Applying the LOT methodology to a Public Bus Transport Ontology aligned with Transmodel: Challenges and Results

Edna Ruckhaus [a,*], Adolfo Antón Bravo [a], Mario Scrocca [b] and Oscar Corcho [a]
[a] *Ontology Engineering Group, Universidad Politécnica de Madrid, Spain*
E-mails: eruckhaus@fi.upm.es, adolfo.anton.bravo@upm.es, ocorcho@fi.upm.es
[b] *Cefriel – Politecnico di Milano, Milano, Italy*
E-mail: mario.scrocca@cefriel.com

Abstract. We present an ontology that describes the domain of Public Transport by bus, which is common in cities around the world. This ontology is aligned to Transmodel, a reference model which is available as a UML specification and which was developed to foster interoperability of data about transport systems across Europe. The alignment with such a complex non-ontological resource required the adaptation of the Linked Open Terms (LOT) methodology, which has been used by our team as the methodological framework for the development of many ontologies used for the publication of open city data. The ontology is structured into three main modules: (1) agencies, operators and the lines that they manage, (2) lines, routes, stops and journey patterns, and (3) planned vehicle journeys with their timetables and service calendars. Besides reusing Transmodel concepts, the ontology also reuses common ontology design patterns from GeoSPARQL and the SSN ontology. As part of the LOT data-driven validation stage, RDF data has been generated taking as input the GTFS feed provided by the Madrid public bus transport provider (EMT). Data transformation rules were expressed using RML mappings, and materialised, and queries corresponding to competency questions were developed and tested. Currently, a generic and reusable REST API is being developed and it can be adopted by other organizations to standardize the publication of open data in this domain.

Keywords: Ontology, Transmodel, Public Bus, Open Cities, RDF

1. Introduction

Open data initiatives among cities in Spain date back more than a decade, with relevant landmarks associated to the first transposition of the EU Public Sector Information directive in 2007[1], the publication of the UNE 178301:2015 technical norm on Open Data for Smart Cities[2] and the development of the open data guide by the Spanish Federation of Municipalities and Provinces (FEMP) in 2017 [1] and the catalogue of high-value open datasets for cities in 2019 [2], in domains such as public sector, demography, environment, economy, commerce, transport and treasury. As part of the initiatives and projects that have led the advancement of open data among cities in Spain we can cite the Ciudades Abiertas (Open Cities http://ciudadesabiertas.es) project, a public-private collaborative project led by four Spanish municipalities (Zaragoza, Madrid, Santiago de Compostela and A Coruña) with the general aim to facilitate the implementation of common Open Government policies that are reusable by many other municipalities inside and outside Spain.

Among the project actions on open data, citizen participation and transparency, several (12) ontologies are being developed using the Linked Open Terms (LOT) methodology [3, 4]. These ontologies allow publishing Open Data homogeneously and following

*Corresponding author. E-mail: eruckhaus@fi.upm.es.
[1] https://www.boe.es/eli/es/l/2007/11/16/37/con
[2] https://www.en.aenor.com/normas-y-libros/buscador-de-normas/une?c=N0054318

Linked Data principles [5]. They are being added to those that had been already developed in the context of the Spanish network of Open Data for Smart Cities (https://github.com/opencitydata), and they correspond to a subset of the catalogue of datasets included in the aforementioned FEMP open data guide [2].The current catalogue of ontologies is available at http://vocab.ciudadesabiertas.es/. All of the ontologies are publicly available and versioned in GitHub, with the corresponding repositories including use cases and user stories, requirements, the ontology implementation in OWL, the ontology HTML documentation in Spanish and English, and example data and queries.

In the area of transport, three ontologies have been developed so far under the umbrella of these initiatives, focused on the representation of open data about Public Bicycles, Motor Vehicle Traffic and Public Bus Transport. In this paper, we will discuss the latter, an ontology that has been specifically developed for structuring how to publish open data about public buses in cities, beyond the current publication as GTFS feeds (Google Transit Feed Specification). We have named it the Public Bus Transport ontology (http://vocab.ciudadesabiertas.es/def/transporte/autobus). Its corresponding GitHub repository with all the intermediate and final artefacts is available at https://github.com/CiudadesAbiertas/vocab-transporte-autobus.

As a general description of the scope of this ontology, the requirements are focused on two types of information about public buses: static and dynamic. Static information refers to lines, stops, routes for each line, timetables, and planned events that may affect a line. Dynamic information refers to the expected arrival time of buses at each stop, and also unplanned incidents such as accidents, protests, etc. Having data on public buses (extensible to other means of transport) is a very valuable element for municipalities and citizens, as well as for third parties. Such data is used for the management of services by transport operators, for the use of these services by citizens, and also for the provision of third-party services in different areas such as traffic management, road infrastructure design and trip planning.

One important assumption that we needed to consider in the development of this ontology was the alignment with European policies, in particular the regulation 2017/1926 [6] that states that starting December 2019, any operator or transport authority must offer its data in formats compatible with Transmodel [7], the European reference data model for public transport information developed by the European Committee for Standardization (CEN). Transmodel underpins two concrete data formats that have been established for the exchange of Transport data: NeTEx [8] and SIRI [9]. Additionally, the European SNAP project [10], where the authors of this paper have been also involved, had already developed an initial ontological transposition of Transmodel in order to facilitate interoperability among these formats and other popular formats such as GTFS.

This alignment to Transmodel has been fulfilled through the application of the LOT methodology, a reuse-based methodology specifically focused on developing ontologies and vocabularies for the generation of Linked Data. In a usual ontology reuse fashion, we tried to map the concepts in our domain with those developed in the Transmodel component ontologies, i.e. ontologies in the SNAP Vocabulary catalog [11]. The main drawbacks to this reuse process were that some of the component ontologies are quite large and complex because they encompass several of the models in the original UML specification (e.g. the Journeys ontology) and also some classes, attributes and restrictions of the Transmodel specification were not developed in the ontology at all or were in an early stage of development, since the goals of the SNAP project were focused on the GTFS transposition of only some of the concepts in Transmodel and NeTEx. Because of these drawbacks it was necessary to resort to the reuse of the non-ontological UML Transmodel specification. Thus, we were faced with the challenge of reusing a semantically rich and complex non-ontological resource.

We then adapted the reuse process in LOT, in order to consider the non-ontological UML specification. Once we started to reuse concepts in the different models that are part of the Transmodel UML specification, we realized that a single conceptual model for the Public Bus Transport ontology was very hard to handle due to the complexity and variety of concepts that were represented. Based on the nature of the concepts defined in Transmodel, we divided our ontological conceptualization into three major parts: (1) Agencies, operators and the lines they manage, (2) Lines, routes, stops, and journey patterns, and (3) Planned vehicle journeys with timetables and service calendars. We also added the relationships (properties) among these three ontology modules. However, the encoding was done as one ontology in order to simplify its publication and future maintenance.

