Turning Transport Data to Comply with EU Standards While Enabling a Multimodal Transport Knowledge Graph

Mario Scrocca[(⊠)], Marco Comerio, Alessio Carenini, and Irene Celino

Cefriel – Politecnico of Milano, Viale Sarca 226, 20126 Milan, Italy
{mario.scrocca,marco.comerio,alessio.carenini,irene.celino}@cefriel.com

Abstract. Complying with the EU Regulation on multimodal transportation services requires sharing data on the National Access Points in one of the standards (e.g., NeTEx and SIRI) indicated by the European Commission. These standards are complex and of limited practical adoption. This means that datasets are natively expressed in other formats and require a data translation process for full compliance.

This paper describes the solution to turn the authoritative data of three different transport stakeholders from Italy and Spain into a format compliant with EU standards by means of Semantic Web technologies. Our solution addresses the challenge and also contributes to build a multi-modal transport Knowledge Graph of interlinked and interoperable information that enables intelligent querying and exploration, as well as facilitates the design of added-value services.

Keywords: Transport data · Semantic data conversion · Multimodal transport knowledge graph · Transport EU regulation

1 Introduction

Semantic interoperability in the transportation sector is one of the European Commission challenges: establishing an interoperability framework enables European transport industry players to make their business applications 'interoperate' and provides the travelers with a new seamless travel experience, accessing a complete multi-modal travel offer which connects the first and last mile to long distance journeys exploiting different transport modes (bus, train, etc.).

With the ultimate goal of enabling the provision of multi-modal transportation services, the EU Regulation 2017/1926[1] is requiring transport service providers (i.e., transport authorities, operators and infrastructure managers) to give access to their data in specific data formats (i.e., NeTEx[2] and SIRI[3]) through the establishment of the so-called National Access Points (NAP).

---

[1] EU Reg. 2017/1926, cf. https://eur-lex.europa.eu/eli/reg_del/2017/1926/oj.

[2] NeTEx, cf. http://netex-cen.eu/.

[3] SIRI, cf. http://www.transmodel-cen.eu/standards/siri/.

A survey conducted by the SNAP project[4] revealed that transport service providers have a poor knowledge of the EU Regulation 2017/1926 and its requirements and they do not use or even know the requested standards (NeTEx and SIRI). The transport stakeholders deem the conversion of their data to these standards as technically complex: they would need to dedicate a notable amount of resources to this effort, but often they lack such resources.

We designed and developed an innovative solution for data conversion, based on Semantic Web technologies, hiding the complexity of the conversion and enabling flexibility to address different scenarios and requirements. Our solution, described in this paper, has been validated within the SNAP project to enable the conversion of complex transportation data into EU-mandated standards. The proposed approach fits the needs of transport service providers rendering their legacy data interoperable and compliant with the regulation. Moreover, enabling a multi-modal transport knowledge graph, it also fosters data harmonization for the design and development of added-value travel services.

This paper is organized as follows: Sect. 2 presents our vision on how to address the transport data conversion using Semantic Web technologies; Sect. 3 describes our solution; Sect. 4 provides technical and business evaluations; Sect. 5 draws conclusions and ongoing works.

2   Challenges and Vision

Complying with the EU Regulation on multi-modal transportation services requires sharing data on the National Access Points in one of the standards indicated by the European Commission. This means that each affected organization – transport authority, transport operator and infrastructure manager – has to produce data in such formats. The goal of the European approach is clear and praiseworthy: limit the heterogeneity of data formats, require specific levels of quality and expressivity of data, and pave the way for interoperability.

As mentioned above, the mandated standards are complex and of limited practical adoption. NeTEx is a very broad and articulated XML schema and it was created as an interchange format rather than an operational format. Moreover, very few production system adopt it, so converting to NeTEx requires a deep understanding of its intricacies. NeTEx contains 372 XSD files with 2898 complex types, 7134 elements and 1377 attributes (without considering the national profiles that bring additional country-specific content).

This means that today existing datasets are natively expressed in other formats and require a data translation process for full compliance. Point-to-point conversion is of course possible, but it requires a complete knowledge of the target formats and the correspondences between the original schemata and the target standards. Our approach hides this complexity from the stakeholders, letting them keep using their current legacy systems.
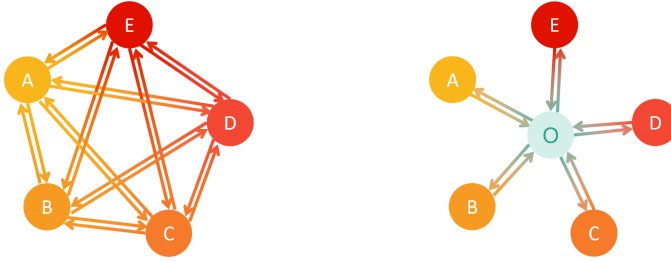

Fig. 1. Mappings required without and with a global conceptual model

The transportation domain is characterised by a proliferation of formats and standards which do not facilitate system interoperability. However, there are efforts that try to overcome the incompatibility challenge, such as Transmodel[5], which aims at representing a global conceptual model for this domain. Indeed, Transmodel is the rich and comprehensive reference schema (its data dictionary only contains 1068 concepts) on which other specific formats are based, such as NeTEx and SIRI themselves.

This is clearly an opportunity for Semantic Web technologies and for the adoption of an any-to-one centralized mapping approach [17], i.e., a global conceptual model (i.e. an ontological version of Transmodel) allowing for bilateral mappings between specific formats and the reference ontology [4].

