently of the references to the input data. In this respect, the reusable mapping definition RML allows the same mapping rules to be reused across different sources as long as they capture the same context only by adjusting the reference to the input source that contains the relevant information.

We, therefore, now propose to normalize the mapping rules from JSON files into RDF data by means of RML, which benefits the software maintenance of our smart accessibility data and thus avoids the need to use the custom-made Java code.

The RML mapping rules that automatically transform data in JSON format into RDF are presented in the following subsections.

### A. Defining RML mappings for smart accessibility data

In this work, we propose the normalization of mappings of the smart accessibility data so as to transform them from JSON format into RDF data by means of RML. RML defines the data based on one or more *Triple Maps*, which specify how the triples will be generated from the source.

Following the structure presented in Fig. 7, we first have to indicate what the *logical source* is. In our case, the source is the JSON file ("events.json") generated by MMA4A, the *iterator* is "\$" and the *reference formulation* is `ql:JSONPath`:

```
rml:logicalSource [
  rml:source "events.json";
  rml:referenceFormulation ql:JSONPath;
  rml:iterator "$"
];
```

We then have to indicate which the *Subject Map* is. In our case, the template is a string, which concatenates a base URL and the JSON branch (“\$.station”), and the class indicates an event (mao:event):

```
rr:subjectMap [
  rr:template
    "http://manto.vortic3.com#{$.station}";
  rr:class mao:event
];
```

Finally, we have to indicate which the *Predicate-Object Maps* are. There are four mandatory predicate-object maps:

(i) The date on which the incident opened, with the open date as the *Predicate* and the date value as the *Object Map*:

```
rr:predicateObjectMap [
  rr:predicate mao:openDate;
  rr:objectMap [
    rml:reference "openDate"
  ]
];
```

(ii) The line on which the incident has occurred, with ofLine as the *Predicate* and the line value as the *Object Map*:

```
rr:predicateObjectMap [
  rr:predicate mao:ofLine;
  rr:objectMap [
    rml:reference "line"
  ]
];
```

(iii) The station at which the incident has taken place, with sch:name as the *Predicate* (from the sch vocabulary) and the station value as the *Object Map*:

```
rr:predicateObjectMap [
  rr:predicate sch:name;
  rr:objectMap [
    rml:reference "$.station"
  ]
];
```

(iv) One or more accessibility elements, indicating whether or not they work: for example, hasLift as the *Predicate* and hasLift value as the *Object Map*:

```
rr:predicateObjectMap [
  rr:predicate mao:hasLift;
  rr:objectMap [
    rml:reference "$.hasLift"
  ]
];
```

**B. Developing RML mappings**

The designated software presented in this work, is related to the AppServer (which is located in the Coruscant server and is supported by Spring Boot technology), as shown in Fig. 1.

As mentioned previously, we have already developed a custom-made Java code, which transforms the JSON java file generated by the MMA4A app into RDF format. We have now implemented a set of mappings by means of

RML in order to transform the JSON file into RDF. The top-level design is shown in Fig. 8.

Before implementing the final solution, we have designed an AppServer package, which is structured in different classes and supports the full functionality of the AppServer. Fig. 9 shows this design by means of UML.

The AppServer software has three actors: the MMA4A Android app, which serves the JSON events file; the QM, which gathers the different real-time event files; and the Semantic Repository, which stores the RDF data generated. Moreover, we consider that the AppServer is a UML package, which is composed of a set of UML classes and their relationships: an MMA4A interface, which receives each JSON events file from the App and serves it to the ServerManager class; a QM interface, which serves the JSON event file from the ServerManager class to the QM actor in order to be consumed by it, and later receives the corresponding event file to be consumed, after which the ServerManager orders to transform the JSON events file into RDF data on the SR interface. In particular, the *addEventsintoGraph* method of the *SRInterface* class receives the JSON file as a parameter and then generates the events RDF graph by means of the RML mappings.