Because of the complexity and sometimes contrived conceptual design, we set ourselves to generate RDF examples of real-world data annotated with the ontology concepts, which could be used for testing purposes as well. RDF data were generated taking as input the GTFS feed provided by the Madrid public bus transport provider (Empresa Municipal de Transportes de Madrid). Data transformations were expressed using RML mappings, and materialised into RDF, and SPARQL queries corresponding to the competency questions established during the requirements identification stage were tested.

In the next section we provide some preliminaries on GTFS and Transmodel and also describe the SNAP project and the LOT methodology. Following, we describe our adaptation of the LOT methodology and its application to our use case. The next section describes in detail the three portions of the Public Bus Transport ontology. Then we describe the alignment to Transmodel and the challenges encountered, and finally we give our conclusions and future lines of work.

2. Preliminaries

In this section we describe current models for the representation and exchange of transport data. We also describe the SNAP project, the Transmodel ontology developed in that project, and the general stages of the LOT methodology, which we have used for our ontology development activities.

2.1. *Transport Data Standards*

Following, we will describe the GTFS standard and Transmodel.

2.1.1. *GTFS*

The General Transit Feed Specification (GTFS) [12] is a *de-facto standard* developed by Google and split into a static component, GTFS Static, and a real-time component, GTFS Realtime, that contains arrival predictions, vehicle positions and service advisories. It has become popular due to its simplicity and to the fact that it has been adopted not only by Google Maps, but also by other route planning systems such as *Open Trip Planner*, or *Navita.io*.

GTFS Static was developed for sharing the transit static information on agencies, routes and their stops, schedules, fares, among others. The specification defines the headers of seventeen CSV files. Each file, as well as their headers, can be mandatory or optional. In particular, Madrid's transport authority (Consorcio Regional de Transportes de Madrid) and Madrid's public bus operator (EMT) provide a GTFS feed that includes information on all of the entities except for Fares. The most recent data is of April 2020.

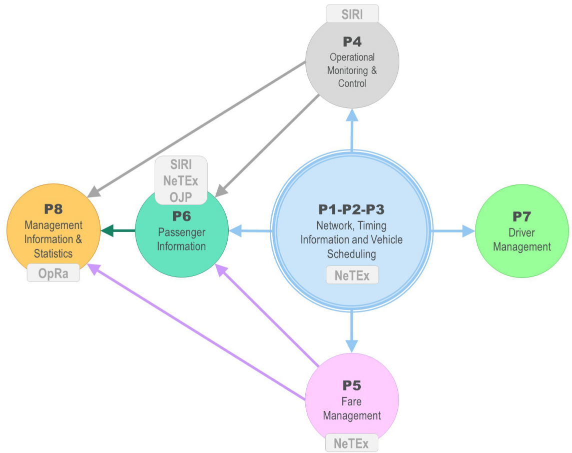

Fig. 1. Subdomains of the Transmodel ontology, their relationships and associated standards.

2.1.2. *Transmodel Data Model*

The Transmodel European Reference Data Model for Public Transport, from now on denoted as Transmodel, has been developed in the context of the Directive 2010/40/EU [6] on the framework for the deployment of Intelligent Transport Systems (ITS). In particular the priority action A includes the definition of the necessary requirements to make EU-wide multimodal travel information services accurate and available across borders to ITS users. This includes the availability of existing and accurate multimodal transport data, and the facilitation of electronic data exchange between public transport stakeholders.

Transmodel provides a complete and extensive set of related concepts that covers diverse aspects of Public Transport information for different transport modes. The standard covers concepts on eight subdomains [7]: Network, Timing Information, Vehicle Scheduling, Operational Monitoring and Control, Fare Management, Passenger Information, Driver Management, and Management Information and Statistics. The connections among these subdomains are shown in Figure 1.

Our Public Bus Transport ontology covers agencies, operators, lines, routes and their stops, timetables, and arrival times. It does not cover passenger, driver or fare information, nor operational monitoring and control or management statistics. Thus, we will describe the main concepts represented in the Network and in the Vehicle Scheduling, and the Timing Information subdomains.

The Network subdomain [13] represents the topological descriptions of the spatial structure of a public transport network which is built with points. An entity Point is defined as the most basic entity of the network model. It marks the location of bus stops, parking places or other types of points. Links represent 1-dimensional connections between points. An ordered set of points or links is called a Link Sequence. These are the generic building blocks of the Public Transport network model. Their specialisations represent concrete special Public Transport objects, like scheduled stop points, routes, journey patterns, among others.

The Timing Information and Vehicle Scheduling subdomains [13], represent a Vehicle Journey to describe the movements of a transport vehicle from the start point to the end point of a journey pattern on an operating day. Among the common concepts that cut across the subdomains are the ones related to organisations [13], as different aspects of public transport could be handled by different organisation parts, and sometimes are subcontracted to third parties.

2.1.3. *Transmodel UML Specification*

The current Transmodel UML specification is the revised V6.0 version (http://www.transmodel-cen.eu/model/index.htm). It has been divided into eight packages, seven of these packages correspond to the subdomains enumerated above, Timing Information and Vehicle Scheduling have been joined into one package. Besides these packages, there is a Common Concepts package [13] that covers concepts that are shared by the different functional domains. Models in each package are subdivided into more specific submodels up to three levels. We would like to point out that time-related concepts are represented in the Service Calendar submodel in the Common Concepts package, in the Tactical Planning Components submodel in the Network Topology package, and in all of the submodels in the Timing Information and Vehicle Scheduling package. This gives an idea if the extension and complexity of the specification.

Our Public Bus Transport ontology is aligned to concepts in the Network and the Vehicle Scheduling and Timing Information packages. Additionally, regarding the Common Concepts package, it is aligned to concepts in the Responsibility, Generic Framework, and Reusable Components models.

3. The SNAP Project. Semantic National Access Point

The European regulation 2017/1926, requires each European Member State to set up a National Access Point (NAP), for multimodal travel information for all transport modes. Each transport stakeholder, should contribute to the NAP with their static and dynamic data, using a set of standard data formats identified by the regulation and based on the European Standard *Public Transport Reference Data Model*, i.e., Transmodel. Specifically, concerning the exchange of static scheduled data, the relevant data in the NAP should use the CEN data exchange standard NeTEx [8]. For the exchange of real-time public transport data, the relevant parts of the CEN public transport data exchange standard SIRI [9] are used.

The SNAP project developed a solution for transport stakeholders that need to convert their data into formats required by the regulation. The proposed solution, based on Semantic Web technologies, implements data conversion, meanwhile, supporting the constitution of a knowledge graph of multi-modal transport data [14]. The SNAP converter adopts a reference ontology as a global conceptual model enabling a two-step conversion between two standards in the transport domain: first, from the input standard to the ontology, and then from the ontology to the target standard. To enable this solution, the SNAP project kick-started an effort to define an ontological transposition of Transmodel[3], as the reference ontology used in the conversion process.

