Intelligent Urban Traffic Management via Semantic Interoperability Across Multiple Heterogeneous Mobility Data Sources

Mario Scrocca[1](✉), Marco Grassi[1], Marco Comerio[1], Valentina Anita Carriero[1], Tiago Delgado Dias[2], Ana Vieira Da Silva[2], and Irene Celino[1]

[1] Cefriel – Politecnico di Milano, Viale Sarca 226, 20126 Milan, Italy
{mario.scrocca,marco.grassi,marco.comerio,valentina.carriero,irene.celino}@cefriel.com

[2] A-to-Be Mobility Technology, S.A. Edificio Brisa, 2785-599 São Domingos de Rana, Portugal
{tiago.delgado.dias,ana.vieira.silva}@a-to-be.com

Abstract. The integrated exploitation of data sources in the mobility domain is key to providing added-value services to passengers, transport companies and authorities. Indeed, multiple stakeholders operate and maintain different kinds of data but several interoperability issues limit their effective usage. In this paper, we present an architecture enabled by Semantic Web technologies to overcome such issues and facilitate the development of an integrated solution for mobility stakeholders. The proposed solution is composed of different components that address challenges for enabling data interoperability, from the findability of data sources to their integrated consumption adopting standardised data formats. We report on its implementation and validation in four European cities to enable data-driven tools for the dynamic management of multi-modal traffic. Finally, we discuss the feedback received by users testing the solution and the lessons learnt during its development.

Keywords: Mobility · Semantic interoperability · Data Integration

1 Introduction

Interoperability is one of the main challenges in enabling collaboration between travel and transport industry players. The interoperability of data and services is essential for creating an ecosystem of transport stakeholders, enabling the definition of new data-driven solutions. The benefits range from enhancing the operations of transport companies to providing integrated and seamless mobility services to users. The TANGENT project[1], co-funded by the European Commission under the Horizon 2020 Programme, is developing new complementary tools

[1] Enhanced Data Processing Techniques for Dynamic Management of Multimodal Traffic (TANGENT), https://doi.org/10.3030/955273.

G. Demartini et al. (Eds.): ISWC 2024, LNCS 15233, pp. 218–235, 2025.
https://doi.org/10.1007/978-3-031-77847-6_12

for optimising traffic operations in a coordinated and dynamic way from a multimodal perspective. In this paper, we discuss how knowledge graphs and semantic technologies can help in tackling interoperability in the mobility domain.

The development and testing of the data-driven solutions developed by TANGENT in the Athens, Lisbon, Greater Manchester, and Rennes Metropole case studies asked for a solution to different data interoperability challenges. The definition of a proper solution for data sharing and usage is not straightforward due to several issues to be addressed: datasets in different formats and/or using different data models, data services relying on different specifications and technologies, and metadata describing data sources according to different profiles. This paper describes the design and implementation of an integrated set of tools that employs Semantic Web technologies to address data interoperability issues for heterogeneous data sources from different stakeholders.

While the proposed solution can be applied to a generic domain, we describe how we implemented and tested it considering the specificities of multimodal transportation. Furthermore, we describe the reusable resources, like metadata specification and ontologies, that are made publicly available.

Within the integrated TANGENT solution, data interoperability is leveraged to support the deployment of an innovative dynamic traffic management platform. The platform provides intelligent services and cutting-edge user interfaces and is enabled in each city by the integrated consumption of several data sources. The data sources were retrieved and harmonised involving different stakeholders and systems, thus demonstrating the flexibility and scalability of the proposed solution. We received positive feedback from users during testing sessions that acknowledged the benefits of integrating data for different transportation modes within a single solution and expressed their interest in adopting the solution within their operations.

The remainder of the paper is organised as follows. Section 2 discusses the motivating challenges and the related work. Section 3 describes the proposed solution and its implementation considering the heterogeneous mobility data sources of the four cities involved in the validation. Section 4 discusses the evaluation and lessons learned. Finally, Sect. 5 draws the conclusions.

2 Challenges and Related Work

Data interoperability is a challenging objective to enable different stakeholders to communicate and exchange information effectively without losing meaning. Indeed, stakeholders adopt different (legacy) systems for data management and exchange that cannot be directly integrated or harmonised. To better understand the problem within the considered domain, we first investigated the existing products and services adopted by the case studies involved in TANGENT. The current landscape is characterised by: (i) open data portals managed at different levels (regional, national, European) that contain datasets and data services not well-documented (i.e., bad quality metadata), provided in not interoperable data formats (e.g., custom CSVs) and often not updated because not directly used by the relevant stakeholders; (ii) solutions from third-party vendors (usually associated with the vendor of the sensors generating the data) that deal with specific sets of data (e.g., road traffic data), use custom data formats and do not provide easy access (e.g., API) to the raw data. Both of these options restrict the ability to find relevant data sources, especially considering non-public data, and to access pertinent information for comprehensive traffic management.

Five major challenges can be identified and should be addressed [5]: 1. **Locate** (*which data is available and where?*), 2. **Access** (*how to obtain the needed data?*), 3. **Harmonise** (*how to convert data according to the required data model?*), 4. **Integrate** (*how to ensure different data sources can be merged?*), 5. **Extract** (*how to consume harmonized and integrated data?*).

Each of these challenges is associated with several issues and identifying a single solution is impossible since a single interoperability problem cannot be formulated. Indeed, data interoperability scenarios are widely heterogeneous and pose various requirements [24] that can be possibly faced only by considering a set of tools appropriately configured. To select such tools and define an integrated solution [4], we reviewed state-of-the-art data interoperability solutions based on Semantic Web technologies and their application for the mobility domain.

2.1 Locate and Access

The first challenge is the findability and discoverability of data. Data cannot be re-used and (made) interoperable if they cannot be found. For this reason, data catalogues/portals are implemented to describe data sources through a set of metadata. The challenge is associated with the need for a proper, structured and machine-readable description of data sources that could also support interoperability across different data catalogues. Once data sources are located, the second challenge is related to data accessibility. Data catalogues adopt different strategies for data access mainly associated with the architectural choices for the hosting and storage of static and dynamic data sources. The challenge is to enable uniform access to heterogeneous data sources for end users.

The *locate* and *access* challenges are also being addressed by the European Commission through National Access Points (NAP) for mobility data. Each Member State should operate a NAP to enable the sharing of mobility data by transport stakeholders as mandated by dedicated Delegated Regulations [7–9] supplementing the Intelligent Transport Systems (ITS) Directive 2010/40/EU [21]. The concept of NAP leverages the one of Data Catalogue, i.e., a digital platform to facilitate the sharing of data sources and their findability by other stakeholders. However, several mobility data platforms exist but are not interoperable. Even in the case of NAPs, each Member State adopted different approaches for their implementation, thus creating interoperability issues at the European level. For this reason, the NAPCORE[2] project is currently working on coordinating and harmonizing such platforms around Europe. One important objective is supporting the findability of data contained in each mobility data

[2] https://napcore.eu/.

platform [25] and defining mobilityDCAT-AP[3], a uniform metadata specification to access the data sources. The adoption of structured metadata descriptors according to well-known vocabularies, e.g., the Data Catalog Vocabulary (DCAT) [1] and the corresponding DCAT Application profile (DCAT-AP) for data portals in Europe [29], is fundamental to facilitate search within one or multiple data catalogues. Moreover, proper data governance must be defined to regulate the usage of the catalogue between the different involved stakeholders. Finally, data catalogues should support the harmonisation of technological access to data sources.

For these reasons, we identified the two core components of the proposed solution as a shared *Data Catalogue* to enable the findability of data sources and a uniform *Data API* for accessibility.