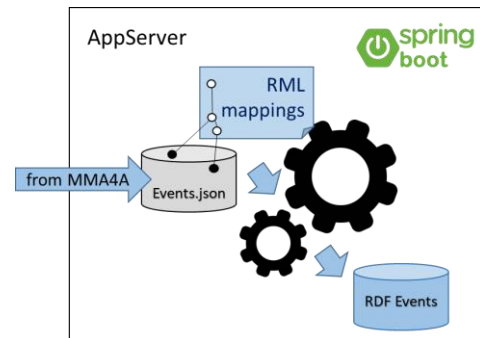


Figure 8. Top-level software design.

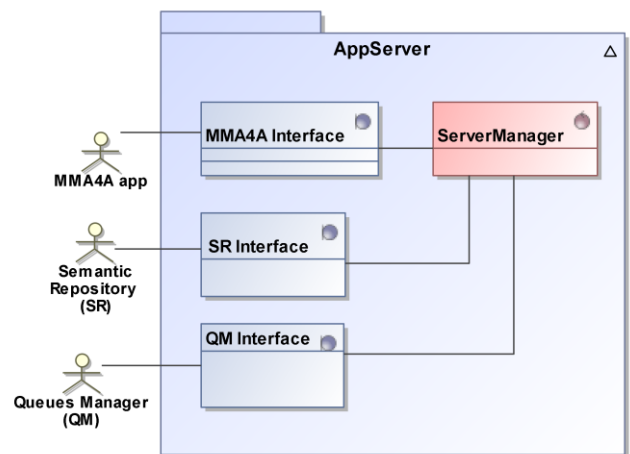


Figure 9. The AppServer package.

TABLE III shows an example of the RML mappings code with which convert the JSON event file into an RDF graph.

## VI. VALIDATION

In this section, we validate our proposal for the normalisation of smart accessibility public transport data by means of RML.

This validation is performed using the MM4A4 App [1]. This App is aimed at users of the Metro of Madrid who can warn when they find an incident in the accessibility elements of the network. By using this App, we have defined the following process for this purpose:

- First step. A user X, who has a specific user need, employs MMA4A to request a route. This route is composed of stations at which each of the accessibility features works correctly.
- Second step. A user Y notifies an event by means of MMA4A, which indicates which accessibility feature does not work at a specific station.
- Third step. The App sends the JSON event file to the Coruscant server, and the SR Interface class then generates the RDF file by means of RML mappings. This RDF file can be downloaded from <http://coruscant.my.to:8080/download/events.xml>.
- Fourth step (\*). We now validate the current RDF file by means of the RDF validator (<https://www.w3.org/RDF/Validator/>) in order to verify its correctness.
- Fifth step (\*\*). User X again requests the same route as in the first step. We should stress that this route is composed of stations at which some accessibility features do not work, including one related to the previously notified event. We verify whether or not the user obtains an accessible route.

This process includes a double validation. The first (\*) is the checking of the correctness by means of the RDF validator, as mentioned in the fourth step. The second (\*\*) concerns the correctness of the route calculated by MM4A4, as mentioned in the fifth step.

Details of this process for a specific example are shown as follows.

First step: A user (User X) requests a route starting at the Vodafone Sol station and finishing at Plaza de Castilla station, considering a phobia as regards escalators (the user interface is shown in Fig. 10a). In this case, a route exists and requires a transfer at Tribunal station. The complete route is: from Vodafone Sol station, take Line 1 (cyan blue) of the Metro to Tribunal station, then transfer to Line 10 (dark blue) and go to Plaza de Castilla. The details of this route are shown in Fig.10b.