This approach provides a twofold advantage. On the one hand, it reduces the number of mappings required for conversion and compliance, making the management of complexity manageable: if there are n different formats, the number of mappings is $2n$ instead of $2n(n-1)$ (cf. Fig. 1).

On the other hand, the two-step conversion process (i.e., lifting from the original format to the reference ontology and then lowering from the ontological version to the target format) has a positive "collateral effect": once lifted to the semantic conceptual level, data contributes to build a multi-modal transport knowledge graph of interlinked and interoperable information (cf. Fig. 2).

Taking the example of static transport information (e.g., timetables of public transportation), the target format required by the EU Regulation is the already mentioned NeTEx. Even when all data from different transport stakeholders are converted to NeTEx and shared on a National Access Point, still the challenges of data integration and reuse are not easily solved. However, the availability of a Semantic Web-enabled knowledge representation level, with data expressed with respect to a reference ontology, allows for seamless integration through automatic linking, for intelligent querying and exploration, and for the facilitation of added-value service design. Examples of such services are Intelligent Transport Systems – with new features for travelers such as planning of door-to-door journeys through the usage of multiple transportation modes – or Decision Support

---

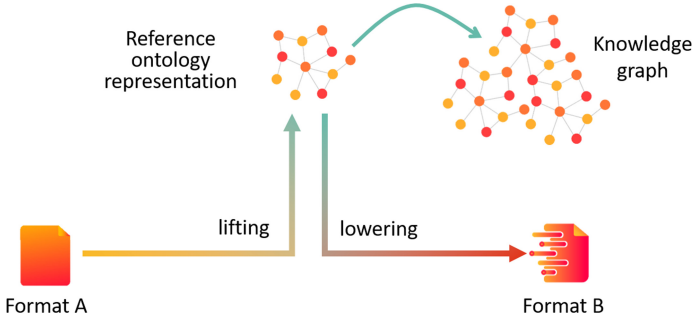[5] Transmodel, cf. http://www.transmodel-cen.eu/.

Fig. 2. Building a knowledge graph as a "collateral effect" of data conversion

Systems for government bodies e.g., for urban planners, providing statistics on mobility usage and supporting the analysis of territory coverage, to comply with the European Green Deal.

Moreover, this multi-modal transport knowledge graph can be implemented with an incremental approach, as suggested by [16], with the possibility to add additional data from multiple providers at different times, still highlighting the advantages and showcasing the opportunities of such a method.

In a nutshell, while addressing the challenge of compliance with the EU Regulation on transport information sharing, we advocate the adoption of a Semantic Web approach that not only solves the issue of data conversion, but also provides a strong and solid solution for data interoperability and for the creation of a new generation of Intelligent Transport Systems, which support the European Commission vision on seamless travel experience for travellers.

3   Our Solution

In this section, we show how we addressed the challenges and implemented the vision of Sect. 2. In particular, we designed a solution that implements the any-to-one centralized approach to semantic interoperability, with a global conceptual model to allow for bilateral mappings between specific formats and the reference ontology. After an overview on similar approaches and related efforts, in this section we explain the requirements, the architecture and the technical choices of Chimera, our open source framework to address the conversion challenge.

3.1   Related Works

Data transformations based on a common data model, such as the ones described in the previous section, require implementing two different processes. The former, which in the context of semantic technologies have been named "lifting", extracts knowledge from the data source(s), and represents it according to the

common data model. The latter, which has been named as "lowering", accesses such information and represents it according to a different data model and serialisation format.

Different approaches have been proposed over the years to deal with data lifting to RDF. Most of them rely on code or format-specific transformation languages (such as XSLT applied to XML documents). Declarative languages have nonetheless been specified to deal with specific data sources, such as relational databases (R2RML [6]), XML documents (xSPARQL [2]) or tabular/CSV data (Tarql [5]). Other languages instead allow creating data transformations that extract information from multiple data sources, such as RML [9] (a generalisation of the R2RML language) and SPARQL-Generate [15]. All the cited approaches and languages support the representation of the source information according to a different structure, and the modification of values of specific attributes to adapt them to the desired data model.

Data lifting using semantic technologies has been also streamlined inside different semantic-based ETL ("Extract, Transform and Load") tools. Unified-Views [14] and LinkedPipes [13] have been implemented during the years, providing environments fully based on Semantic Web principles to feed and curate RDF knowledge bases. A different approach is used by Talend4SW[6], whose aim is to complement an already existing tool (Talend) with the components required to interact with RDF data. An ETL process is based not only on transformation steps. Other components, such as message filtering or routing, are usually involved. The main categorisation of the components and techniques that can be used in an integration process is the Enterprise Integration Patterns [12], which influenced heavily our work (cf. Sect. 3.3).

At the opposite side of the transformation process, lowering has received less attention from the academic community, and there is not a generic declarative language to lower RDF to any format. xSPARQL [2] provides a lowering solution specifically for XML, letting developers embed SPARQL inside XQuery. A downlift approach was proposed in [8], by querying lifting mappings to recreate a target CSV format. In our work, we adopt a declarative approach to implement lowering to XML, relying on SPARQL for querying data and on template engines to efficiently serialise data according to the target format.

Addressing both lifting and lowering is also possible by adopting an object-relational mapping approach (ORM), using and object-oriented representation of both the ontological and non-ontological resources through annotations and mashalling and unmashalling libraries. This approach is implemented in RDF-Beans[7] and Empire[8] and we also applied it in the past [3]. However, its main drawback is memory consumption, making it unsuitable for large data transformation; this is why we turned to the approach described in this paper.