As already mentioned, Transmodel is a really large and complex model. The SNAP project, to start validating the proposed solution, implemented a first version of the ontology focusing on a portion of the overall specification. The initial modules of the Transmodel ontology have been defined to enable the conversion of a static GTFS [12] feed, widely adopted among transport stakeholders, to NeTEx, the required standard by the European regulation. Given that NeTEx is almost a serialization of Transmodel, the GTFS specification has been used to identify the relevant portion of Transmodel that should be prioritized in the ontology engineering process.

The Transmodel ontology currently defines five modules, shown in Figure 2, that cannot be directly mapped on the presented Transmodel sub-domains,

[3]Published online at http://w3id.org/transmodel

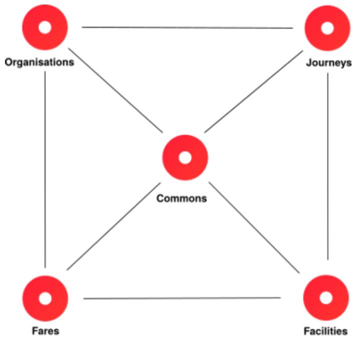

Fig. 2. Initial modules of the Transmodel ontology and their relations.

but reuse their terminology to define concepts and properties. To obtain a proper ontological transposition, the implemented modules integrate newly defined entities with already available vocabularies (e.g., the Organization ontology[4], the Geo WGS84 ontology[5], Schema.org[6], etc.).The *Commons* module defines general concepts and properties that can be reused across all the other modules. The *Organisations* module can be used to describe different information about entities operating and/or offering transportation services. The *Journeys* module can be used to represent data related to a transportation service, e.g., time tables, routes, vehicles and their scheduling, etc.. The *Facilities* module contains concepts and properties to describe facilities and, in particular, stations and stop places. Finally, the *Fares* module contains a simplified model for information about fares, that needs to be extended to support the full related Transmodel sub-domain.

3.1. LOT methodology

LOT is a lightweight methodology for the development of ontologies and vocabularies [3]. It is based on the previous NeOn methodology [15] and includes the following four major stages that can be seen in Figure 3: (1) Requirements Specification, (2) Implementation, (3) Publication, and (4) Maintenance.

The aim of the ontology Requirements Specification stage is to identify and define the requirements the ontology should fulfil [4]. At the beginning of this stage, the goal and scope of the ontology is defined, following this, the domain is analyzed in more detail by looking at documentation, data that has been published, standards, formats, among others. Also, the use cases and user stories are identified. Then, the requirements in form of competency questions and statements are specified and validated by the stakeholders.

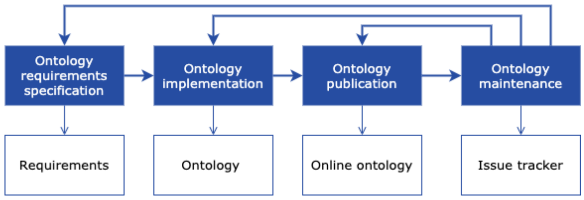

Fig. 3. Linked Open Data Methodology (LOT) Stages.

The goal of the Implementation stage is to build the ontology using a formal language, based on the ontological requirements identified by the domain experts [4]. This stage is developed iteratively through several sprints and it is comprised of the Conceptualization, Encoding, and Evaluation processes. During the conceptualization process, an ontology model is built and represented in a graphical language. We follow the Chowlk visual notation[7] that ”provides a set of visual blocks to represent each element from the OWL ontology implementation language...”.

One of the activities of the implementation stage is ontology reuse. LOT is based in the NeOn methodology which develops nine scenarios for the development of ontology networks. Among these scenarios there is Scenario 3 on reusing ontological resources. In this scenario, ontology developers reuse ontological resources (ontologies as a whole, ontology modules, and/or ontology statements). Developers search, assess, compare, select, and integrate the ontological resources. Terms that have been extracted from the requirements, that correspond to entities and relationships, may be used for searching existing ontologies that cover these concepts. Scenario 2 considers the reuse of non-ontological resources (NORs), it is meant for the transformation of mostly textual NORs with underlying low epressivity models such as thesaurus, classification schemes, etc.

In the Encoding process, the ontology development team generates a computable model represented in the OWL language. The Evaluation process includes two aspects: (1) ensuring that the requirements are fulfilled; this is done through the translation of the competency questions into the corresponding SPARQL queries, and executing these queries against RDF test

[4]https://www.w3.org/TR/vocab-org/
[5]https://www.w3.org/2003/01/geo/
[6]https://schema.org/docs/schemas.html
[7]https://github.com/oeg-upm/chowlk_spec

data that has been annotated with terms in the ontology, (2) guaranteeing that the ontology does not have syntactic, modelling or semantic errors. The syntactic validation may be done with any existing OWL validator tool, and the semantic and modelling evaluation is done by running an OWL (DL) reasoner, and by discovering modelling pitfalls. Currently, we use the OOPS! tool [16] for this task.

The aim of the Publication stage is to provide an online human-readable documentation which generally includes metadata (e.g. creators, contributors, creation date, version), a description of the conceptual model diagram, and all of the class and property restrictions and annotation properties. The ontology and all of its associated documents are usually published as a public repository. The Maintenance stage includes updates to ontology requirements that were not originally identified, and improvements, which in consequence may derive in another ontology development iteration.

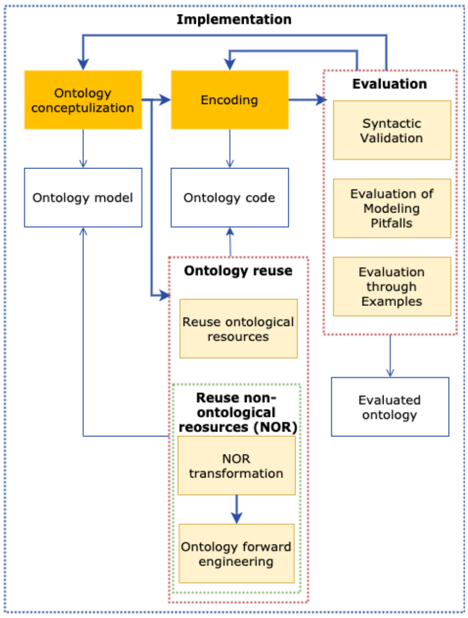

Fig. 4. Public Bus Transport Ontology. Adaptation of LOT Implementation Stage. The ”Ontology reuse” activity is expanded and the ”Evaluation through Examples” has been added to the Evaluation activity.

4. Adaptation of the LOT methodology

The description of the adaptation of the LOT methodology is focused on the ontology Implementation stage, specifically we have adapted the activities of conceptualization, ontology reuse, and evaluation in order to carry out the alignment to Transmodel. The reuse includes ontological resources as well as non-ontological resources (the Transmodel UML specification, the glossary of Transmodel terms and the NeTEx Schema). Also, we have added an additional form of evaluation, that is, the validation through RDF real-world examples. The adaptation of the Implementation stage can be seen in Figure 4.