TABLE III SPECIFIC RML MAPPINGS FROM OUR JSON DATA.

```
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix rml: <http://semweb.mmlab.be/ns/rml#>.
@prefix ql: <http://semweb.mmlab.be/ns/ql#>.
@prefix mao: <http://manto.vortic3.com#>.
```

```
@prefix sch: <http://schema.org/>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.

<#JSONEvents>
  rml:logicalSource [
    rml:source "event.json";
    rml:referenceFormulation ql:JSONPath;
    rml:iterator "$"
  ];

  rr:subjectMap [
    rr:template
      "http://manto.vortic3.com#{$.station}";
    rr:class mao:event
  ];

  rr:predicateObjectMap [
    rr:predicate mao:openDate;
    rr:objectMap [
      rml:reference "$.openDate"
    ]
  ];

  rr:predicateObjectMap [
    rr:predicate mao:closedEvent;
    rr:objectMap [
      rml:reference "$.closedEvent"
    ]
  ];

  rr:predicateObjectMap [
    rr:predicate mao:closeDate;
    rr:objectMap [
      rml:reference "$.closeDate"
    ]
  ];

  rr:predicateObjectMap [
    rr:predicate mao:ofLine;
    rr:objectMap [
      rml:reference "$.line"
    ]
  ];

  rr:predicateObjectMap [
    rr:predicate sch:name;
    rr:objectMap [
      rml:reference "$.station"
    ]
  ];

  rr:predicateObjectMap [
    rr:predicate mao:hasLift;
    rr:objectMap [
      rml:reference "$.hasLift"
    ]
  ];

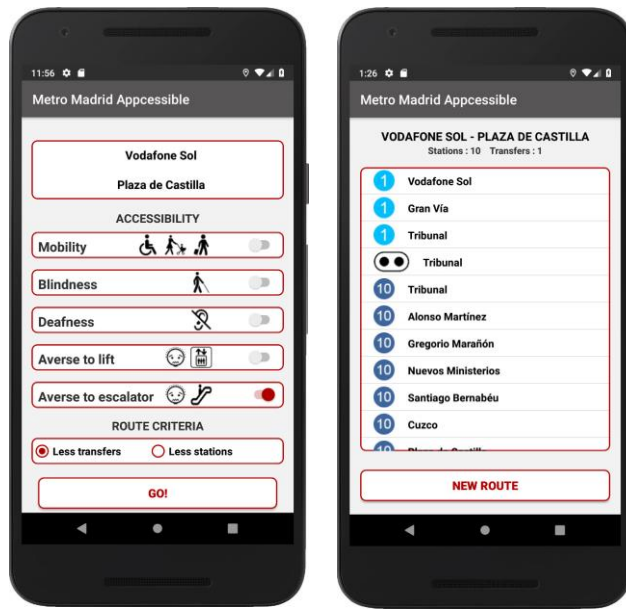
  rr:predicateObjectMap [
    rr:predicate mao:hasEscalator;
    rr:objectMap [
      rml:reference "$.hasEscalator"
    ]
  ];

  rr:predicateObjectMap [
    rr:predicate mao:hasStairs;
    rr:objectMap [
      rml:reference "$.hasStairs"
    ]
  ];

  rr:predicateObjectMap [
    rr:predicate mao:hasTravelator;
    rr:objectMap [
      rml:reference "$.hasTravelator"
    ]
  ];

  rr:predicateObjectMap [
    rr:predicate mao:hasRamp;
    rr:objectMap [
      rml:reference "$.hasRamp"
    ]
  ];
].
```





(a) Request a route

(b) The calculated route

Figure 10. User interface of MM4A4.

Second step: A user (User Y) notifies two events (the lift and stair accessibility features do not work) at Plaza de Castilla station.

Third step: The JSON event file sent to the Coruscant server and the RDF file is generated by means of RML mappings. TABLE IV shows the partial code (the stair accessibility feature does not work): the first row shows the JSON file of this event and the second row shows the corresponding RDF graph. We then download the RDF event file and, in the next step, we verify this RDF code.

Fourth step: The validator successfully checks the RDF file. Fig. 11 shows a partial screen capture.