The specific problem of producing datasets complying to the EU Regulation 2017/1926, converting GTFS datasets into NeTEx, is currently being addressed

---

[6] Talend4SW, cf. https://github.com/fbelleau/talend4sw.

[7] RDFBeans, cf. https://github.com/cyberborean/rdfbeans.

[8] Empire, cf. https://github.com/mhgrove/Empire.

by means of a non-semantic approach by Chouette [10]. Even if the implementation does not use Semantic Web languages, the conversion process is implemented in the same way. In Chouette, GTFS data is first loaded inside a relational database strongly based on the Neptune data model, and users can then export data from the database to NeTEx. Such database and related data model perform the same role than RDF and the Transmodel ontology in our approach. Chouette, as well as any direct translation solution, can address and solve the conversion problem; however, it does not build a knowledge graph nor it helps in enabling an easy integration or further exploitation of the converted data.

3.2   Chimera Requirements

The design of our software framework – that we named Chimera – aimed to address the data conversion challenge highlighted in Sect. 2 and, in the meantime, to offer a generic solution for data transformation adoptable in domains different than the transportation one. The requirements that guided our work can be summarized as follows: (i) the solution should address the conversion challenge following the any-to-one centralized mapping approach [17], by employing Semantic Web technologies; (ii) the conversion should support two different data transformation scenarios: batch conversion (i.e., conversion of an entire dataset, typical of static data) and message-to-message mediation (i.e., translation of message exchanges in a service-centric scenario, typical of dynamic data); (iii) the solution should minimise the effort required to the adopters to customize the data conversion process.

3.3   Chimera Architecture

The design of the Chimera architecture follows a modular approach to favour the conversion customization to different scenarios and formats, thus guaranteeing the solution flexibility. The decision of the modular approach is based on the assumption that a data conversion process could be broken down in a pipeline of smaller, composable and reusable elements.

We took inspiration from the already mentioned Enterprise Integration Patterns (EIP) approach, that offers best practices and patterns to break down a data processing flow into a set of building blocks. In this sense, Chimera is similar to the other data transformation solutions mentioned in Sect. 3.1. With respect to EIP terminology, a Chimera converter implements a *Message Translator* system, i.e., a pattern to translate one data format into another; in particular, a Chimera generic pipeline can be seen as a composition of specialized transformers called *Content Enricher*.

The generic pipeline for a data conversion process is illustrated in Fig. 3. The basic conversion includes the already mentioned lifting and lowering elements; however, to support a wider set of scenarios, the Chimera architecture includes additional building blocks.

Therefore, in the design of the Chimera architecture we identified the following typologies of building blocks:
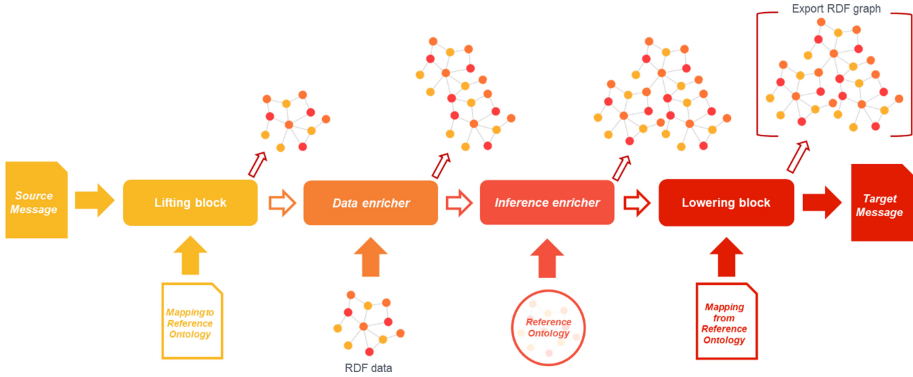
Fig. 3. A data conversion pipeline supported by Chimera.

– **Lifting block** (lifting procedure): this block takes structured data as input and, by means of mappings between the source data model and the reference ontology, transforms it into its ontological representation (an RDF graph);

– **Data enricher block** (merging/linking procedure): this (optional) block takes a set of additional RDF graphs and integrates them within the current graph; this block can be useful when additional data sources, already represented using the reference ontology, are available to enrich the current knowledge graph (i.e., information not present in input messages but needed for the conversion);

– **Inference enricher block** (inference procedure): this (optional) block takes the RDF graph and, based on a set of ontologies (containing axioms or rules), enables an inference procedure to enrich the graph with the inferred information;

– **Lowering block** (lowering procedure): this block queries the RDF graph and, by means of mappings between the reference ontology and the target data format, transforms the data into the desired output format.

In the conversion pipeline, the RDF graph is an intermediate product and it is not a necessary output of the data transformation, thus it could be discarded once the pipeline execution is completed. However, we believe that this "collateral effect" of building an RDF graph is one of the strengths of our approach, because it enables the incremental building of a knowledge graph. In our multi-modal transport scenario, this is of paramount importance, because it allows to build a knowledge graph covering different modes and therefore representing a precious source for multi-modal intelligent transport systems. The Chimera pipelines, therefore, can be configured to save the enriched RDF graph as an additional output of the transformation.

3.4   Chimera Implementation

We implemented Chimera on top of Apache Camel[9], a Java integration framework to integrate various systems consuming or producing data. We chose Camel to inherit its features and advantages: not only it is a completely open-source, configurable and extensible solution, but it also implements best practices and patterns to solve the most common integration problems, including the already-mentioned Enterprise Integration Patterns; finally, being a robust and stable project, Camel supports out-of-the-box several components, runtimes and formats to access and integrate a large set of existing system and environments.