For the Public Bus Transport ontology, we developed an initial conceptual model without reuse of Transmodel, that satisfied the requirements identified in the first stage of the methodology. Next, in the reuse activity, under the scenario 2 of reuse of ontological resources, we tried to map the concepts in the initial conceptualization with those developed in the Transmodel component ontologies, i.e. ontologies in the SNAP Vocabulary catalog [10]. We reused some of the concepts in these component ontologies such as Transmodel Organisations, but were not able to reuse the rest of the ontologies in the SNAP catalogue. The main drawbacks were that some of the component ontologies are quite large and complex because they encompass several of the models in the original UML specification (e.g. the Journeys ontology) and also some classes, attributes and restrictions of the Transmodel specification were not developed: the ontologies contain those concepts where there exists a mapping from the GTFS feed specification to Transmodel.

Because of these drawbacks it was necessary to resort to scenario 2 of the NeOn methodology that covers the reuse of non-ontological resources (NORs), in this case the reuse of the UML Transmodel specification. To this effect, the NeOn methodology [15] defines a re-engineering pattern which is comprised of three main activities: (1) NOR reverse engineering to create an abstract representation of the resource, (2) NOR transformation to create the ontological model, and (3) Ontology forward engineering to generate an implementation of an ontology. Although these activities are meant for the transformation of NORs such as thesaurus, classification schemes, lexicons, etc., they can be applied to semantically richer UML specifications. For UML specifications there is already an abstract representation of the resource, so step (1), reverse engineering, is not required. NOR transformation implies translating the UML representation to an ontology representation, and Ontology forward engineering consists in the implementation of the relevant parts of the reused resource. It should be mentioned that although we reused the UML specification, prefixes and entity names that are defined in the Transmodel ontologies were used in our conceptualization.

Following the reuse process, we realized that a single conceptual model for the Public Bus Transport ontology was very hard to manage due to the complexity and variety of concepts that were represented. Thus, we divided the ontology into three component ontology conceptualizations.

Additionally, because of the complexity and sometimes contrived conceptual design, we set ourselves to generate examples of real-world data annotated with the ontology. In the LOT methodology these examples are part of the Publication stage. However, they were used during the evaluation activity to validate with the public transport experts if the Transmodel concepts included in our ontology did in fact represent our specific Public Bus Transport domain. This verification triggered some adjustments on the ontology conceptual design.

5. Public Bus Transport Ontology

5.1. Public Bus Transport Ontology

In this section we will describe the application and results of the LOT stages and activities.

5.1.1. Requirements specification

The Public Bus Transport ontology represents information about the public urban bus service for municipalities in Spain. The requirements cover transport authorities and operators, information on lines, routes, journey patterns and their timetables, stops in each route, information on expected bus arrival times for each stop, and information on planned and unplanned incidents that may affect the bus routes and their journeys. Our use cases[8] are summarized as follows:

1. The bus is a transport mode that reaches most parts of the city and where value-added services can be provided by the operator or third parties. A user requires information on the route(s) and their destination, timetables, and the stops to board and alight. A third party app could compute in real time the best route based on stop arrival times.
2. The quality of the public bus network is related to offering an adequate public bus service, it is essential to know if it is being provided normally or if incidents are affecting it in such a way that corrective actions are needed.
3. Journalists or researchers, non-governmental organizations, among others, may want to analyze if public transport services are being provided "correctly": if the frequency established for the line and its routes is met, and in general, the degree of fulfillment of the service.

Requirements in the form of statements and competency questions were grouped in sections: transport service, lines, stops, incidents, and buses. It should be mentioned that specific vehicles (buses) and their information, are not part of the scope of this vocabulary. However, expected arrival times of buses at each stop is one of the requirements. It should be noted that part of the competency questions are related to geospatial data, for example ”Which are the stops that are nearest to a certain location in a 500 mt. ratio?”

5.1.2. Implementation

An initial conceptualization model was developed with reuse of the GeoSPARQL pattern to represent the stop location, and of the https://schema.org/ `Organization` class and properties. The reuse of the GeoSPARQL location pattern has been one of the practices for open cities ontologies, as well as the reuse in general, of schema.org concepts. The model covers roughly the main entities and relationships in the requirements as can be seen in Figure 5. It should be noted that in case of following a no-reuse approach, this model would have needed further refinement in order to cover aspects of the requirements such as different route patterns, timetables, and expected arrival times.

The next activity in this stage involved the reuse of the Transmodel ontologies developed in the SNAP project. A first attempt to reuse these ontologies resulted in a very large and complex conceptual model. At this point we divided the ontology conceptualization into three parts: (1) Bus organisations, (2) Bus routes, and (3) Bus vehicle journeys.

Part I - Bus organisations (Base Model).

The reuse of the SNAP Organisations ontology (`tmorg`) seemed straightforward so we included this portion of the SNAP ontology. This conceptualization can be considered a high level model for this ontology as it covers the information on bus operators and the authorities they serve to, the lines that they manage

[8] https://github.com/CiudadesAbiertas/vocab-transporte-autobus/wiki

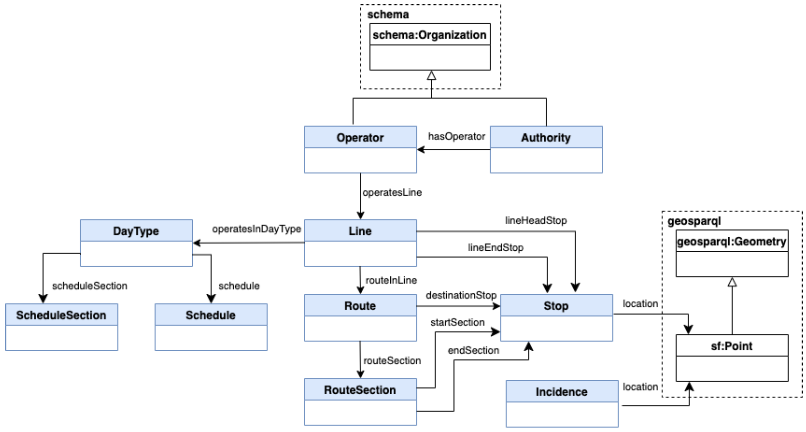

Fig. 5. Public Bus Transport Ontology. Initial Conceptual Model.

together with their graphical "presentation", and the routes made by each line (journeys and timetables are expanded in the other parts). The conceptual model is shown in Figure 6. In this model we also represent information on Incidences for which we reuse the Traffic vocabulary[9] developed in the context of the Ciudades Abiertas project. We reuse the Transmodel `Line` concept, however, a subclass `esautob:Line` was created to relate the line to the incidences that affect it.