Fifth step: User X again requests the same route as in the first step (from Vodafone Sol to Plaza de Castilla station and considering a phobia as regards escalators); we should highlight that the previous route generated for user X is now composed of stations at which some accessibility features do not work, which are those related to the event previously notified by user Y. We must, therefore, verify whether or not user X can obtain an accessible route.

It is important to stress that our algorithm provides a route only after examining whether the start (in this case, Vodafone Sol station), destination (Plaza de Castilla station) and any transfer stations (the previous transfer station was Tribunal) have the required accessibility features. In our case study, we have requested a route starting at the Vodafone Sol station and ending at the Plaza Castilla station. Then, as previously indicated, user Y notifies events at Plaza de Castilla station and the app now notifies user X of their existence (see Fig. 12a and b).

TABLE IV EVENTS: JSON AND RDF FORMAT

<pre>{   "station": "Plaza de Castilla",   "line": 10,   "hasStairs": false,   "openDate": "2019-06-11T09:50:33.960",   "closedEvent": false }</pre>
<pre>&lt;http://manto.vortic3.com#Plaza%20de%20Castilla&gt; a   &lt;http://manto.vortic3.com#event&gt; ;   &lt;http://schema.org/name&gt; "Plaza de Castilla" ;   &lt;http://manto.vortic3.com#openDate&gt; "2019-06-11T09:50:33.960" ;   &lt;http://manto.vortic3.com#ofLine&gt; "10" ;   &lt;http://manto.vortic3.com#hasStairs&gt; "false" ;   &lt;http://manto.vortic3.com#closeDate&gt; "false" .</pre>

Then, the App offers an alternative route (Line 2 - red, and Line 9 - violet) including the closest accessible station (in our case study, the closest accessible station is Ventilla, also shown in Fig. 12b).

We have carried out many other tests according to the different user needs identified in this solution (auditory, visual and mobility disability, phobia to lifts and phobia to escalators). We have also verified that the routes provided by MMA4A are correct for these kinds of users.

In summary, our experience with the normalization process of smart accessibility data guarantees that the events dataset is correct and can be used to generate the routes for users with special needs, while the incidents are registered and considered while creating those routes. The validation experiment described here shows that the behaviour of the system is correct at this moment. Nonetheless, as in our previous work [1], we cannot guarantee that event notifications from the users will always be true.

## VII. CONCLUSION AND FUTURE WORK

One of the major challenges of smart city initiatives is to achieve an inclusive society for all citizens, including those with special needs. One of the pillars for that inclusion is urban mobility. Our work is focused on public transport as a part of that mobility. In order to address this challenge, more thorough information about the means of transport and their accessibility features is required, thus making it possible to arrange and provide accessible routes for everybody, including those with special needs.

Several web applications and tools that provide information and services for transport users already exist. We have studied some of them and, to the best of our knowledge, no software application currently takes the state of the public transport network with respect to its accessibility features at a particular moment fully into account. This is particularly the case as regards the existence of an application that updates data with a crowdsourcing strategy in order to compute accessible routes for special needs users.

Validation Results

Your RDF document validated successfully.