Chimera implements data conversion by exploiting Camel's *Routes* to process data, i.e., custom pipelines with different components to enable a specific transformation. Those Routes can be defined programmatically or, exploiting the integration with Spring[10], they can be simply configured through an XML file. Thanks to the huge set of Camel's pre-defined components, Chimera can achieve both batch conversion and message-to-message mediation, and it can leverage multiple input and output channels: data can be acquired by polling directories or FTP servers, or by exposing REST services, Web APIs and SOAP services, by using different publish-subscribe technologies such as JMS or AMQP.

The Chimera framework implements the specific blocks illustrated in Sect. 3.3 with the help of Semantic Web technologies. On top of Chimera implementation, pipelines can be defined by simply providing a configuration file, without the need to add any code.

The basic idea of Chimera is to define a pipeline composed of different blocks and to "attach" an RDF graph to the specific "message" going through the Camel route. In this way, each block can process the incoming data by relying on a shared global RDF graph. All Chimera components use the RDF4J library[11] to process and handle RDF data and to operate on a Repository interface that can be in-memory, local or a proxy to a remote triplestore.

In the current release of Chimera, we provide the following blocks:

- **RMLProcessor**: a *lifting block* based on RML [9], exploiting our fork[12] of the rml-mapper library[13]; our modified version of the mapper extends RML to declare a set of InputStreams (i.e. a generic source stream rather than individual files[14]) as logical sources in the RML mapping file.

– **TemplateProcessor**: a *lowering block* based on Apache Velocity[15] to implement a template-based solution to query the RDF graph, process the result set and generate the output data in the desired format[16].
– **DataEnricherProcessor**: a *data enricher block* to add a set of RDF sources to the pipeline graph.
– **InferenceEnricherProcessor**: an *inference enricher block* loading a set of ontologies to enrich the pipeline graph with inferred information; it exploits the inferencing capabilities and configurations of the repository[17].
– A set of utility processors (**AttachGraph**, **DumpGraph**) to handle the initialization and export of the RDF graph.

Chimera is released as an open source software framework, with an Apache 2.0 license, and is available at https://github.com/cefriel/chimera. An example pipeline showing how to configure a semantic conversion pipeline using the described lifting, lowering and data enricher blocks is also available[18] and can be tested by following the instructions of the README file.

4   Real-World Evaluation of Our Solution

In this section, we describe the activities performed within the SNAP project to assess the proposed approach and the Chimera converter on concrete scenarios and real datasets. In Sect. 4.1, we describe the pilot scenarios, the involved stakeholders and the differences among the use cases. Then, we provide an evaluation of our proposed solution on four main dimensions:

– *Mappings*: in Sect. 4.2 we discuss the mappings' definition procedure, presenting a conceptualization of the different steps required to enable a conversion pipeline, and an assessment considering the pilots scenarios.
– *Flexibility*: in Sect. 4.3 we discuss the flexibility provided by Chimera in composing customized pipelines to address the specific requirements of the different pilots.
– *Performance*: in Sect. 4.4 we present the actual performance results and comment on conversion pipelines' execution using different input datasets.
– *Business Viability*: in Sect. 4.5 we present considerations emerged from pilots on the business viability of the proposed solution.

4.1   Evaluating the SNAP Solution

To evaluate our solution on both a technical and business side, in the context of the SNAP project we identified a set of stakeholders in the transport domain

---

[15] Cf. https://velocity.apache.org/.
[16] We implemented this approach both as a Chimera block and as a standalone tool available at https://github.com/cefriel/rdf-lowerer.
[17] Cf. for instance RDFS inference on in-memory/native RDF4J stores https://rdf4j. org/documentation/programming/repository/#rdf-schema-inferencing.
[18] Cf. https://github.com/cefriel/chimera/tree/master/chimera-example.

affected by the regulation and necessitating to convert their data. The identified pilots allowed us to test both functional and non-functional requirements of our conversion solution and let us generate valuable outputs for the involved stakeholders that not only obtained data compliant with the regulation, but they also received the RDF representation of their data, with respect to the common reference ontology based on Transmodel.

Three different pilots, covering the region of Madrid, the city of Milan, and the city of Genova, were executed. In each region, we engaged with the corresponding stakeholders to gather uses cases involving different actors identified by the EU regulation, i.e., transport operators, transport authorities and infrastructure managers.

In the region of Madrid, our *transport authority* pilot, we transformed into NeTEx the GTFS data sources of both the Consorcio Regional de Transportes de Madrid[19] (Light Railway, Metro Madrid, Regional Railway) and EMT Madrid[20] (the public bus company inside the city of Madrid). This pilot showcases the ability of our solution to deal with *large dataset conversion.*

In Milano, our *infrastructure manager* pilot, we transformed into NeTEx some data sources of SEA[21], the company managing both Linate and Malpensa airports. In particular, we focused on two different datasets: the airport facilities description (e.g., help desks, information points, ticket booths, lifts, stairs, entrances and exit locations, parking, gates) and the airport internal transport data (i.e., shuttle service between terminals). This pilot showcases the ability of our solution to build *custom conversion pipelines* that take multiple input sources into account and that involve *proprietary data formats*; indeed, we converted both datasets into an integrated NeTEx representation.