Part II - Bus routes

We followed the steps described in Section 4 for the reuse of the non-ontological Transmodel UML specification: (2) NOR transformation to create the ontological model, and (3) Ontology forward engineering to create an implementation. The ontological graphical representation (step (2)) of parts of the Line Network and Route submodels of the UML specification can be seen in Figure 7; this step was necessary to ensure the correct (semantic) reuse of these concepts. We then integrated this partial model into the complete conceptual model (step (3)) that is shown in Figure 8.

This part reuses the Transmodel `Line` concept. Again, the subclass `esautob:Line` is defined in order to relate the line to the stops at the beginning and end of the line. A line is made up of several routes, and each route is composed of a series of points on the route, each point in the route is associated to a point that is the functional centroid for a certain place; we defined a class for stop, `esautob:Stop`, as a subclass of `Place`, due to the need to represent data and object properties that are specific to our domain. We relate the stop to the postal address; we reuse the existing Postal Address ontology[10]. The location of the stop is represented through the GeoSPARQL geolocation pattern (this was also part of the initial conceptual model). Each route may have several journey patterns, requirements state that stops may vary for example during weekends, and a different journey pattern may have been generated by an incidence. For expected stop arrival times, we reused the SOSA ontology[11] in order to represent the arrival times as observations. Note that the Journeys ontology prefix (`tmjourney`) and its entity names were reused. An example of RDF data for this model where there are two journey patterns is shown in Figure 9.

Part III - Bus vehicle journeys

This part presented in Figure 10 represents the planned vehicle journeys and its service data (timetables). Similarly to Part II, we represented the corresponding portions of the UML specification in our graphical notation, and then integrated this partial model into the conceptual model. A vehicle journey follows a certain journey pattern and can be made in one or more day types like for example a holiday or weekday. A service calendar has beginning and ending dates, and each day in the service calendar is associated to a day type. Thus, on a certain date, the information on which vehicle journeys are planned for that date may be extracted from the model. Also, each vehicle journey as it is frequency-based, is associated to a headway journey group which is determined by minimum, maximum and planned headway intervals. This part mainly reuses the `tmjourney` prefix.

[9] http://vocab.ciudadesabiertas.es/def/transporte/trafico

[10] http://vocab.linkeddata.es/datosabiertos/def/urbanismo-infraestructuras/direccion-postal

[11] https://www.w3.org/TR/vocab-ssn/

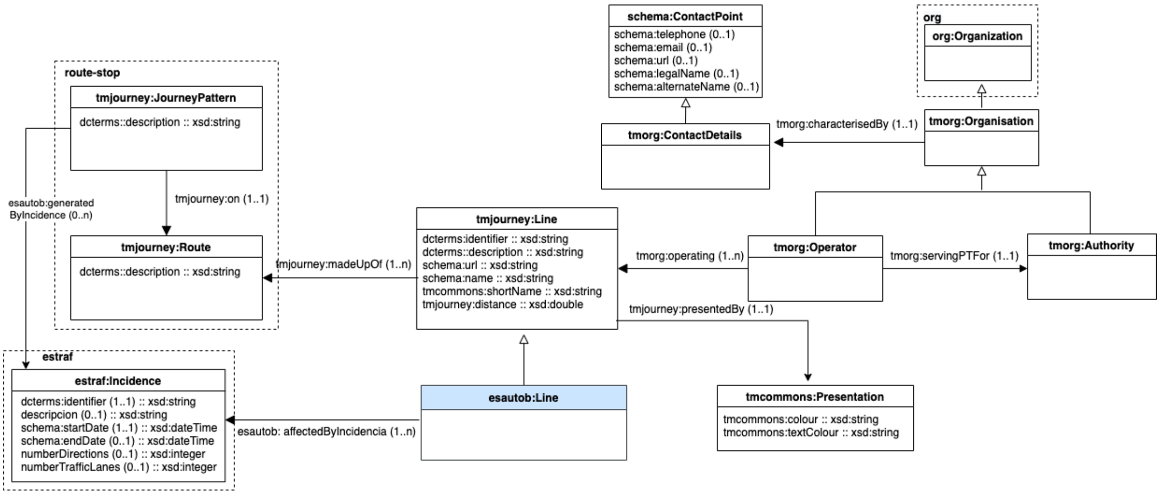

Fig. 6. Public Bus Transport Ontology. Organisation Conceptual Model (Base Model).

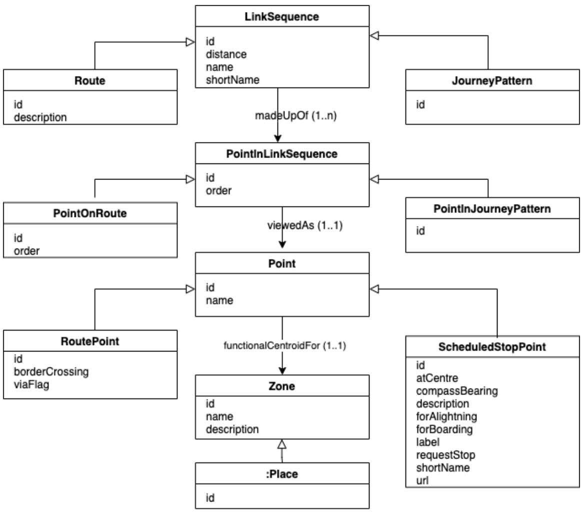

Fig. 7. NOR Transformation of Transmodel UML Network building blocks: points, points in link sequences and link sequences.

Once the reuse activity was completed, we encoded the ontology using the Protegé tool[12]. The next activity was the Evaluation. We used the OOPS! tool to evaluate modelling pitfalls; only minor pitfalls were encountered. Next, we developed a few real-world examples that allowed us to validate if the model is adequate for representing the data in our domain. This was specially important for the concepts of `Route`, `JourneyPattern` and `VehicleJourney`. With these examples we determined the need to simplify the third part on bus vehicle journeys.

Finally, as part of the LOT methodology, oriented-by-data evaluation, RDF data was produced using (CSV to RDF) RML mappings that were generated with the Mapeathor tool [17], a tool that eases mapping rules creation by using a spreadsheet for the specification. Source data was the GTFS feed provided by the Madrid Regional Transport Consortium. Once the mappings were generated, we constructed several knowledge graphs (KG) using the RDFizer tool [18]. Again the mappings and KG were divided in correspondence with the three modules. Examples of the input to Mapeathor, the mappings, and the KG are presented in Figure 11. SPARQL queries that correspond to the competency questions were developed. Queries can be tried out through the GitHub repository[13].