Triples of the Data Model

Number	Subject	Predicate	Object
1	<a href="http://com.vortic3.MANTO#VodafoneSol-lineal-escalator">http://com.vortic3.MANTO#VodafoneSol-lineal-escalator</a>	<a href="http://com.vortic3.MANTO#hasEscalator">http://com.vortic3.MANTO#hasEscalator</a>	"FALSE"
2	<a href="http://com.vortic3.MANTO#VodafoneSol-lineal-escalator">http://com.vortic3.MANTO#VodafoneSol-lineal-escalator</a>	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</a>	<a href="http://com.vortic3.MANTO#Event">http://com.vortic3.MANTO#Event</a>
3	<a href="http://com.vortic3.MANTO#VodafoneSol-lineal-escalator">http://com.vortic3.MANTO#VodafoneSol-lineal-escalator</a>	<a href="http://com.vortic3.MANTO#ofLine">http://com.vortic3.MANTO#ofLine</a>	"https://www.metromadrid.es//es/viaja_en_metro/red_de_met:
4	<a href="http://com.vortic3.MANTO#VodafoneSol-lineal-escalator">http://com.vortic3.MANTO#VodafoneSol-lineal-escalator</a>	<a href="http://com.vortic3.MANTO#openDate">http://com.vortic3.MANTO#openDate</a>	"2019-06-03T14:01:42.185"
5	<a href="http://com.vortic3.MANTO#VodafoneSol-lineal-escalator">http://com.vortic3.MANTO#VodafoneSol-lineal-escalator</a>	<a href="http://com.vortic3.MANTO#closedEvent">http://com.vortic3.MANTO#closedEvent</a>	"FALSE"
6	<a href="http://com.vortic3.MANTO#VodafoneSol-lineal-escalator">http://com.vortic3.MANTO#VodafoneSol-lineal-escalator</a>	<a href="http://com.vortic3.MANTO#StopPlace">http://com.vortic3.MANTO#StopPlace</a>	"https://www.metromadrid.es//es/viaja_en_metro/red_de_met:

Figure 11. Partial RDF Validator screen.

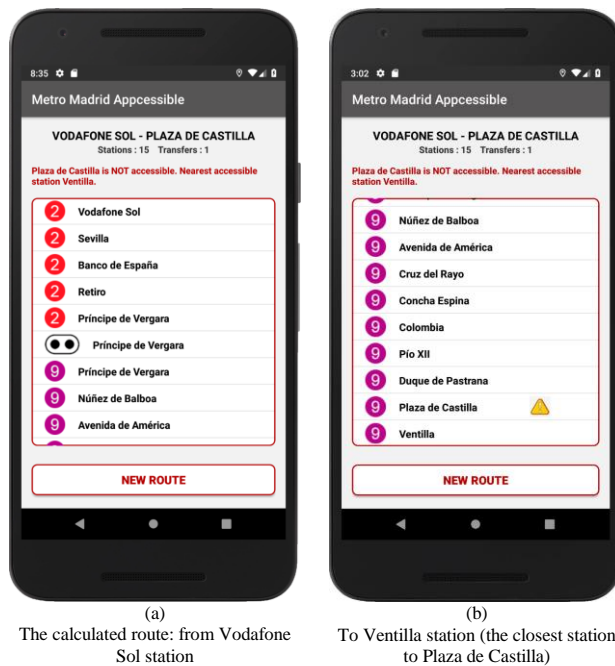


Figure 12. User interface of MM4A4.

In order to deal with this issue, we have proposed a *smart data architecture* to manage both the data sources related to the public transport infrastructure.

Our proposal considers accessibility features when it computes a route. But it goes beyond that, by taking real-time information into account, such as incidents that may occur at any time. Our dataset, therefore, provides a smarter response in real time. These real-time data, which originate from crowdsourcing, are collapsed and integrated

with existing accessibility information, thus providing a *real-time accessibility dataset*.

We provide a Kafka Queuing Manager, which is used to gather users' notifications about the state of accessibility features on public transport, and a Jena Semantic Repository to store both these data and the infrastructure data concerning the transport network. However, in order to save the data regarding notifications in this repository, it is necessary to convert them into RDF. Each time a notification is produced, data have to be transformed in real time. This work presents a smart data framework that is able to manage public transport accessibility data and to gather information in real time about incidents in the transport network. These data are annotated semantically and published as open data using automatic transformations defined with RML. This makes it possible to ensure that data are immediately available for the calculation of new accessible routes. Furthermore, the use of this language provides the possibility of integrating new data into existing datasets.

We have also developed a smart app (still a prototype) for public transport users, which is able to compute accessible routes by taking the user's needs into account, and also provide notifications of incidents or events as regards the accessibility features in the network. In order to semantically annotate the data before storing them, we have also defined a specific vocabulary as a domain-specific RDF Schema.