In Genova, our *transport operator* pilot, we involved AMT[22], the public transport service provider of the metropolitan area of Genoa. This pilot showcases the ability of our solution to take a generic GTFS-to-NeTEx conversion pipeline (similar to the one used for the Madrid pilot) and to customise it by adding an initial *data preparation* step to operate data cleaning and data enrichment of the original GTFS source provided by the transport operator.

4.2   Mappings' Definition

In all the pilots presented above, the expected outcome is data expressed in the NeTEx standard. Since NeTEx is a concrete XML serialization of Transmodel [1] and since Transmodel is a rich UML representation of the transportation domain, we took Transmodel as reference conceptualization. Therefore, together with the OEG group of UPM, we kick-started an effort to produce the ontological version of Transmodel; even if the ontology is far from being complete, the initial modules that allowed us to address our pilots' data conversion are already available[23].

---

[19] CRTM, cf. https://www.crtm.es/.

[20] EMT Madrid, cf. https://www.emtmadrid.es/.

[21] SEA Aeroporti, cf. http://www.seamilano.eu/.

[22] AMT Genova, cf. https://www.amt.genova.it/amt/.

[23] Cf. https://w3id.org/transmodel/.

At the time of writing, the following modules, encompassing 285 classes, 178 object properties and 66 data properties, are available: Organisations (information about the public transport organisations), Fares, Facilities, Journeys (trips and different types of journeys for passengers and vehicles), and Commons. It is important to note that the union of all these modules does not allow representing yet all the information that is expressed in the Transmodel specification, since the current focus was only to enable pilots' data conversion.

All data conversion pipelines share two steps: the *lifting* from the original data format to the Transmodel ontology and the *lowering* from the ontological version to NeTEx. The former requires the definition of lifting mappings in RML, the latter implies the creation of Apache Velocity templates to generate the final output.

To understand the complexity and the difficulties in defining the **lifting mappings**, we schematize the process in terms of its required steps, as follows:

1. *Assessing the input data model*, to identify the main concepts and relations in the input data.
2. *Aligning the identified concepts/relations with the respective ones in the reference ontology*, considering both the explicit information and the data that could be inferred. Some special cases should also be considered, in particular, cases where a one-to-one mapping cannot be identified.
3. *Extending the reference ontology*, in case some concepts or properties of the input data are not present in the reference ontology. In particular, this can be required if: (i) an information in the input format cannot be represented by the ontology but cannot be discarded, or (ii) the RDF representation of a given piece of information in the reference ontology requires the instantiation of intermediate entities and relations that cannot be materialized given a lack of information in the input data format.
4. *Coding the actual lifting mappings*, which implies creating the files that encode all the above, as well as specific configurations, such as custom functions applied in the process (e.g., to change the format for date and time). Considering the RML specification, a human readable text-based representation can be defined using YARRRML [11] then compiled to RML, and the Function Ontology can be used to declare functions in mappings [7] (FNO is very useful for conversion of specific formats, like date-time for example: if the source doesn't use `xsd:dateTime`, a custom conversion function is needed to generate the proper triples).

The definition of lifting mappings from GTFS to Transmodel (Madrid and Genova pilots) required the completion of all the above mentioned steps. The alignment was possible only through a deep study and understanding of the two data models and their complexity, since they are completely different, even if covering overlapping concepts. The need for ontology extension was very limited but the RML mapping coding required the definition of several custom functions to properly access and manipulate the input data.

In other words, this lifting activity was expensive, because it required a lot of manual work; however, the definition of mappings for a widespread format

such as GTFS can be reused with a large set of potential customers, hence this activity cost can be recovered through economy of scale. The definition of lifting mappings for proprietary formats (Milano pilot) required a similar effort, which however cannot be leveraged with additional customers, hence it should be taken into account when valuing such custom data conversion activities.

Similarly to the lifting procedure, we schematize the steps of the **lowering mappings** definition as follows:

1. *Assessing the output data model*, to identify the main concepts and relations expected in the output format.
2. *Aligning with the reference ontology*, to identify how to query the knowledge graph expressed in the reference ontology to get the information required to produce the output format; the querying strategy should take into account both the explicit facts and the information that can be extracted with non trivial queries. Special cases should also be considered, in particular, cases requiring the definition of queries with complex patterns involving multiple entities.
3. *Extending the output format*, in case the knowledge graph contains relevant information that is not foreseen in the output format; of course, this is possible if those extensions are allowed by the output format, otherwise the additional knowledge should be discarded.
4. *Coding the actual lowering mappings*, which implies creating the actual templates that encode all the above to generate the output data. Considering Velocity templates, this coding phase includes: (i) identifying the output data structures to compose the skeleton for the Velocity template, (ii) defining the set of SPARQL queries to extract the relevant data from the knowledge graph, (iii) encoding the logics to iterate on the SPARQL results and properly populate the skeleton according to the output data format.

It is worth noting that the choice of Transmodel as reference conceptualization has a twofold advantage: on the one hand, as already said, Transmodel is a broad model that encompasses a rich spectrum of concepts and relations of the transport domain, and, on the other hand, it is the model on which the NeTEx XML serialization was defined, hence the correspondence between the ontology and the output format is quite large.

Therefore, the definition of lowering mappings between the Transmodel ontology and NeTEx was quite straightforward given the strict relation between the two models and the use of similar terminology. For this reason, the assessment of the output data model and the reference ontology alignment activities were simplified, and the output format extension activity was not necessary. Moreover, since all our pilots have NeTEx as output format, we reused the lowering mappings across the different scenarios.