2.2 Harmonise, Integrate, and Extract

The remaining three data interoperability challenges (harmonise, integrate, and extract) are related to the processing of (meta)data to enable their integration and exploitation according to common semantics. A flexible solution is required to address heterogeneous requirements in terms of: (a) **schema and data transformation**: information manipulation to obtain syntactic (structural) and semantic interoperability of (meta)data; (b) **integration** with existing information systems as data sources (i.e., components generating or storing the data) and/or data sinks (i.e., components consuming or archiving the data).

Different approaches can be exploited and implemented, spanning from ad-hoc solutions targeting a specific scenario to more general and scalable solutions supporting multiple stakeholders and data representations. The semantic any-to-one mapping approach based on [30] and validated in [27] reduces the number of mappings, i.e., translations from one representation to another, that are needed to implement interoperability by different stakeholders. Such an approach is based on the identification of a reference model for the domain of interest. Each stakeholder is responsible for defining mappings from their own data representation to the reference model (*lifting*) and vice versa (*lowering*). In this paper, we discuss how we adopted this approach.

Considering the mobility domain, different reference models are proposed based on existing standards. Chouette [12] and the SNAP solution [23] rely on a reference model based on Transmodel[4] for the conversion of Public Transport (PT) data in different formats. The `transit_model` tool[5] adopts the Navitia Transit Feed Specification (NTFS)[6] to manage, convert and enrich transit data from/to different formats. Moreover, considering traffic data, the Datex II[7] specification is often used as a reference model to convert custom data formats and share harmonised data [14].

[3] https://w3id.org/mobilitydcat-ap.
[4] https://www.transmodel-cen.eu/.
[5] https://github.com/hove-io/transit_model.
[6] https://github.com/hove-io/ntfs-specification.
[7] https://www.datex2.eu/.

Different approaches for lifting and lowering can be suitable considering a specific scenario. Moreover, the harmonisation, integration and extraction process may require the definition of custom pre- and post-processing, considering different interoperability issues. Therefore, composing and configuring different components should be possible considering the specific requirements for integrating certain data sources.

Different approaches for lifting and lowering can be suitable based on a specific scenario. Moreover, the harmonisation, integration, and extraction process may require the definition of custom pre- and post-processing, taking into account different interoperability issues. Therefore, it should be possible to compose and configure different components while considering the specific requirements for integrating certain data sources.

Different semantic-based ETL (Extract, Transform and Load) tools have been proposed to define composable procedures with Semantic Web technologies [13]. Technologies for declarative knowledge graph construction [28] can effectively support lifting transformations, while a standardised lowering solution to convert RDF to any format using a generic declarative language is currently missing [26]. Moreover, other components, such as message filtering or routing, are usually required within a transformation pipeline. Enterprise Integration Patterns [16] offer a relevant categorisation of the components and techniques for system integration.

In conclusion, two additional components are identified to support the solution: a *Reference Conceptual Model* defining common semantics and the composition and configuration of *Semantic Harmonisation and Fusion Pipelines*.

3 TANGENT Solution for Dynamic and Intelligent Multimodal Traffic Management

This section describes the TANGENT solution for dynamic and intelligent multimodal traffic management. We discuss each macro-component of the proposed architecture and then its integration within the overall TANGENT solution [19] as shown in Fig. 1. The main original contributions of this work are the implementation and integration of different technologies to propose an holistic architecture for data interoperability based on Semantic Web technologies and its customisation for the multimodal traffic management domain. In the following we describe them, highlighting their value in solving the discussed challenges and their impact on the business scenarios.

3.1 Data Catalogue

This component is a catalogue of digital assets available online and accessible by users via a web browser. Different digital assets can be characterized by specifying a metadata descriptor according to a common metadata profile. The data catalogue may also harvest metadata descriptions from existing data portals (e.g., NAPs) [3]. Programmatic access to the list of assets published and

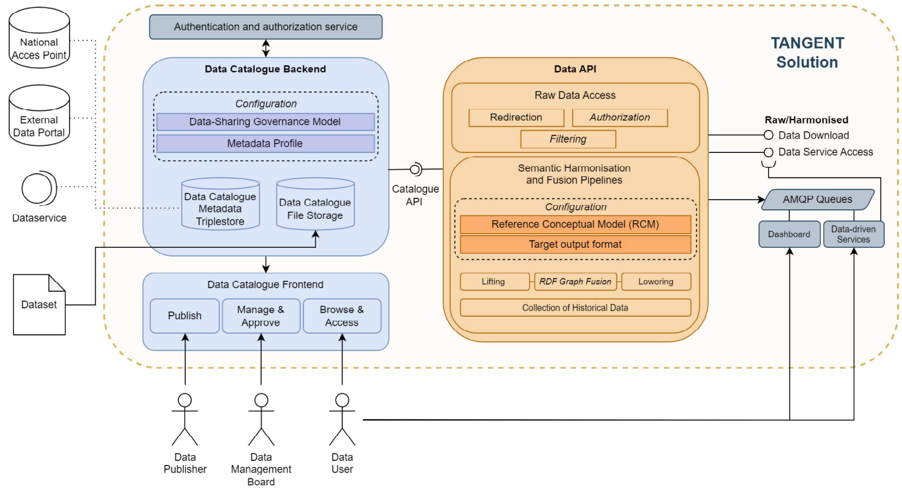

Fig. 1. Overview of the proposed solution for data interoperability

their metadata is implemented via a dedicated API (*Catalogue API*). Metadata serialized in RDF enables advanced functionalities based on querying and/or automated processing by agents of such metadata (e.g., in federation scenarios). Finally, the catalogue enforces processes for the governance of digital assets.

The *Data Catalogue* provides a single location where all the collected data sources for each city are described and can be explored. Its development is based on the Knowledge Catalog and Governance (KCONG) framework[8], developed by Cefriel, which is customised considering the TANGENT *Data Sharing Governance Model* [2,6] and the TangentDCAT-AP metadata specification as metadata profile.

The *Data-Sharing Governance model* supports the strategic and operational management of data-sharing. The model focuses on the tasks/processes related to providing access to data sources needed by the various technical components. It identifies key processes (data publication, data quality, data access, data storage, data usage) to be addressed by stakeholders with different roles and following specific rules. The identified roles are: (i) *Data publisher*, a person responsible for publishing and describing a data source within the catalogue; (ii) *Data Management Board (TMB)*, a group of people responsible for the management and control of a (set of) data source(s) within the catalogue; (iii) *Data user*: a person accessing and using a data source available in the catalogue. As an example of the defined rules (fully described in [6]), the Lisbon case study leader acts as a data publisher and can create and modify only the metadata descriptions of data sources related to the Lisbon case study.

The definition of the TangentDCAT-AP metadata specification considered best practices, particularly the reuse of well-known vocabularies for metadata.

[8] https://kcong.cefriel.com/.

For this reason, TangentDCAT-AP is defined as an extension of DCAT-AP [29], considering the requirements for mobility data platforms elicited by the NAP-CORE project [25] and specific requirements for the description of data sources elicited within the TANGENT project [6]. The final release of TangentDCAT-AP is compatible with the first official release of mobilityDCAT-AP[9] by the NAPCORE project. The mobilityDCAT-AP specification extends DCAT-AP by focusing on requirements for data sources in the mobility domain. It will be recommended as the reference metadata profile for National Access Points[10] and adopted for the European Mobility Data Space[11].

TangentDCAT-AP is documented at https://knowledge.c-innovationhub.com/tangent/tandcatap. Following the best practices [25], the additional properties defined by TangentDCAT-AP have been also published online together with the defined controlled vocabularies. The published vocabularies are hosted on GitHub[12] and served through content negotiation. The vocabularies define possible statuses assigned to a data source, data requirements to categorise the content of data sources and their types.