6. Implementation and Alignment with Transmodel

In this section we describe the main challenges encountered in the development of the Public Bus Transport ontology and we give details on the alignment with the Transmodel UML specification. In table 1 we present the general challenges and solutions. Most of the challenges that we encountered are the result of trying to reuse resources that were developed under different perspectives of the domain (Transmodel, NeTEx and SNAP), each with its own documentation, glossary of concepts and implementations. Therefore, when we

[12]https://protege.stanford.edu/

[13]https://github.com/CiudadesAbiertas/vocab-transporte-autobus/blob/master/Examples/queries.md

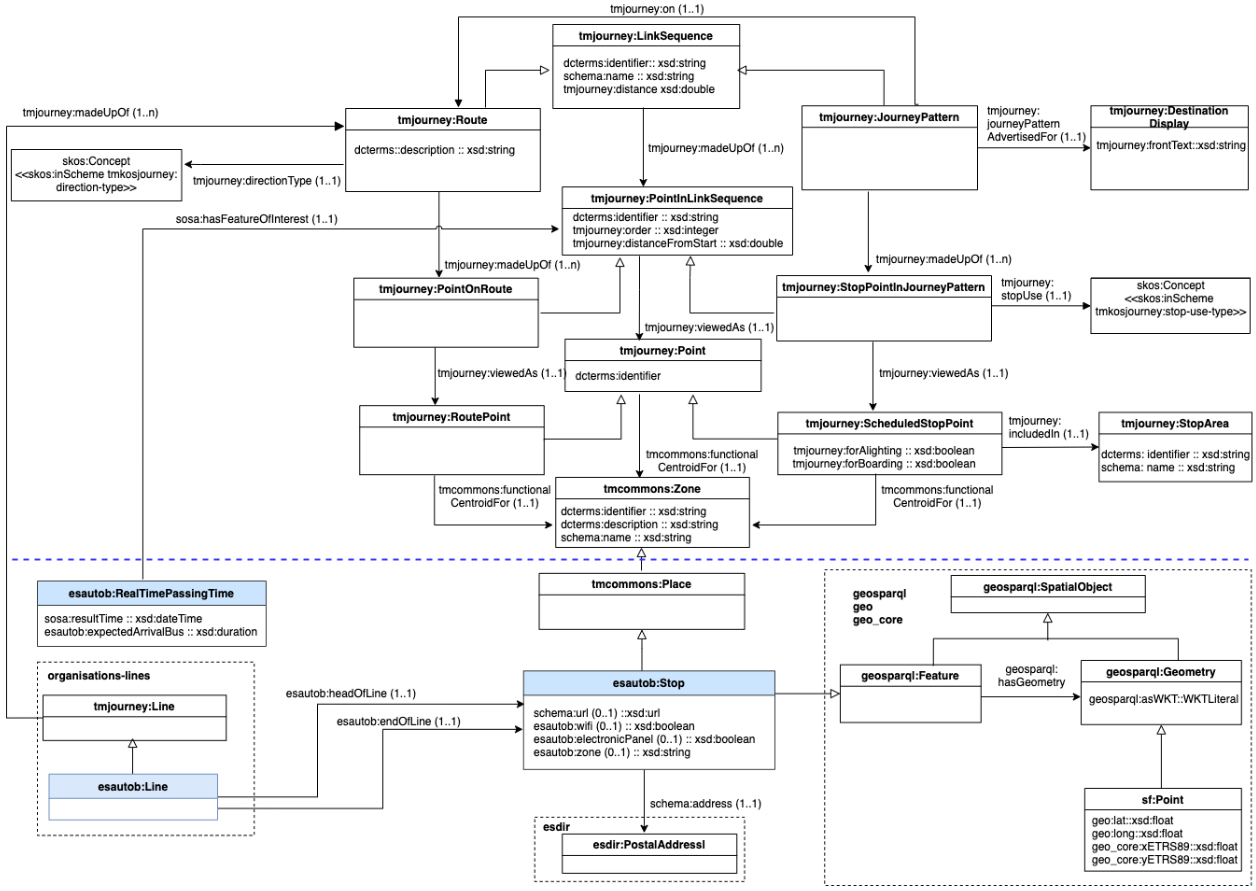

Fig. 8. Public Bus Transport Ontology. Route, Journey Pattern, Stop Conceptual Model.

deal with ontologies that cover specific aspects of this broad domain, we come across with issues like breadth of documentation, over-representation, and ambiguity. Details of the alignment to the UML Transmodel specification follow:

– The UML Public Transport Network Topology package is complex. The hierarchy for the classes `LinkSequence`, `PointInLinkSequence`, `Point`, and its subclasses for a route and journey pattern is scattered across several UML models.
  Solution. A graphical ontological model of these classes using our ontological graphical representation language for a clearer visualisation. Integration into Part II that covers routes and journey patterns. This model was presented in Section 5 under NOR ontology transformation.
– The UML timing-related information is represented in the Common Concepts, Network Topology, and in the Timing Information and Vehicle Scheduling packages. As the relevant concepts are scattered in several packages, the individual UML model graphical representations did not provide clear information and were not used.
  Solution. A graphical ontological model that represents all of the timing-related classes and properties was created, it was validated through real-world examples realizing that not all of the concepts were needed. A reduced ontological model was integrated into Part III that covers vehicle journeys and their schedules.
– There is no clear `Stop` class in the UML specification. Several options exist for representing a `Stop` in Transmodel, e.g. `StopPlace`, `Place`.
  Solution. As the stop in our domain is a physical place with certain data and object properties, a class `esautob:Stop` is defined as a subclass of `Place` that in turn is a subclass of `Zone`.
– There is not a clear match of the Transmodel UML specification with the requirement to rep-

```
# Line 138
<http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/linea/138> a esautob:Line ;
    dcterms:description "Línea 138, comienzo en Cristo Rey y final en San Ignacio de Loyola"^^xsd:string ;
    esautob:startOfLine <http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/parada/4608> ;
    esautob:endOfLine <http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/parada/5481> ;
    tmjourney:madeUpOf <http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/ruta/138a> ;
    tmjourney:madeUpOf <http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/ruta/138b> .
# Route 138a from beginning to end of line
<http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/ruta/138a> a tmjourney:Route ;
    dcterms:description "Ruta de ida de la línea 138 con inicio en Cristo Rey y destino en San Ignacio de Loyola"^^xsd:string ;
    tmjourney:madeUpOf <http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/pointonroute/138a-4608> ;
    ...
    tmjourney:madeUpOf <http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/pointonroute/138a-5481> .
# Point on Route 138a-4608
<http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/pointonroute/138a-4608> a tmjourney:PointOnRoute ;
    tmjourney:order "1"^^xsd:integer ;
    tmjourney:viewedAs <http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/point/4608> .
# JourneyPattern 138a1
<http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/journeypattern/138a1> a tmjourney:JourneyPattern ;
    tmjourney:distance "12,660"^^xsd:double ;
    tmjourney:on <http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/ruta/138a> ;
    tmjourney:madeUpOf <http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/stoppointinjourneypattern/138a1-4608> ;
    tmjourney:madeUpOf http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/stoppointinjourneypattern/138a1-5678 ;
    ...
    tmjourney:madeUpOf <http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/stoppointinjourneypattern/138a1-5481> .
# StopPointInJourneyPattern 138a1-4608
<http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/stoppointinjourneypattern/138a1-4608> a tmjourney:StopPointInJourneyPattern ;
    tmjourney:order "1"^^xsd:integer ;
    tmjourney:viewedAs <http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/point/4608> .
# JourneyPattern 138a2
<http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/journeypattern/138a2> a tmjourney:JourneyPattern ;
    esautob:generatedByIncidence <http://vocab.ciudadesabiertas.es/recurso/transporte/trafico/incidencia/51960FE2-42D7-4B2F-88B0-3774B40E1770> ;
    tmjourney:distance "11,194"^^xsd:double ;
    tmjourney:on <http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/ruta/138a> ;
    tmjourney:madeUpOf <http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/stoppointinjourneypattern/138a2-5678> ;
    ...
    tmjourney:madeUpOf <http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/stoppointinjourneypattern/138a2-5481> .
# Point 4608
<http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/point/4608> a tmjourney:Point ;
    tmcommons:functionalCentroidFor <http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/parada/4608> .
# Stop 4608
<http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/parada/4608> a esautob:Stop ;
    dcterms:description "Cristo Rey"^^xsd:string ;
    schema:address <http://vocab.linkeddata.es/datosabiertos/def/urbanismo-infraestructuras/direccion-postal/cristo-rey> .
```