With regard to future work in this area, we intend to improve the MMA4A app by introducing some characteristics of the behaviour change support systems. These kinds of systems take persuasive techniques into account with the aim of engaging and retaining the app users [47][49]. In our case, retaining the app user could imply many advantages for people with special needs, because the app user will update a lot of public transport and accessibility information. Moreover, we intend to

include more public transport information from other sources and means of transport, and to integrate them into the current architecture. It will, therefore, be necessary to semantically harmonize them using our MAnTO ontology, which has already been used to semantically annotate the data stored in the infrastructure collection of our Jena repository. This will make it possible to provide fully open datasets for different public transport networks. Moreover, these datasets must be published on an open platform, thus providing free access to accessibility and special needs data. In this respect, the use of RML to define the transformation into RDF is very useful as regards integrating different datasets.

In order to guarantee that the information from the crowdsourcing is truthful, we want to incorporate a user registration system and a supervision module that determines the importance of the incidence to be incorporated into the system.

We have also worked on gathering information about accessible pedestrian routes in the city, obtained via crowdsourcing techniques, capturing the geographical information and accessibility features on these routes. This information will also be incorporated as smart data into the Multiply@City platform.

#### ACKNOWLEDGMENTS

This work is supported by the Multiply@City and the Access@City projects (TIN2016-78103-C2-1-R), funded by the Spanish Ministry of Science, Innovation and University. We would also like to thank Isaac Lozano Osorio for his assistance during the development process.