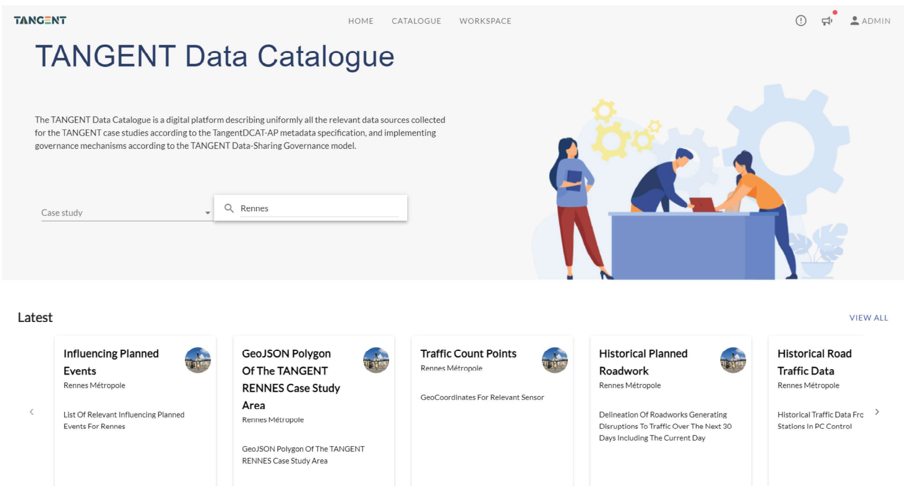

Fig. 2. Homepage of the Data Catalogue.

Figure 2 shows the homepage of the Data Catalogue and some available data sources. Three main areas are available to the user: the homepage, the catalogue and the workspace. The homepage is designed to show the latest changes and the most common interactions with the catalogue, i.e., searching. The catalogue

[9] https://w3id.org/mobilitydcat-ap/releases/1.0.0/.
[10] https://napcore.eu/release-of-the-mobilitydcat-ap/.
[11] https://www.linkedin.com/company/deployemds.
[12] https://github.com/cefriel/tandcatap.

area allows the user to browse all available data sources and perform more fine-grained searches. Lastly, the workspace area allows visualising/editing a specific data source, or getting an overview of the state of the data sources owned by the current user. An external identity and access management solution is used to authenticate and authorize different users to access data sources.

Currently, the Data Catalogue contains metadata about 145 data sources.

3.2 Reference Conceptual Model

The *Reference Conceptual Model* (RCM) supports the representation of heterogeneous information from different data sources through a common ontological model to enable shared semantics and interoperability. The model is based on existing data standards to adopt the correct domain terminology and covers the representation of all the entities and properties required to implement meaningful data exchanges among the involved stakeholders.

In the mobility domain, several ontological models have been proposed. However, they cover specific requirements, and it is not possible to identify a generic and well-adopted ontology [18]. For these reasons, we started our work by analysing existing data standards to support the identification of the relevant semantics. Then, following the best practice of reusing existing models [11], relevant ontologies to be included in the RCM were identified.

We started with the analysis of data standards requested by the European Commission (EC) Delegated Regulations (DR) mentioned in Sect. 2.1. The data standards mentioned in those directives are: (i) *DATEX II* (https://www.datex2.eu/): the EU standard for the exchange of traffic-related data; (ii) *NeTEx* (https://netex-cen.eu/): the CEN Technical Standards for exchanging Public Transport schedules and related data; (iii) *SIRI* (https://www.siri-cen.eu/): the CEN technical standard for the exchange of real-time information about the planned, current, or projected performance of public transport operations.

To support the definition of the RCM, the existing ontologies encoding the semantics of the mentioned standards have been analysed.

Considering the DATEX II format, two different models have been identified. The first one, directly developed by the DATEX II organization [17], is a JSON-LD serialisation of the DATEX II conceptual model version 3, divided into five main modules[13] : *Payload*, *Common*, *LocationReferencing*, *Situation*, *Road Traffic Data*, *Variable Message Sign*. The second model [15] based on DATEX II was developed by the LOD-RoadTran18 project to support the publication of DATEX II data as Linked Open Data. The advantage of the first model is its full coverage with respect to the DATEX II specification, the one-to-one mapping to classes and properties, and the fact that it is directly defined and published by the DATEX II organization; however, this model is defined as an almost automatic conversion of the DATEX II specification. The advantage of the second

[13] Models used available at https://datex2.eu/vocab/3. Additional modules are now available to accommodate the new versions of the DATEX II specification.

Table 1. Overview of the TANGENT Reference Conceptual Model

Module	Base Standard	Data Requirements	Reused ontologies
Road Transport Network	Datex II	Road Transport Network (roads, limited access zones, etc.)	GeoSPARQL, Basic Geo, Datex II JSON-LD (location, common)
Road Equipment	Datex II	Road Equipment Position	Datex II JSON-LD (location), DC Terms, Schema.org
Road Traffic Data	Datex II	Road Traffic Measurements (traffic occupancy, speed, flow) Floating Vehicle Data (GPS, mobile, etc.)	Datex II JSON-LD (traffic)
Road Travel Times	Datex II	Road Travel Times (external services, statistics, etc.)	Datex II JSON-LD (location, traffic)
Events	Datex II	Road Transport Network Events (planned) / Incidents (unplanned) Influencing Planned Events (sports, entertainment, etc.) Weather Events	Datex II JSON-LD (situation, location, common)
Weather Data	Datex II	Forecasted Weather Data Weather Data (measurements, e.g., temperature, humidity, etc.)	Datex II JSON-LD (location, traffic, common)
Stop Points	NeTEx	Public Transport Network	Transmodel ontology (commons, journeys), Basic Geo
Schedules	NeTEx	Public Transport Schedules and Lines	Transmodel ontology (commons, journeys, organisations)
Situation Exchange	SIRI	Public Transport Network Events (planned) / Incidents (unplanned)	Basic Geo
Vehicle Monitoring	SIRI	Floating PT Vehicle Data Public Transport Delays	Basic Geo

model is that the LOD-RoadTran18 project followed proper ontology engineering methodologies to define the model; however, it covers only a portion of the DATEX II specification. For these reasons, we selected the first model.

The NeTEx and SIRI standards are based on the Transmodel[14] conceptual model. The Mobility Ontology Catalogue[15] defines a suite of ontologies based on existing standards, including a Transmodel ontology. The Transmodel ontology, firstly defined within the SNAP[16] project, has been extended and reviewed over the years [23] and currently defines five submodules: *Core*, *Commons*, *Fares*, *Facilities*, *Journeys*. The Transmodel ontology does not cover the entire Transmodel but was used to effectively support mappings to NeTEx [27]. To the best of our knowledge, a dedicated SIRI ontology does not exist. However, since SIRI is derived from Transmodel, the current Transmodel ontology can be exploited to represent common concepts and possibly extended to represent the missing ones. We developed and published a dedicated ontology representing concepts and relations mapped from SIRI, adopting an approach aligned to the one leveraged for the definition of the DATEX II JSON-LD ontology from the DATEX II specification. Moreover, we manually curated the SIRI ontology to improve the alignment with the other ontologies adopted in the RCM. The SIRI ontology is available and documented at https://knowledge.c-innovationhub.com/siri. The current version of the ontology (v1.0.0) focuses on the modelling of situations affecting the transport network and monitored vehicle journeys.

[14] https://www.transmodel-cen.eu/standards/.
[15] https://w3id.org/mobility.
[16] https://snap-project.eu/.

Based on the performed analysis, the RCM was defined as a suite of ontologies considering the semantics of relevant EU-mandated standards and the already available related ontologies. The DCI Metadata Terms[17] and Schema.org[18] vocabularies are reused by the Transmodel ontology and similarly also in other RCM modules. Table 1 provides a complete overview of the ten different modules defined for the *Reference Conceptual Model*, summarising the considered base standard for concepts and relationships, the data requirements covered by the module, and the reused ontologies.