4.3   Pipeline Composition

The three pilots illustrated in Sect. 4.1 have different requirements, hence they need different data conversion pipelines. In this section, we explain the three

pipelines we built on top of Chimera, to show the flexibility of our approach. In general, setting up a pipeline is a matter of adapting and configuring Chimera blocks to address the specific scenario.

The basic pipeline, implemented for the *Madrid pilot*, executes the conversion from GTFS to NeTEx. This case demonstrates the main advantages of our solution in composing custom conversion pipelines. The first advantage is the possibility to create custom blocks in Chimera pipelines, besides the default ones. In this case, a custom *GTFS Preprocessing* block checks the file encoding (e.g., to handle UTF with BOM files) and generates input streams for the lifting procedure (e.g., creating different *InputStream*s from a single file to overcome a known limitation of the current RML specification, i.e. the impossibility to filter rows in CSV data sources). The second advantage is the possibility of including existing Camel blocks in the pipeline. In this case, the *ZipSplitter* block accesses the different files in the zipped GTFS feed, and other utility blocks deal with input/output management and routing. To design the final pipeline, the *lifting* and *lowering* blocks are configured with the specific mappings, and the additional blocks are integrated to define the intended flow: (i) an *AttachGraph* block to inizialize the connection with the remote RDF repository, (ii) a *ZipSplitter* block, (iii) the custom *GTFS Preprocessing* block, (vi) a *RMLProcessor* lifting block configured using RML mappings from GTFS to the Transmodel Ontology, (v) a *TemplateProcessor* lowering block configured using a Velocity template querying a knowledge graph described by the Transmodel Ontology and producing NeTEx output, and (vi) a *DumpGraph* block to serialize the content of the generated knowledge graph.

The pipeline implemented for SEA in the *Milano pilot* extends the Madrid pipeline showcasing how Chimera allows to easily integrate data in different formats. In this scenario, data on airports' facilities provided in a proprietary data format should be merged with GTFS data describing the shuttle service. To produce a unified pipeline for SEA, we configure Chimera to execute in parallel two lifting procedures: (i) a GTFS-to-Transmodel lifting portion, similar to the Madrid one, to obtain shuttle service data represented in the Transmodel ontology, and (ii) a custom SEA-to-Transmodel lifting portion, to obtain facility data represented in the Transmodel ontology. After the lifting procedures, a *DataEnricher* block is added to the pipeline to merge the RDF triples materialized in the two different lifting procedures. Finally, the Transmodel-to-NeTEx lowering templates are executed to map the shared graph to an integrated NeTEx representation.

The pipeline implemented for AMT in the *Genova pilot* extends the Madrid pipeline by integrating Chimera with functionalities of external systems. In this case, the conversion pipeline needed to interact with an external data preparation component. This external system, based on Fiware[24], enriches an initial GTFS feed with additional data sources and it can be configured to call a REST API whenever a new enriched feed is available. In the implemented pipeline, pre-existing Camel blocks are configured to accept POST calls containing an enriched

---

[24] Fiware, cf. https://www.fiware.org/.

GTFS feed. Each API call triggers the GTFS-to-NeTEx conversion pipeline, as in the other pipelines, and in the end also returns the results to the initial requester.

The pipelines of the three pilots demonstrate how the Chimera modular approach, inherited from Camel, allows to easily define customized conversion pipelines. The basic conversion pipeline can be configured with the default blocks for lifting and lowering. Complex scenarios can be addressed implementing additional blocks or employing already defined Camel blocks. Moreover, the same pipeline can be manipulated to fulfill different requirements with minimal modifications.

4.4   Performance Testing

In this section, we provide some statistics on the actual execution of the Chimera conversion pipelines presented in Sect. 4.3 and we comment on performance results and addressed issues.

We executed the conversion using Docker Containers on a machine running CentOS Linux 7, with Intel Xeon 8-core CPU and 64 GB Memory. Memory constraint is set to 16 GB using the Docker `--memory-limit` option on running containers, no limits are set on CPU usage. GraphDB 9.0.0 Free by Ontotext[25] was used as remote RDF repository by Chimera pipelines for storing the materialized knowledge graph and querying triples during the lowering phase.

In Table 1, we report the execution results. For each conversion case, we detail: the dataset size (for GTFS feeds, the total number of rows in all CSV files); the number of triples in the materialized knowledge graph; the lifting, lowering and total execution times; the NeTEx output dataset size.

The numbers show that NeTEx is much more verbose than GTFS and, since Transmodel has similar terminology, the knowledge graph and the output dataset are much bigger than the input data.

The conversion times showcase the ability of our solution to handle large knowledge graphs. Considering the size of CRTM, AMT and EMT datasets, we notice how conversion time grows almost linearly with respect to the input size. The biggest dataset conversion required one hour, which is reasonable for batch processing of static transport data that changes sporadically.

To make our solution efficient, we implemented in Chimera a set of optimizations as follows. The rml-mapper library, used by the Chimera lifting block, stores all triples generated in an in-memory object before serializing them at the end of the procedure. To reduce memory consumption, we implemented incremental upload of triples generated to a remote repository during the materialization. This modification reduced the memory consumption, but shifted the performance bottleneck to the triplestore. For this reason, we allowed to set in the pipeline the number of triples in each incremental *insert* query, we created a queue collecting pending queries and we spawn a thread pool of configurable size to consume the queue. Since the Free version of GraphDB allows for only two

---

Table 1. Conversion execution statistics; * indicates facilities in Malpensa and Linate airports, only a subset of this dataset is converted to NeTEx.