#### REFERENCES

- [1] P. Cáceres, C. E. Cuesta, A. Sierra-Alonso, B. Vela, J. M. Cavero, and A. Garrido, "Even Smarter Data: Using Crowdsourcing to Improve Accessibility in Real-Time." In Proceedings on the Fourth International Conference on Universal Accessibility in the Internet of Things and Smart Environments (SMART ACCESSIBILITY 2019), pp. 26-32, 2019.
- [2] E.J. Tomaszewska and A. Florea, "Urban smart mobility in the scientific literature - bibliometric analysis", *Engineering Management in Production and Services*, Volume 10, Issue 2, pp. 41-56, 2018, ISSN 2543-6597, Poland, DOI: 10.2478/emj-2018-0010.
- [3] Gaggi, S., Fluhrer, T., and Janitzek, T. (2013). Innovation in urban mobility: Policy making and planning. Directorate-General for Mobility and Transport. Luxembourg: European Union. <https://www.kowi.de/Portaldata/2/Recourses/fp/tripurban-mobility.pdf>, [retrieved: 2019.06.30].
- [4] P. Cáceres, A. Sierra-Alonso, B. Vela, J. M. Cavero, and C. E. Cuesta, "Towards Smart Public Transport Data: A Specific Process to Generate Datasets Containing Public Transport Accessibility Information" In Proceedings on the Third International Conference on Universal Accessibility in the Internet of Things and Smart Environments (SMART ACCESSIBILITY 2018), pp. 66-71, Rome (Italy), 2018.
- [5] P. Cáceres, A. Sierra-Alonso, C. E. Cuesta, J. M. Cavero, and B. Vela, "Enhancing Accessibility Information in Google Maps: Adding new pieces of information to GTFS to improve accessibility" In Proceedings on the Second International Conference on Universal Accessibility in the Internet of Things and Smart Environments (SMART ACCESSIBILITY 2017), pp. 6-9, Nice (France), 2017.
- [6] P. Cáceres, C. E. Cuesta, B. Vela, J. M. Cavero, and A. Sierra-Alonso, "Smart Data at Play: Improving Accessibility in the Urban Transport System", *Journal of Behaviour & Information Technology*, ISSN: 0144-929X, To be published.
- [7] P. Cáceres, A. Sierra-Alonso, C. E. Cuesta, B. Vela, and J. M. Cavero, "Modelling and Linking Accessibility Data in the Public Bus Network", *Journal of Universal Computer Science*, vol. 21, no. 6, pp. 777-795, 2015.
- [8] T. Sarjakoski et al., "Landmarks and a Hiking Ontology to Support Wayfinding in a National Park During Different Seasons". In M. Raubal, D. M. Mark, and A. U. Frank (Eds.), *Cognitive and Linguistic Aspects of Geographic Space*, Lecture Notes in Geoinformation and Cartography. Springer, pp. 99-119, 2013. DOI:10.1007/978-3-642-34359-9.
- [9] Wheelmate, <http://myhealthapps.net/app/details/13/wheelmate>, [retrieved: 2018.12.22].
- [10] Wheelmap, <https://wheelmap.org/>, [retrieved: 2019.05.22].
- [11] Accessmap, <https://www.accessmap.io/>, [retrieved: Dec. 2019.05.22].
- [12] Ciudades Patrimonio de la Humanidad, <http://www.ciudadespatrimonio.org/accesibilidad/?idioma=en>, [retrieved: 2019.6.22].
- [13] Access2All, EU FP7 project, Mobility schemes ensuring accessibility of public transport for all users, <https://cordis.europa.eu/project/rcn/89881/factsheet/en>, 2010, [retrieved: 2019.06.30].
- [14] Mediate, EU FP7 project - Methodology for describing the accessibility of transport in Europe, <https://trimis.ec.europa.eu/>, 2011, [retrieved: 2019.06.30].
- [15] J. Zimmerman et al., "Field Trial of Tiramisu: Crowd-Sourcing Bus Arrival Times to Spur Codesign". In Proc. CHI 2011, New York, pp. 1677-1686, 2011.
- [16] Moovit, <https://moovit.com/>, [retrieved: 2019.06.30].
- [17] Swiftly, <https://www.goswift.ly/>, [retrieved: 2019.06.30].
- [18] B. Ferris, K. Watkins, and A. Borning, "OneBusAway: Results from Providing Real-Time Arrival Information for Public Transit". In Proc. CHI 2010, Atlanta, Georgia, April 10-15, pp. 1807-1816, 2010.
- [19] BUSUP, <https://www.busup.com/>, [retrieved: 2019.06.30].
- [20] CIVITAS Initiative (2012), "Cleaner and Better Transport in Cities", <http://www.civitas.eu>, [retrieved: 2019.07.20].
- [21] OpenTRipPlanner, <http://www.opentripplanner.org/>, [retrieved: 2019.07.20].
- [22] OpenStreetMap, <https://www.openstreetmap.org>, [retrieved: 2019.07.20].
- [23] GTFS, Google Transit Feed Specification, <https://developers.google.com/transit/gtfs/>, [retrieved: 2019.05.20].
- [24] A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, and R. Van de Walle, "RML: A generic language for integrated RDF mappings of heterogeneous data", in: Proceedings of the 7th Workshop on Linked Data on the Web, 2014.
- [25] S. Das, S. Sundara, and R. Cyganiak. "R2RML: RDB to RDF Mapping Language", W3C Recommendation, 2012, <https://www.w3.org/TR/r2rml/>, [retrieved: 2019.07.20].
- [26] A. Langegger and W. Wöß. "XLWrap - Querying and Integrating Arbitrary Spreadsheets with SPARQL". In Proceedings of the 8th International Semantic