The latest release of the RCM is published online[19] including the documentation of the different modules and the serialisation of the additionally defined classes/properties. The definition of the RCM has been guided by the requirements elicited in TANGENT for the harmonisation and fusion of data sources for multimodal traffic management. Nevertheless, the RCM is made available and we recommend its reuse and extension to address additional requirements.

3.3 Semantic Harmonisation and Fusion Pipelines

Transformation pipelines should support the harmonisation and fusion of data sources available by leveraging the *Reference Conceptual Model*. The definition of such pipelines requires the elicitation of harmonization and fusion requirements based on: (a) the analysis of raw data sources stored in the *Data Catalogue*, and (b) the definition of the information required by the other components that are integrated into the solution and the related target output format. We mainly addressed the requirements of two downstream data usage: the need for large-scale historical data for training of machine learning models to support traffic management, the need to access static and (quasi) real-time data to empower a set of innovative applications for traffic managers.

In both cases, the need to overcome data heterogeneity and sparsity requires transformation pipelines. The basic pipeline is composed of a lifting operation, a (set of) graph operations (e.g., to perform data fusion), and a lowering operation.

A flexible and scalable technology for the implementation of the pipelines should provide (i) a set of reusable building blocks that can be configured according to specific requirements, and (ii) a declarative approach to configure the lifting and lowering transformations without developing ad-hoc and hard-to-maintain solutions. Chimera[20] [13] is an open-source solution based on Apache Camel to enable the definition of semantic data transformation pipelines with different components for knowledge graph construction, transformation, validation, and exploitation. The advantage of Chimera is its integration with Apache Camel, providing off-the-shelf and production-ready components to implement Enterprise Integration Patterns and to integrate pipelines with heterogeneous systems (e.g., HTTP API, WebSocket, MQTT). For these reasons, Chimera was

[17] http://purl.org/dc/terms/.
[18] https://schema.org/.
[19] https://knowledge.c-innovationhub.com/tangent/schema.
[20] https://github.com/cefriel/chimera.

selected to implement the semantic any-to-one mapping approach and Apache Camel is leveraged to implement the Data API and smoothly integrate the defined pipelines.

The data transformation from a source format and source semantics to a target ontology, i.e., the lifting process, can be handled by either the RML Component or the Mapping Template Component of Chimera. The lifted data, in the form of an RDF graph aligned with the Reference Conceptual Model, can be manipulated using the operations defined by the Chimera Graph Component. For example, these operations enable the fusion of data and/or the filtering/extraction of certain information. The Mapping Template Component handles the lowering process from RDF to the target data format[21]. For the TANGENT pipelines, we decided to use the Mapping Template Component since it can be applied for both lifting and lowering [26].

The definition of harmonisation and fusion requirements for the solution required the analysis of (i) raw data sources collected in the Data Catalogue, and (ii) the information required by the other components to be integrated into the overall solution. As a result of a collaborative effort among partners in charge of developing downstream data-driven services, we identified the requirements in terms of target output format and the set of data sources to be harmonised. Considering the target output, we identified the need for a harmonised CSV format to support the training of traffic prediction machine learning models, and of JSON schemas to feed the real-time services at runtime via AMQP[22] queues. The semantics of the RCM was used as a basis to support both the annotation of columns in CSV and of fields in JSON[23].

All in all, we were able to harmonize and fuse data from 43 data sources adopting different data formats (mainly CSV, XML, JSON) and more than 30 data models across the 4 urban case studies. Indeed, many data sources were based on custom data models, thus requiring dedicated lifting mappings, and we could reuse them only for GTFS[24] feeds and data from the same data provider. On the other hand, we could leverage the any-to-one approach to define a single lowering mapping for each target output (10 lowering mappings to JSON schemas). Additionally, via dedicated pipelines, we generated 8 historical datasets targeting a CSV format. These pipelines are based on the same lifting mappings but are configured to regularly (e.g., 1-minute frequency) collect, harmonise and fuse data from real-time data sources. We run these pipelines for over 6 months collecting 92GB of compressed data.

[21] Regarding the lowering, we initially investigated the possibility of applying JSON-LD frames [20] to convert the RDF Graph to the target JSON Schemas. However, we encountered difficulties in addressing cases in which the structure of the RDF graph does not directly correspond to the structure of the target JSON.

[22] https://www.amqp.org/.

[23] JSON Schemas are available at https://github.com/cefriel/tangent-model and contain a simplified representation of the information modelled in RDF to minimise the amount of exchanged data; if the data should be consumed as JSON-LD, a proper context can be associated with each JSON message.

[24] https://developers.google.com/transit/gtfs/.

3.4 Data API

The *Data API* provides uniform access to data sources collected through the Data Catalogue and represents the integration point for the overall integrated solution. The Data API aims at solving access issues for both *raw data sources* (raw data as collected and shared by the data publisher) and *harmonised and/or fused data sources* (data produced as the result of a semantic harmonisation and fusion pipeline). Harmonised data sources are represented in the Data Catalogue as different *Distributions*[25] of the same data source, i.e., different serialisations of the same information. The same approach is also used for data sources provided in multiple raw formats, e.g., CSV and JSON. The result of a fusion process is instead added to the catalogue as a new record since it represents a new data source. The Data API gives access to two main types of data sources: (i) *datasets* usually directly downloadable from a specific URL, and (ii) *data services* implementing different interaction mechanisms (e.g., a REST API). The Data API implements the API Gateway pattern [22], thus providing a single and coherent entry point for the final user. The user can access a data source by knowing the endpoint at which the Data API is located and the identifier of the data source to be accessed. If the user is authorized through the Data Catalogue, the Data API handles authorization mechanisms for the different data sources in a transparent way and provides access to them. Moreover, in cases where a data source should be filtered according to specific requirements, the Data API can be configured to provide access only to the relevant data (e.g., adding the proper parameters to filter data according to a defined temporal/geographical scope). A dedicated parameter can be used to request a specific distribution of the data source (e.g., the harmonised format). The Data API was implemented using the Apache Camel framework since: (i) it provides all the relevant components to interact with the different data platforms (NAPs, Open data/private portals, etc.) hosting data sources, (ii) it can be easily integrated with the semantic harmonisation and fusion pipelines.

The integration of an external data source published on the Data Catalogue within the Data API requires different steps. First of all, we perform an analysis of the accessibility metadata provided for the Dataset (access URL/download URL) or the Data service (endpoint URL/endpoint documentation). If the data source is not available online, we contact the responsible stakeholder to get access. In this case, we leverage the storage layer of the Data Catalogue to upload the data and we evaluate the need to define a data service. If the data source is already available online, we investigate the expected data access interaction (e.g., used protocol); in particular, we evaluate authorization and authentication mechanisms. We configure the required Apache Camel component to retrieve the data source (e.g., HTTP component), and define the integration logic (e.g., filtering the data of a data source considering the relevant temporal/spatial scope for the case study). Finally, if available, we integrate the semantic harmonisation and fusion pipeline to allow users to request data in their harmonised format.

[25] https://www.w3.org/ns/dcat#Distribution.

Once a data source is integrated into the Data API, the corresponding metadata to access it are updated in the Data Catalogue. In total, we developed 87 integrations to provide access via the Data API to all the raw/harmonised data sources approved for usage.

4 Evaluation and Lessons Learned

This section discusses an evaluation of the proposed solution considering various perspectives and the lessons learned.