		CRTM	EMT	SEA	AMT
GTFS total no. rows		168,759	2,381,275	106	710,104
GTFS size	MB	12.46	100	0.005	45
Other data size	MB	-	-	1.62*	-
Lifting time	sec	121	1,697	6	546
Lowering time	sec	98	1,823	5	624
Conversion time	min:sec	3:39	58:40	00:11	19:30
No. triples		1,692,802	27,981,607	7,597	9,361,051
NeTEx size	MB	182.3	2,450	0.728	502

concurrent queries on a single core, the execution time is affected and directly influenced by the performances of the single core, slowing down the queue consumption.

We also experienced that a clear performance improvement in the RML materialization comes from a reduction of the number of *join* conditions in the RML mappings. In RML, triple generation is often achieved with joins between different data sources; this powerful RML construct, however, is computationally expensive, especially in case of large datasets and nested structures. In our tests, join conditions exponentially increase the required conversion time. Therefore, in our mappings we opted to limit the use of join conditions and instead use separate and simpler Triple Maps with the same IRI generation pattern, to achieve the same result without heavily affecting the conversion time.

Considering the lowering procedure, since a standard lowering approach has not emerged yet, we adopted a generic templating solution based on Apache Velocity, which is flexible to adapt to any target format. Any SPARQL SELECT query supported by the triplestore (that returns a table) can be included in the template; the processing of the SPARQL result table can be as complex as needed (using the Velocity Template Language). In the described scenario of Transmodel-to-NeTEx lowering, we implemented a set of generic templates, each with SPARQL queries with multiple *optional* patterns, to avoid assumptions on the available data. Of course, writing templates with focused queries (e.g. avoiding *optional* clauses) would improve the processing time, because the lowering performance is directly related to the query performance.

To further reduce the execution time, we generated supporting data structures to access data more efficiently, with queries avoiding nested loops. To reduce memory consumption, we removed white spaces and newlines in the Apache Velocity template, negatively impacting the output readability, which can however be improved at a later stage.

4.5   Business Viability

Finally, we add some considerations regarding the business evaluation of the presented work.

Initially, the goal of our Chimera framework was limited to provide a *data conversion solution*, to ensure the compliance with the mentioned EU Regulation. This objective was fully achieved, and all the involved stakeholders – representing transport authorities, transport operators and infrastructure managers – expressed their full satisfaction with the results obtained with our solution. We run a final meeting with each of them to illustrate the approach, the results, the pricing model and to ask for feedback. They all received their input data enriched, integrated and transformed into the required NeTEx format, with no effort on their side. Regarding the conversion effectiveness, they all agreed that we had successfully provided a solution to the problem of compliance to the European regulation and that they will adopt it, once the respective National Access Point will be setup, requiring them to supply their data in the NeTEx format.

With respect to the conversion issue, therefore, we showed the feasibility and viability of our solution. Furthermore, to preserve our business proposition, we keep the lifting and lowering mappings together with the pipeline configuration as our unique selling point, while we decided to release the Chimera framework as open source, as already mentioned, to facilitate its adoption and improvement by the larger community.

It is worth noting that developing a conversion solution on top of Chimera still requires all the technical skills to manually develop RML mappings (for the lifting step) and SPARQL queries to fill in the templates (for the lowering step), even when the full domain knowledge is available. This means that there is room for improvement in terms of supporting tooling by our community. At the moment, in business terms, this represents a competitive advantage for us and for Semantic Web-based solution providers.

While addressing the data conversion issue, however, we discovered that we can indeed offer a broader set of services to the target customers in the transport domain. As a result of our business modeling effort, we defined three offers:

- *Requirements analysis and compliance assessment*, to investigate the feasibility and complexity of converting the customer data in the desired format;
- *Conversion services*, including data preparation and enrichment, lifting to a reference ontology, data integration and lowering to the desired format;
- *Supporting services*, including long-term maintenance (e.g., repeated conversion, incremental addition of data sources), knowledge transfer and training.

In terms of pricing model, on the basis of the described experience, we now opt for different pricing depending on the *estimated difficulty of the conversion*, which we measure along three dimensions of the source data: (i) popularity of the data format (widespread or proprietary, because this impact the re-usability of the mappings), (ii) data volume, and (iii) input format complexity (again, because it impacts the mapping definition).

As a concluding remark, we highlight that the Madrid transport authority CRTM – which is already adopting Semantic Web solutions and managing RDF data – gave us a very positive feedback with respect to our solution's "collateral effect" of generating a nucleus of a *multimodal transport knowledge graph*: they perfectly understood and appreciated the enabled possibility to build additional added value services on top of the knowledge graph itself, well beyond the pure data conversion for EU Regulation compliance. Indeed, once National Access Points will be in place and transport stakeholder will provide their data, the actual exploitation of the multimodal transport knowledge graph will be enabled, e.g. by developing intelligent journey planning offering solutions based on the usage of different transport modes.

The availability of an integrated and harmonized knowledge graph can also pave the way for assessing the conversion completeness. However, in most of the pilot cases illustrated in this paper, the input data sources are in GTFS (a very simple and basic tabular data format), the target standard is NeTEx (a very complex and articulated XML format), and the Transmodel reference ontology is very close to NeTEx. Thus the coverage of the target standard (and the reference ontology) is always quite limited even if the entire information contained in the source data is completely used in the mapping. Therefore we leave this kind of analysis as future work to support and improve the mapping development (e.g. using shape validation to assess the quality/completeness of the mapping results with respect to the reference ontology).