- WebConference, ISWC'09, pp. 359-374. Springer-Verlag, 2009.
- [27] Tarlq, SPARQL for Tables, <https://github.com/tarlq/>, [retrieved: 2019.07.20].
- [28] OpenRefine, <http://openrefine.org/>, [retrieved: 2019.07.20].
- [29] Virtuoso, Springer, <http://vos.openlinksw.com/owiki/wiki/VOS/VirtSponger>, [retrieved: 2019.07.20].
- [30] C. E. Cuesta, P. Cáceres, B. Vela, and J. M. Cavero, "CoMobility: A Mobile Platform for Transport Sharing", ERCIM News, no. 93, pp. 22-23, 2013.
- [31] Access@City project (TIN2016-78103-C2-1-R), funded by the Spanish Ministry of Science, Innov. & University, 2016.
- [32] Multiply@City project (TIN2016-78103-C2-1-R), funded by the Spanish Ministry of Science, Innov. & University, 2016.
- [33] Regional Consortium for Public Transports of Madrid, <http://datos.crtm.es/>, [retrieved: 2019.07.20].
- [34] EMT Madrid home page, <http://www.emtmadrid.es/Index.aspx?lang=en-GB>, [retrieved: 2019.07.20].
- [35] Spanish Society of Blind People (ONCE, Organización Nacional de Ciegos de España) <http://www.once.es/new/otras-webs/english>, [retrieved: 2019.07.20].
- [36] Transmodel standardization, E. C. Transmodel, Road Transport and Traffic Telematics Public Transport. Ref. Data Model, EN 12896, 2019, <http://www.transmodel-cen.eu/>, [retrieved: 2019.07.20].
- [37] CEN/TC 278. Intelligent transport systems - Public transport - Identification of Fixed Objects In Public Transport (IFOPT), EN 28701:2012.
- [38] D. Brickley and R.V. Guha, "RDF Schema 1.1" W3C Recommendation, 2014, <https://www.w3.org/TR/rdf-schema/>, [retrieved: 2019.06.20].
- [39] SpringBoot. (2019). Spring Boot. <http://spring.io/projects/spring-boot>, [retrieved: 2019.06.20].
- [40] Apache Kafka (n.d.). <https://kafka.apache.org/intro>, [retrieved: 2019.06.20].
- [41] Apache Jena (n.d.), <https://jena.apache.org/index.html>, [retrieved: 2019.06.20].
- [42] RDF Working Group, "Resource Description Framework (RDF)" W3C Recommendations, 2014, <http://www.w3.org/RDF/>, [retrieved: 2019.07.20].
- [43] R. Cyganiak, D. Wood, and M. Lanthaler, "RDF 1.1: Concepts and Abstract Syntax". W3C Recommendation, 2014, <https://www.w3.org/TR/rdf11-concepts/>, [retrieved: 2019.07.20].
- [44] T. Berners-Lee, R. Fielding, and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax" 2005. [http://www.hjp.at/doc/rfc/rfc3986.html#sec\\_1.1](http://www.hjp.at/doc/rfc/rfc3986.html#sec_1.1), [retrieved: 2019.07.20].
- [45] F. Gandon and G. Schreiber, "RDF 1.1 XML Syntax" W3C Recommendation, 2014. <https://www.w3.org/TR/rdf-syntax-grammar/>, [retrieved: 2019.07.20].
- [46] E. Prud'hommeaux and A. Seaborne, "SPARQL 1.1 Query Language for RDF", W3C Recommendation, 2008, <https://www.w3.org/TR/rdf-sparql-query/>, [retrieved: 2019.07.20].
- [47] A. Dimou, D. Kontokostas, M. Freudenberg, R. Verborgh, J. Lehmann, E. Mannens, S. Hellmann, and R.V. de Walle, "Assessing and refining mappings to RDF to improve dataset quality", in: The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, Proceedings, Part II, pp. 133-149, 2015.
- [48] T. Lehto and H. Oinas-Kukkonen, "Explaining and predicting perceived effectiveness and use continuance intention of a behaviour change support system for weight loss", *Behaviour & Information Technology*, 34(2), 176-189, 2015.
- [49] H. M. Mohadis, N. Mohamad Ali, and A. F. Smeaton, "Designing a persuasive physical activity application for older workers: understanding end-user perceptions", *Behaviour & Information Technology*, 35(12), 1102-1114, 2016.