Technical evaluation. The integrated on-cloud deployment consisted of testing and production environments with a cluster of four virtual machines and a set of managed services (e.g., database and load balancer). The runtime data for the four case studies are handled by 32 AMQP message exchanges fed by the Data API. The current data are refreshed with different periodicity depending on the considered data source and are organised with static data in 91 distinct collections on the database. The solution processes 130 messages per minute at peak time and manages around 4 GB of data at a time only to visualise the network's current status. The implemented solution demonstrates the feasibility and advantages of adopting an approach for data harmonisation and fusion based on Semantic Web technologies also to support real-time visualisations and data-driven services in a production-ready environment. The semantic any-to-one approach supported good scalability considering the high number of data sources involved and their substantial heterogeneity. Moreover, we demonstrated how it is possible to define system integrations that seamlessly combine semantic harmonisation and fusion to enable interoperability of data exchanges. We also positively assessed that the executed transformations introduced negligible latency (in the order of milliseconds) considering the update frequency of the data sources (often in the order of minutes). Finally, we highlighted the possibility of leveraging the same solution to generate datasets for training machine learning models. A pipeline defined for harmonisation could be easily configured to collect and aggregate data from real-time data, thus generating a historical dataset of harmonised data. Such an approach, relying on common semantics, reduces enormously the effort needed by data scientists to assess heterogeneous input datasets and facilitates the reuse of training algorithms (e.g., for different cities considering different data sources).

User evaluation. To gather opinions and feedback from real users, we performed a qualitative evaluation involving 43 business stakeholders such as transport operators and authorities. They highlighted the importance of a Data Catalogue with structured metadata descriptions to reduce the current scattering of information on data sources and facilitate their retrieval. Indeed, we experienced this difficulty ourselves during the data collection phase: retrieving updated and consistent information about data sources often required the involvement of different people within the same company and/or third-party solution vendors. Based on the evaluation feedback, we also improved the Data Catalogue by

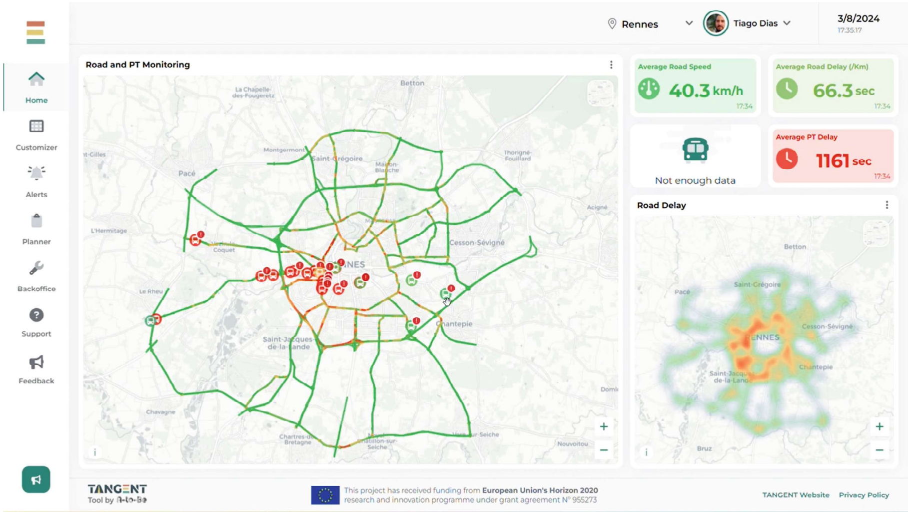

Fig. 3. TANGENT Dashboard for Rennes.

customising filtering operators to further facilitate data discovery. In parallel, we performed the testing of the integrated TANGENT solution (cf. Fig. 3) for the visualization of the current and forecast status of the multimodal network for city and transport authorities, integrating different data-driven services for intelligent incident detection, response plan prescription and the cooperative management of incidents [10]. The stakeholders involved in its evaluation highlighted the advantage of having data from different transport modes available on the same platform in integrated visualizations. The final detailed assessment of the case studies will be included in the future TANGENT deliverables D7.3-D7.6 (due Q4 2024).

Significance Evaluation. The advantages of the proposed solution over potential alternatives are: (i) the collection of structured and actionable metadata in a common machine-readable format that facilitates the findability of heterogeneous data source through the Data Catalogue, and (ii) the reduction of the integration effort for downstream interoperable data-driven services, through the Semantic Harmonisation and Fusion Pipelines and the subsequent Data API: we removed the need for point-to-point integration between different components and we reduced at minimum the custom development for the different deployments in each urban case study.

Uptake and Impact Evaluation. The current uptake of the solution is confined to the involved stakeholders in the four urban case studies, which, however, represent a significant sample of the target market of traffic management solutions and were able to test the solution in their daily operations over an extended period. Indeed, they pointed out the absence of a similar solution in their existing infrastructures and its relevance for both internal governance and traffic management. Moreover, an analysis of the possible exploitation in other cities and the integration of the presented solution with the commercial offering of A-to-Be are currently ongoing. Concerning the potential impact, we defined a Reference Conceptual Model based on the semantics of existing standards that could be adopted (and possibly extended) to foster interoperability of different solutions for traffic management. Similarly, we demonstrated how to define a metadata extension that supports compatibility with other data portals (leveraging mobilityDCAT-AP) but fulfils additional requirements. Finally, the technological solutions developed for traffic management are not dependent on the mobility domain: with the opportune use of metadata specifications, domain ontologies and the definition of specific mappings, the presented components can be easily configured and adapted to any other domain or market, bringing the same interoperability advantages.

Lessons Learned. We now discuss some of the lessons learned referencing the five challenges in Sect. 2. Regarding the *Locate* challenge, we experimented the difficulty of obtaining good quality metadata. Structured descriptions of existing data sources are often not available, and the Data Catalogue not only enforced common metadata descriptors but also provided users with useful guidelines to collect high-quality metadata (e.g., guided and dynamic forms for metadata insertion). Concerning the *Access* challenge, we experienced the advantage of having an integrated solution for data findability and access. Indeed, often data portals act as simple metadata catalogues, only referencing existing data sources and without taking into account difficulties in accessing the actual data (e.g., authorization, missing documentation of data services, etc.); our Data Catalogue also incorporates and hides the complexity of data harmonisation, giving access to a uniform data API. Considering the *Harmonise* challenge, it is often hard to define common semantics to support different use cases. For this reason, it is important to leverage the semantics already encoded in existing standards without reinventing the wheel and to facilitate the adoption of ontologies by domain experts. Regarding the *Integrate* challenge, it is often difficult to integrate certain types of data (e.g., geographical data considering different location referencing methods or identifiers of stop stations across different transport modes). In these cases, we managed to implement complex transformations by defining mapping rules that integrate custom functions, and by leveraging data fusion with external data sources that specify the correct correspondence between values. Finally, concerning the *Extract* challenge, we demonstrated that a reference ontology supports the generation of harmonised outputs also in formats different from RDF (in our case, CSV for model training and JSON schemas for runtime interactions). This not only facilitates the integration with systems unable to process RDF, but also reduces the size of exchanged data (and, consequently, latency) by avoiding possibly verbose RDF representations.

5 Conclusions

This paper has comprehensively described the proposed solution to address data interoperability challenges within a complex traffic management scenario and its validation within four European cities. The solution guarantees interoperable descriptions of the data sources and applies the any-to-one centralized approach for semantic interoperability, enabling data exchange with unambiguous and shared meaning.

The proposed solution consists of four components: the Data Catalogue, for sharing uniform data source descriptions according to the TangentDCAT-AP metadata profile and for enforcing governance mechanisms; the Reference Conceptual Model, a reference ontology defining common semantics for multimodal traffic data; the Semantic Harmonisation and Fusion Pipelines, to fulfil heterogeneous data integration requirements; and the Data API, a uniform mechanism to access all data sources. Semantic Web technologies proved their efficacy in addressing data interoperability challenges and providing a production-ready to integrate data in downstream services and applications.