Fig. 9. Example of Lines, Routes and Journey Patterns. There are two journey patterns for route 138a, the second journey pattern was generated by an incidence and has changed its first stop.

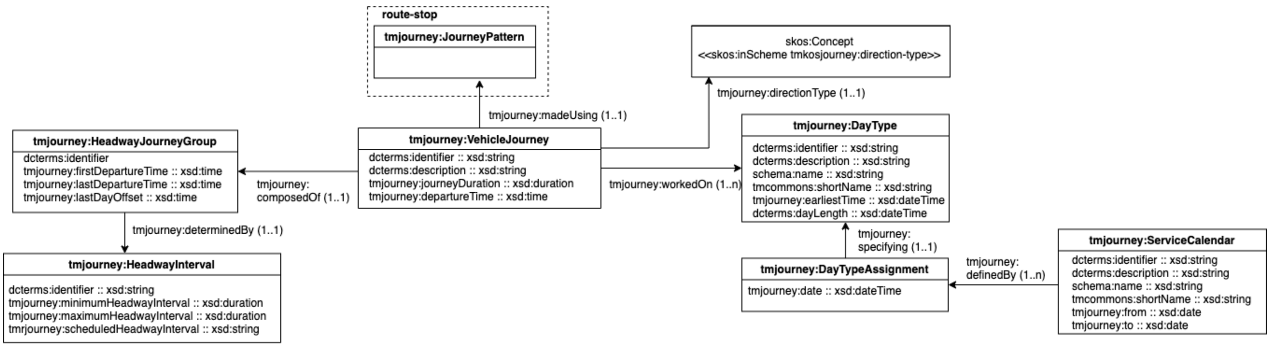

Fig. 10. Public Bus Transport Ontology. Vehicle Journey Conceptual Model.

resent the frequency-based schedule of a journey pattern for a certain type of day. A frequency-based service is represented in the Timing Information and Vehicle Scheduling UML package, specifically in the Frequency Based Service model and it is related to the Vehicle Journey class. However, the representation of an individual vehicle journey is not relevant to our domain.
Solution. In Part III the `VehicleJourney` class is represented and related to the `HeadwayJourneyGroup` that in turn is associated with its `HeadwayInterval`. The RDF data examples and the generated RDF data used for queries, represent one vehicle journey instantiation per journey pattern with its relationships to the frequency-based timetables.

– There is not a clear match of the Transmodel UML specification with the requirement to represent the expected arrival times of buses in a certain stop.
Solution. Reuse of the SOSA ontology. A stop in a route is a `Sensor` where for a given timestamp (`sosa:resultTime`) there is an expected arrival waiting period.
– The Transmodel UML specification defines the `Point` class as a "A 0-dimensional node of the network used for the spatial description of the net-

```
stop_id,stop_code,stop_name,stop_desc,stop_lat,stop_lon,zone_id,stop_url,location_type,parent_station,stop_timezone,wheelchair_boarding
par_6_161,161,Puerta de Alcalá,Plaza de la Independencia 3,40.4206852894155,-3.68919127035982,A,http://www.crtm.es,0,,Europe/Madrid,2
par_6_162,162,Retiro,Avda de Méjico SN,40.4196998317684,-3.68826031065316,A,http://www.crtm.es,0,,Europe/Madrid,2
par_6_164,164,Círculo de Bellas Artes,Calle   Gran Vía 3,40.4189619874668,-3.69714696173351,A,http://www.crtm.es,0,,Europe/Madrid,2
par_6_168,168,Santo Domingo,Calle  Gran Vía 56,40.4215877200394,-3.70761018385723,A,http://www.crtm.es,0,,Europe/Madrid,2
```

(a) Madrid GTFS feed for Stops

ID	Class	URI
idParada	esautob:Parada	http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/parada/{stop_code}
idPoint	sf:Point	http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/punto/p-{stop_code}
idAddress	esdir:DireccionPostal	http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/direccion/a-{stop_code}
idRuta	tmjourney:Route	http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/ruta/{route_id}
idPatronViaje	tmjourney:JourneyPattern	http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/patron-viaje/pv-{route_id}
idHeadsign	tmjourney:DestinationDisplay	http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/headsign/h-{route_id}
idLinea	esautob:Linea	http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/linea/{line_id}
idPointLinkSequence	tmjourney:PointInLinkSequence	http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/punto-en-secuencia/pls-{route_id}-{id_parada}
idPuntoParada	tmjourney:Point	http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/punto-parada/pp-{route_id}-{id_parada}
idRutaParada	tmjourney:Route	http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/ruta/{route_id}
idPatronParada	tmjourney:JourneyPattern	http://vocab.ciudadesabiertas.es/recurso/transporte/autobus/patron-viaje/pv-{route_id}

(b) Input to Mapeathor. Rules for Triples Subjects

Predicate	Object	DataType	ID
schema:name	{trip_short_name}	string	idRuta
tmjourney:on	recurso-autobus:ruta/{route_id}	iri	idPatronViaje
dcterms:description	{stop_desc}	string	idParada
schema:name	{stop_name}	string	idParada
schema:url	{stop_url}	iri	idParada
geosparql:hasGeometry	recurso-autobus:punto/p-{stop_code}	iri	idParada
geo:lat	{stop_lat}	double	idPoint
geo:long	{stop_lon}	double	idPoint
schema:address	recurso-autobus:direccion/a-{stop_code}	iri	idParada
locn:fullAddress	{stop_desc}	string	idAddress
tmjourney:madeUpOf	recurso-autobus:ruta/{route_id}	iri	idLinea

(c) Input to Mapeathor. Rules for Triples Predicates and Objects

Fig. 11. CSV to RDF mappings specified in Mapeathor.

work. The ontologies in the Open Cities project reuse the GeoSPARQL location pattern to represent locations of municipality-related "equipment" e.g., buildings, bus stops. This pattern also defines a `Point` class.
Solution. Represent in the ontology both `Point` concepts: `tmjourney:Point` is needed to represent the relation between points on routes and journey patterns, and physical stop places, and `sf:Point` represents the geographical location of a stop.