5   Conclusions and Future Works

In this paper, we presented our solution to enable the conversion of transport data, into standards required by the European Commission, using Semantic Web technologies. To support the implementation, we designed the Chimera framework, providing a modular solution to build semantic conversion pipeline configuring a set of pre-defined blocks. The described solution has been employed within the SNAP project on concrete scenarios and real datasets involving transport stakeholders in Italy and Spain. The different pilots have been presented proposing an evaluation of the solution on different dimensions. The performed activities acknowledge the feasibility of the solution on a technological side, the desirability for stakeholders and the business viability of the approach.

We are also adopting Chimera in the ongoing SPRINT project[26] with different reference ontologies and source/target standards, to implement and evaluate conversion pipelines in both cases of batch transformation and message translation; we have already reached a clear scalability improvement with respect to our previous ORM-based solution [3] and we are proving the generalizability of our technological solution beyond the scenario offered in this paper.

As future works, we plan to: (i) explore additional tools to facilitate the mappings' definition (e.g. collaborative and visual tools), (ii) implement additional blocks for Chimera to offer more options in the pipeline definition (e.g.,

---

[26] Cf. http://sprint-transport.eu/.

approaches based on a virtualized graph to extract data from the original sources) and, (iii) perform an evaluation of the proposed solution considering dynamic data (e.g., DATEX II and SIRI formats) and requirements in real-time scenarios.

References

1. Arneodo, F.: Public transport network timetable exchange (NeTEx) - introduction. Technical report, CEN TC278 (2015). http://netex-cen.eu/wp-content/uploads/2015/12/01.NeTEx-Introduction-WhitePaper_1.03.pdf
2. Bischof, S., Decker, S., Krennwallner, T., Lopes, N., Polleres, A.: Mapping between RDF and XML with XSPARQL. J. Data Semant. **1**(3), 147–185 (2012)
3. Carenini, A., Dell'Arciprete, U., Gogos, S., Pourhashem Kallehbasti, M.M., Rossi, M.G., Santoro, R.: ST4RT - semantic transformations for rail transportation. Transp. Res. Arena TRA **2018**, 1–10 (2018). https://doi.org/10.5281/zenodo.1440984
4. Comerio, M., Carenini, A., Scrocca, M., Celino, I.: Turn transportation data into EU compliance through semantic web-based solutions. In: Proceedings of the 1st International Workshop on Semantics for Transport, Semantics (2019)
5. Cyganiak, R.: TARQL (SPARQL for tables): Turn CSV into RDF using SPARQL syntax. Technical report (2015). http://tarql.github.io
6. Das, S., Sundara, S., Cyganiak, R.: R2RML: RDB to RDF Mapping Language. W3C recommendation, W3C, September 2012. https://www.w3.org/TR/r2rml/
7. De Meester, B., Dimou, A., Verborgh, R., Mannens, E.: An ontology to semantically declare and describe functions. In: Sack, H., Rizzo, G., Steinmetz, N., Mladenić, D., Auer, S., Lange, C. (eds.) ESWC 2016. LNCS, vol. 9989, pp. 46–49. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47602-5_10
8. Debruyne, C., McGlinn, K., McNerney, L., O'Sullivan, D.: A lightweight approach to explore, enrich and use data with a geospatial dimension with semantic web technologies. In: Proceedings of the Fourth International ACM Workshop on Managing and Mining Enriched Geo-Spatial Data, pp. 1–6 (2017)
9. Dimou, A., Vander Sande, M., Colpaert, P., Verborgh, R., Mannens, E., Van de Walle, R.: RML: a generic language for integrated RDF mappings of heterogeneous data (2014)
10. Gendre, P., et al.: Chouette: an open source software for PT reference data exchange. In: ITS Europe. Technical Session (2011)
11. Heyvaert, P., De Meester, B., Dimou, A., Verborgh, R.: Declarative rules for linked data generation at your fingertips!. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 11155, pp. 213–217. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98192-5_40
12. Hohpe, G., Woolf, B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley Professional (2004)

13. Klímek, J., Škoda, P., Nečaský, M.: LinkedPipes ETL: evolved linked data preparation. In: Sack, H., Rizzo, G., Steinmetz, N., Mladenić, D., Auer, S., Lange, C. (eds.) ESWC 2016. LNCS, vol. 9989, pp. 95–100. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47602-5_20

14. Knap, T., Kukhar, M., Macháč, B., Škoda, P., Tomeš, J., Vojt, J.: UnifiedViews: an ETL framework for sustainable RDF data processing. In: Presutti, V., Blomqvist, E., Troncy, R., Sack, H., Papadakis, I., Tordai, A. (eds.) ESWC 2014. LNCS, vol. 8798, pp. 379–383. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11955-7_52

15. Lefrançois, M., Zimmermann, A., Bakerally, N.: Flexible RDF generation from RDF and heterogeneous data sources with SPARQL-generate. In: Ciancarini, P., et al. (eds.) EKAW 2016. LNCS (LNAI), vol. 10180, pp. 131–135. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58694-6_16

16. Sequeda, J.F., Briggs, W.J., Miranker, D.P., Heideman, W.P.: A pay-as-you-go methodology to design and build enterprise knowledge graphs from relational databases. In: Ghidini, C., et al. (eds.) ISWC 2019. LNCS, vol. 11779, pp. 526–545. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30796-7_32

17. Vetere, G., Lenzerini, M.: Models for semantic interoperability in service-oriented architectures. IBM Syst. J. **44**(4), 887–903 (2005)