Supplemental Material Statement: Public deliverables are available on the TANGENT website at https://tangent-h2020.eu/deliverables/. Implementation reports and related artifacts (e.g., source code) for the TANGENT solution are part of confidential deliverables and can not be shared. The Reference Conceptual Model is published online at https://github.com/cefriel/tangent-model, TangentDCAT-AP and controlled vocabularies at https://github.com/cefriel/tandcatap. The Chimera framework used for the implementation of the pipelines is available at https://github.com/cefriel/chimera. A video describing the TANGENT solution can be visualised at https://youtu.be/lrwu79lIx4k?feature=shared.

Acknowledgments. The presented research was partially supported by the TANGENT project (Grant Agreement 955273), which is funded by the European Commission under the Horizon 2020 Research and Innovation Programme. The authors would like to acknowledge Antonia Azzini, Alessio Carenini, Andrea Fiano, and Gianluca Gizzi from Cefriel for their contribution to the development of the TANGENT solution.

References

1. Albertoni, R., Browning, D., Cox, S., Gonzalez Beltran, A., Perego, A., Winstanley, P.: Data Catalog Vocabulary (DCAT) - Version 2. https://www.w3.org/TR/vocab-dcat-2/
2. Azzini, A., Comerio, M., Metta, S., Scrocca, M.: The TANGENT governance model for mobility data sharing. In: Transport Transitions: Advancing Sustainable and Inclusive Mobility - Proceedings of the 10th TRA Conference. LNM, Springer (2024), (to appear)
3. Carenini, A., Fiano, A., Scrocca, M., Comerio, M., Celino, I.: Enabling cross-border travel offers through National Access Point federation via metadata harmonisation. In: Proceedings of the 3rd International Workshop Semantics And The Web For Transport. CEUR Workshop Proceedings, vol. 2939. CEUR, Online, September (2021). http://ceur-ws.org/Vol-2939/paper6.pdf, iSSN: 1613-0073
4. Comerio, M., Fiano, A., Grassi, M., Scrocca, M.: Mobility data harmonisation: the TANGENT solution. In: Transport Transitions: Advancing Sustainable and

Inclusive Mobility - Proceedings of the 10th TRA Conference. LNM, Springer (2024), (to appear)
5. Comerio, M., et al.: TANGENT project deliverable D2.1: data requirements and available data sources (2022). https://tangent-h2020.eu/deliverables/
6. Comerio, M., et al.: TANGENT project deliverable D2.2: data sharing governance model (2022). https://tangent-h2020.eu/deliverables/
7. European Commission: Delegated Regulation No 886/2013 supplementing directive 2010/40/EU of the european parliament and of the council with regard to data and procedures for the provision, where possible, of road safety-related minimum universal traffic information free of charge to users (2013). https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32013R0886&from=EN
8. European Commission: delegated regulation No 2015/962 supplementing directive 2010/40/EU of the european parliament and of the council with regard to the provision of EU-wide real-time traffic information services (2015). https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32015R0962&from=EN
9. European Commission: delegated regulation No 2017/1926 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/legal-content/EN/TXT/PDF/?uri=CELEX:32017R1926&rid=6
10. Dias, T., Vieira da Silva, A., Trigueiro Moura, L., Matos, D.: Software patterns and architectural decisions for a next-generation traffic management system in the TANGENT project. In: Transport Transitions: Advancing Sustainable and Inclusive Mobility - Proceedings of the 10th TRA Conference. LNM, Springer (2024), (to appear)
11. Garijo, D., Poveda-Villalón, M.: Best practices for implementing FAIR vocabularies and ontologies on the web. In: Cota, G., Daquino, M., Pozzato, G.L. (eds.) Applications and Practices in Ontology Design, Extraction, and Reasoning, Studies on the Semantic Web, vol. 49, pp. 39–54. IOS Press (2020). https://doi.org/10.3233/SSW200034, https://doi.org/10.3233/SSW200034
12. Gendre, P., et al.: CHOUETTE: an open source software for PT reference data exchange. In: ITS Europe (2011)
13. Grassi, M., Scrocca, M., Carenini, A., Comerio, M., Celino, I.: Composable semantic data transformation pipelines with Chimera. In: Proceedings of the 4th International Workshop on Knowledge Graph Construction co-located with 20th Extended Semantic Web Conference. CEUR Workshop Proceedings, vol. 3471. CEUR, Hersonissos, Greece (2023). https://ceur-ws.org/Vol-3471/paper9.pdf, iSSN: 1613-0073
14. Guerreiro, G., Figueiras, P., Silva, R., Costa, R., Jardim-Goncalves, R.: An architecture for big data processing on intelligent transportation systems. an application scenario on highway traffic flows. In: 2016 IEEE 8th International Conference on Intelligent Systems (IS), pp. 65–72 (2016). https://doi.org/10.1109/IS.2016.7737393
15. Gutierrez, J., Samper, J.J., Delgado, A., Rocha, J., Petr, B., Zuzana, P.: LOD-RoadTran18: supporting the cross-border use of road traffic data with linked open data based on DATEX II. In: Proceedings of the 11th Euro American Conference on Telematics and Information System, pp. 1–8. EATIS 2022, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3544538.3544651
16. Hohpe, G., Woolf, B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley Professional (2004)

17. Jäderberg, J.: DATEX II and Linked Open Data - Session 4 (2021). https://www.youtube.com/watch?v=5XkMefwptyE. Accessed 16 July 2024
18. Katsumi, M., Fox, M.: Ontologies for transportation research: a survey. Transp. Res. Part C: Emerging Technol. **89**, 53–82 (2018). https://doi.org/10.1016/j.trc.2018.01.023
19. Landaluce, H., et al.: The TANGENT project architecture: towards new traffic management approaches. In: Transport Transitions: Advancing Sustainable and Inclusive Mobility - Proceedings of the 10th TRA Conference. LNM, Springer (2024), (to appear)
20. Longley, D., Kellogg, G., Champin, P.A.: JSON-LD 1.1 Framing. https://www.w3.org/TR/json-ld11-framing/
21. Parliament, E.: Directive 2010/40/eu on the framework for the deployment of intelligent transport systems in the field of road transport and for interfaces with other modes of transport (2010). https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32010L0040&from=EN
22. Richardson, C.: API Gateway pattern. http://microservices.io/patterns/apigateway.html. Accessed 16 July 2024
23. Ruckhaus, E., Anton-Bravo, A., Scrocca, M., Corcho, O.: Applying the LOT methodology to a public bus transport ontology aligned with Transmodel: challenges and results. Semant. Web **14**(4), 639–657 (2023). https://doi.org/10.3233/SW-210451, publisher: IOS Press
24. Sadeghi, M., Carenini, A., Corcho, O., Rossi, M., Santoro, R., Vogelsang, A.: Interoperability of heterogeneous systems of systems: from requirements to a reference architecture. J. Supercomput. **80**(7), 8954–8987 (2024). https://doi.org/10.1007/s11227-023-05774-3
25. Scrocca, M., Azzini, A., Bureš, P., Comerio, M., Lubrich, P.: Towards napDCAT-AP: roadmap and requirements for a transportation metadata specification. In: Şimşek, U., Chaves-Fraga, D., Pellegrini, T., Vahdat, S. (eds.) Proceedings of Poster and Demo Track and Workshop Track of the 18th International Conference on Semantic Systems. CEUR Workshop Proceedings, vol. 3235. CEUR, Vienna, Austria (2022). https://ceur-ws.org/Vol-3235/paper22.pdf, iSSN: 1613-0073
26. Scrocca, M., Carenini, A., Grassi, M., Comerio, M., Celino, I.: Not everybody speaks RDF: knowledge conversion between different data representations. In: Proceedings of the 5th International Workshop on Knowledge Graph Construction. CEUR Workshop Proceedings, vol. 3718. CEUR, Hersonissos, Greece (2024). https://ceur-ws.org/Vol-3718/paper3.pdf, iSSN: 1613-0073
27. Scrocca, M., Comerio, M., Carenini, A., Celino, I.: Turning transport data to comply with EU standards while enabling a multimodal transport knowledge graph. In: Proceedings of the 19th International Semantic Web Conference, vol. 12507, pp. 411–429. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62466-8_26
28. Van Assche, D., Delva, T., Haesendonck, G., Heyvaert, P., De Meester, B., Dimou, A.: Declarative RDF graph generation from heterogeneous (semi-)structured data: a systematic literature review. J. Web Semant. 100753 (2022). publisher: Elsevier
29. Van Nuffelen, B.: DCAT Application profile for data portals in Europe (DCAT-AP). https://joinup.ec.europa.eu/collection/semantic-interoperability-community-semic/solution/dcat-application-profile-data-portals-europe/release/211. Accessed 16 July 2024
30. Vetere, G., Lenzerini, M.: Models for semantic interoperability in service-oriented architectures. IBM Syst. J. **44**(4), 887–903 (2005). https://doi.org/10.1147/sj.444.0887