7. Conclusions

In this work we have presented an ontology for the representation of data about public buses operating in cities. This ontology is aligned to the the Transmodel reference model. For this development we followed the LOT methodology and adapted the Reuse activity to the scenario of reuse of non-ontological resources, in our case the Transmodel UML specification.

Although the Open Cities project did not require this alignment we considered it as an added value, due to the fact that such an alignment may facilitate the generation of Transmodel-compliant data in the future, as required by the corresponding EU regulation. The ontology development team had also participated in the development of the initial version of the Transmodel ontology in the context of the SNAP project. Therefore, both the complexity of the UML specification and the early state of development of the SNAP ontologies were known beforehand and were an advantage to the development of the alignment.

In order to identify the Transmodel concepts that represented the requirements, we followed a bottom-up approach where we identified the concepts through the Transmodel glossary, and then we built the graphical representation of these concepts by examining portions of the UML models and submodels, and building our conceptualization. This may be a useful experience for other ontology developers in this or other domains who wish to address similar ontology development problems.

Future work includes improving the ontologies originally developed in SNAP with updates to existing concepts from Transmodel, as well as adding other classes and associations from the specification that were not developed in the initial version of those ontologies. Additionally, we suggest dividing the encoding of the Public Bus Transport ontology into the three portions that were conceptualized and presented in this paper.

Table 1
General Challenges to the Development of the Public Bus Transport Ontology

Challenge	**Solution**
Documentation is scattered and there are different versions available	Compilation of a set of documents to be consulted. Transmodel UML V6 2017 packages is our reference documentation.
Transmodel official documentation is work in progress (last version published on September 2019)	Constant review of documentation that required several iterations to make our ontology more consistent
Extensive information on standards vs. lack of information on implementation or examples	Creation of examples from the very beginning to test the implementations
Complex UML Transmodel specification	Generation of a graphical ontological representation of parts of the UML and integration into the ontology
The same concept with different semantics (Transmodel, NeTEx)	Creation of a consistent glossary based on the examples
Concepts that are not represented in the Transmodel UML specification	Definition of new classes and properties. Subclasses of reused classes when appropriate
Complexity of the resulting Part II that covers bus routes and journey patterns	Division of the conceptual diagram in two sections: (1) Transmodel reused concepts and (2) Public Bus Transport ontology concepts
Complexity of modeling the data for API development	Simplification of the API data modeling. Fewer tables were defined by merging the properties of several classes in the ontology

References

[1] FEMP. Federación Española de Municipios y Provincias, DATOS ABIERTOS. Guía estratégica para su puesta en marcha Conjuntos de datos mínimos a publicar, 2017. http://femp.femp.es/files/3580-1617-fichero/Gu%C3%ADa%20Datos%20Abiertos.pdf.

[2] FEMP. Federación Española de Municipios y Provincias, DATOS ABIERTOS FEMP 2019. 40 conjuntos de datos a publicar por las Entidades Locales, 2019. http://femp.femp.es/files/3580-1938-fichero/DATOS%20ABIERTOS%20FEMP%202019.pdf.

[3] M. Poveda-Villalón, A Reuse-Based Lightweight Method for Developing Linked Data Ontologies and Vocabularies, in: *The Semantic Web: Research and Applications*, 2012, pp. 833–837.

[4] D2.2 Detailed Specification of the Semantic Model, Technical Report, Universidad Politécnica de Madrid (UPM), 2017. https://vicinity2020.eu/vicinity/content/d22-detailed-specification-semantic-model.

[5] W3C Linked Data, 2006. https://www.w3.org/DesignIssues/LinkedData.

[6] Commission Delegated Regulation (EU) 2017/1926 of 31 May 2017 supplementing Directive 2010/40/EU of the European Parliament and of the Council with regard to the provision of EU-wide multimodal travel information services, 2017. https://eur-lex.europa.eu/eli/reg_del/2017/1926/oj.

[7] Public Transport Reference Data Model, 2015. http://www.transmodel-cen.eu.

[8] NeTEx, Network Timetable Exchange, 2015. http://netex-cen.eu/.

[9] SIRI, Standard Interface for Real-time Information, 2016. http://www.transmodel-cen.eu/standards/siri/.

[10] SNAP. your transport data into EU compliance, 2019. https://www.snap-project.eu/.

[11] Transmodel Vocabularies, 2019. https://oeg-upm.github.io/snap-docs/.

[12] GTFS Static Overview, 2005. https://developers.google.com/transit/gtfs/.

[13] Transmodel. Common Concepts, Public Transport Network. Timing Information and Vehicle Scheduling, 2017. http://www.transmodel-cen.eu/wp-content/uploads/sites/2/2015/01/TUTORIAL-Part1-3-v0.2.pdf.

[14] M. Scrocca, M. Comerio, A. Carenini and I. Celino, Turning Transport Data into EU Compliance while Enabling a Multimodal Transport Knowledge Graph (forthcoming), in: *International Semantic Web Conference*, Springer, 2020.

[15] M.C. Suárez-Figueroa, A. Gómez-Pérez and M. Fernández-López, *The NeOn Methodology for Ontology Engineering*, in: *Ontology Engineering in a Networked World*, M.C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta and A. Gangemi, eds, 2012, pp. 9–34.

[16] M. Poveda-Villalón, A. Gómez-Pérez and M.C. Suárez-Figueroa, OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation, *International Journal on Semantic Web and Information Systems (IJSWIS)* **10**(2) (2014), 7–34.

[17] A. Iglesias-Molina, D. Chaves-Fraga, F. Priyatna and O. Corcho, Towards the Definition of a Language-Independent Mapping Template for Knowledge Graph Creation, 2019.

[18] E. Iglesias, S. Jozashoori, D. Chaves-Fraga, D. Collarana and M. Vidal, SDM-RDFizer: An RML Interpreter for the Efficient Creation of RDF Knowledge Graphs, in: *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management*, 2020.

[19] Evaluation of the Intelligent Transport Systems (ITS) Directive, 2017. https://ec.europa.eu/transport/themes/its/consultations/2017-evaluation-its-directive_en.

[20] A. García-Silva, A. Gómez-Pérez, M.C. Suárez-Figueroa and B. Villazón-Terrazas, A Pattern Based Approach for Re-engineering Non-Ontological Resources into Ontologies, in: *The Semantic Web*, 2008, pp. 167–181.

[21] M.P. Villalón, A.F. Izquierdo and R.G. Castro, Linked Open Terms (LOT) Methodology (2019). doi:10.5281/zenodo.2539305.

[22] D1.1.2 Updated Version of the Networked Ontology Model, 2007. http://neon-project.org/deliverables/WP1/NeOn_2007_D1.1.2.pdf.