Quality in Color: Using Knowledge Graphs for Enhanced Quality Control in an Automotive Paintshop

Bram Steenwinckel[1(✉)], Colin Soete[1], Pieter Moens[1], Joris Mussche[2], Sofie Van Hoecke[1], and Femke Ongenae[1]

[1] IDLab, Ghent University – imec, 9000 Gent, Belgium
bram.steenwinckel@ugent.be
[2] Volvo Cars Group, 9000 Gent, Belgium
joris.mussche@volovcars.com

Abstract. Sensors and their derived data have reshaped manufacturing by enabling real-time monitoring, improved quality control and enhanced safety, particularly in automotive processes. Despite these benefits, challenges arise within the realm of data discovery when they have to align all this data, originating from different systems and built by different manufacturers using different technologies. This paper shows our solution to resolve these challenges within the Volvo Cars paintshop. By organizing sensor metadata, link them with the originating devices and the place where the data will be stored in a knowledge graph, our approach facilitates seamless data integration for multiple paintshop applications. We show how the resulting knowledge graph helps with dashboard techniques, machine learning, and semantic reasoning to provide insights for the paintshop operators. The obtained findings clearly show the potential of knowledge graphs in production lines, paving the way for future advancements in automotive manufacturing.

Keywords: Automotive paintshop · Knowledge Graph · Semantic reasoning · Machine learning · Dashboard visualization · Industry 4.0

1 Introduction

Industry 4.0 has fundamentally transformed manufacturing operations due to the creation of autonomous systems for production processes [13]. In automotive production lines, many of these automated processes are nowadays applied to manufacture cars. Applying paint is one such process, playing an essential role in creating the final aesthetics and durability of cars [1]. This process typically involves applying various paint layers, including primer, topcoat, and clear coat, to the vehicle bodies. Each paint layer can be seen as a different process that each exists out of multiple procedures that follow one another.

Within the Volvo Cars paintshop, these different paint procedures are physically segregated across various stations. Stations are equipped with a range

G. Demartini et al. (Eds.): ISWC 2024, LNCS 15233, pp. 236–252, 2025.
https://doi.org/10.1007/978-3-031-77847-6_13

of devices, so-called Internet of Things (IoT) sensors and actuators, e.g. paint robots and ventilation units, which are systems from various manufacturers. This results in a diverse and complex ecosystem of technologies that must seamlessly interact. Figure 1 schematically shows car bodies on a conveyor belt going through four different stations. At the end of all these paint processes, each car body undergoes a final inspection at the Quality Control (QC) station. This station is vital to ensure that the vehicles meet the stringent quality standards of the Volvo Cars group before leaving the production line. The QC inspection is a manual process. An operator conducts visual checks to identify any defects or errors that may have occurred during the painting process, such as uneven paint layers, scratches, or paint defects.

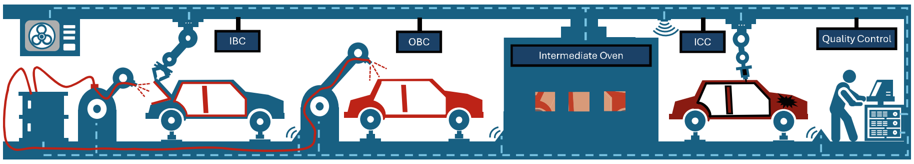

Fig. 1. Schematic overview of a subprocess within an automated paintshop. The Interior Base Coat (IBC) station is responsible for painting the interior parts of the car. Next, the Outer Base Coat (OBC) station applies the exterior base coat. An intermediate oven is used to dry the car body before moving towards the Interior Clear Coat (ICC) station. At the end, a manual inspection of the car is performed to control the quality, e.g. detect scratches.

The robots within these stations are precisely programmed to perform a predetermined sequence of actions. They are programmed using industry standards, and controlled by Programmable Logic Controllers (PLCs). PLC code can be seen as a set of instructions that are sequentially executed. It is the PLC that steers every actuator within the station and starts the different robot programs based on available parameters and sensor inputs. The PLC plays a crucial role in coordinating all these operations. As car body trackers are placed at the input and output regions of each station, different robot programs with adjusted parameters can be loaded by the PLC based on the car type and the desired car color.

While the tracker data directly steers the dynamic behaviour of the paint application, the Volvo Cars group extensively collects all the available station data. This includes car types, color compounds and more, beyond basic production information, including the robot operations, robot faults, PLC operations, PLC faults, ventilation metrics and temperature values at each station. The records defining the paint faults observed during the QC process are also stored alongside each car body. The data is stored in separate databases, laying the foundation for further analytics. It opens the possibility for system operators to optimize the production and enhance the overall efficiency or quality of the product [2].

An important shortcoming of the current QC process is the lack of additional information that the operator can use to identify the causes of any defects and assess whether subsequent cars might have similar issues. While it is not directly indicated in Fig. 1, many stations are duplicated in separate lines to parallelize and speed up the production process. Cars with similar quality issues might have passed the same station and identifying these issues provides information towards necessary maintenance operations for that particular station. On the other side, if the quality operator has station-related information alongside each car body they have to inspect, quality issues could be more easily detected or even predicted upfront.

Providing interoperability over different stations and different devices with different technologies is challenging. This challenge is indicative of broader issues faced within Industry 4.0, where data standardization is crucial for achieving efficient and interconnected manufacturing systems. Data stored within the databases is most of the time defined at an overly granular level. This data exists with meaningless identifiers, making it challenging for operators to further use and digitize the data [11]. To be able to build the necessary insights directly from the data, one often neglected aspect is the comprehensive management of metadata and knowledge about these sensors, defining not only their location and relation to specific processes, but also what they observe and what the semantic significance is of this observation. Such a semantic meaning to the data requires human expertise and it is the station operators that understand how the different devices work together to define the paint process. A common understanding of each subprocess is required to further contextualize and semantically annotate the data.

In this paper, we demonstrate the substantial benefits that can arise from the implementation of a Knowledge Graph (KG) within an Industry 4.0 car paintshop process to obtain further insights. We resolve the following 4 challenges:

- We enable the semantic annotation of the process data using a KG which is created and maintainable with minimal human effort.
- We show how we can use this KG to enable interoperability across production lines and stations and how the data sources can be linked to the available process knowledge.
- We showcase how semantic enrichment enables more insightful visualization of the data for quality control.
- Due to enhancing the identification and investigation of quality issues in the paintshop, we can perform semantically enhanced reasoning and machine learning on the data.

The remainder of this paper is defined as follows. Section 2 defines the process of how the Volvo Cars KG was created. Section 3 showcases three applications benefiting from the created KG for enhanced quality control. Section 4 and Sect. 5 define the different lessons learned from this setup and conclude this work respectively.

2 From Knowledge to Knowledge Graph

To address the uniformity challenges mentioned earlier, a solution based on KGs was devised, integrating data, metadata information, and insights from the paintshop processes. This KG was developed in three sequential phases: 1) establishing unique identifiers, 2) incorporating expert knowledge using ontologies, and 3) connecting the knowledge with existing data resources. Important requirements for this design process were (1) to enable minimal human intervention, automating annotation as much as possible, and (2) when manual input is needed, making this input as user-friendly as possible.

2.1 Establishing Unique Identifiers

Device identifiers in a factory are usually vendor-specific. This can cause confusion, errors, and inefficiencies in the production line. Diverse machinery and stations need to work together, but different identifiers complicate inventory, asset management, tracking, risk management, and quality control. They also hinder system interoperability and data integration. Thus, standardizing identification systems across vendors is crucial for streamlining production and maximizing efficiency.

Device identifiers are typically vendor-specific. Using different methods to identify machines within a production line can cause confusion, errors, and inefficiencies. This can complicate inventory and asset management, tracking, risk management, and quality control. They hinder further interoperability between systems and obstruct data integration efforts. Therefore, standardizing the identification of machines and systems across vendors is essential for streamlining production processes and maximizing efficiency.

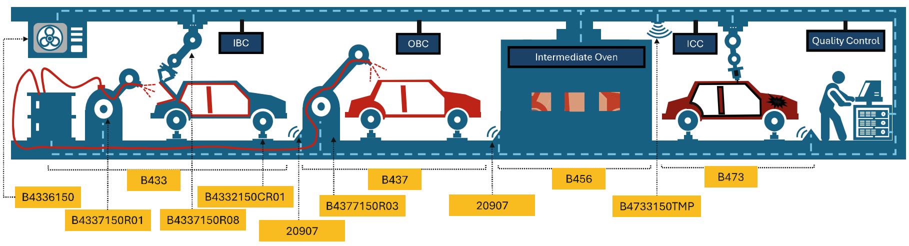

Fig. 2. Overview of the identification of the different components within the paintshop production line. Formatting the identifiers is based on a hierarchical schema where the location of a certain component can be deduced from the identifier.

When the identification of systems and machines adheres to a well-defined standard, it not only ensures the uniqueness of the numbers, but can also provide substantial information through the formatting of the IDs. As shown in Fig. 2, Volvo let their IDs follow a specific standard to define the different equipment within the whole factory. Different shop floors are first identified with a single character (here 'B', indicating the specific building where the paintshop resides).

Next, three numbers are used to define the zone and specific line followed by three digits to specify the station. Finally, more specific devices within each station are defined, such as 'R' with two digits to identify a specific robot. These types of hierarchical identifiers help operators to understand and directly identify each component within the factory. As opposed to the car body tracking devices, which are identified in Fig. 2 by simple integer numbers, they do provide, even across Volvo Cars factories, some of the necessary information about tracking and tracing components. Hierarchical unique identifiers are necessary to start linking machines and systems in process together. However, the semantic description of each component's functioning and how they interact with each other is limited.

2.2 Production Line Ontology Design

While unique identifiers offer precision in tracking and tracing, it is the consolidation of shared knowledge of the domain experts, i.e. operators, on how the paintshop operates and how the data should be interpreted that presents a significant stride towards a holistic understanding and operational optimization [7]. This knowledge can be contextualized in the form of domain-specific ontologies. However, the journey of ontology design in specific domains is full of challenges, time-consuming and labor intensive [21]. They do, however, offer advantages, such as facilitating collaboration and consensus-building among experts within a particular context [15]. In the context of Industry 4.0, domain experts may also face a learning curve, as ontology design paradigms might not be familiar to them. Moreover, the true benefits of a domain ontology manifest only upon its integration into various applications, forming a threshold for domain experts to value its importance during construction. Hence, expediting the construction process, perhaps through automation techniques like extracting common-sense knowledge from documents or using user-friendly tools, is often desired.

Various concepts needed within industry 4.0,e.g. processes, sensors, and faults, are already defined in well-established upper ontologies such as SNN [3] and FOLIO [18]. Both were reused and linked together in the Systematized Procedure for Automating the Retrieval of Knowledge within Smart industries (SPARKS) upper ontology [19]. Instead of focusing on how observations are created following a procedure with one or more devices, SPARKS defines the influence observable properties have on the procedure. These influences are more commonly known as process or procedure parameters. SPARKS additionally allows to order procedures after each other, making it more clear how systems interact with multiple procedures and how the different procedures eventually lead to a process. An overview of this SPARKS ontology is provided in Fig. 3 and this ontology has been open-sourced for future industry 4.0 use cases[1].

To extend this upper ontology to a domain-specific one, e.g. for a paintshop, domain experts will need to define information regarding their specific production processes, systems and faults and link the unique identifiers to the defined concepts. To automate this tedious manual process and to prevent the domain

[1] https://predict-idlab.github.io/SPARKS/.

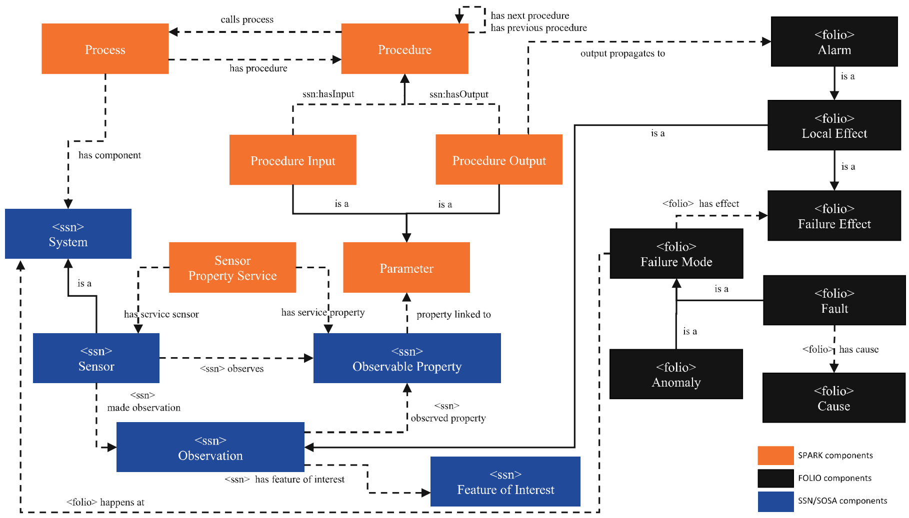

Fig. 3. Overview of the SPARKS upper ontology, which combines different upper ontologies and aligns them to processes and procedures

expert from requiring experience with semantic web technologies, a graphical user interface (GUI) has been developed to provide this information, as shown in Fig. 4. This GUI uses automated scripts to transform the graphical input provided by operators to semantic representations, based on the SPARKS ontology.

To ensure the expert does not have to start from scratch, the GUI enables to load expert documents and automatically map them on the SPARKS ontology. In this research, we made it possible to directly load PLC code. Transformation scripts were created based on the PLC Statement List (STL) standard to connect PLC function blocks and their relationships to the occurring paintshop production processes. These transformation mechanisms are standardized in a well-defined specification, allowing the GUI to be easily extended to also enable the seamless import of other structured files, beyond PLC code. A video defining the different GUI interactions is made available online[2]. Figure 4 shows the semantification process of the *IBC* process and the corresponding station is shown as an example. The GUI interface has a canvas where different widgets can be specified and linked. Each widget corresponds to a key concept within the SPARKS ontology, but can be further characterized by the settings pane (see Fig. 4 on the left). It is also possible to provide additional concepts and provide more information regarding occurring faults at each component, their cause and possible mitigation action. The GUI, the transformation script and the SPARKS methodology can be found in the corresponding repository[3].

[2] https://youtu.be/BzLiNpzNHU8.
[3] https://github.com/predict-idlab/SPARKS.

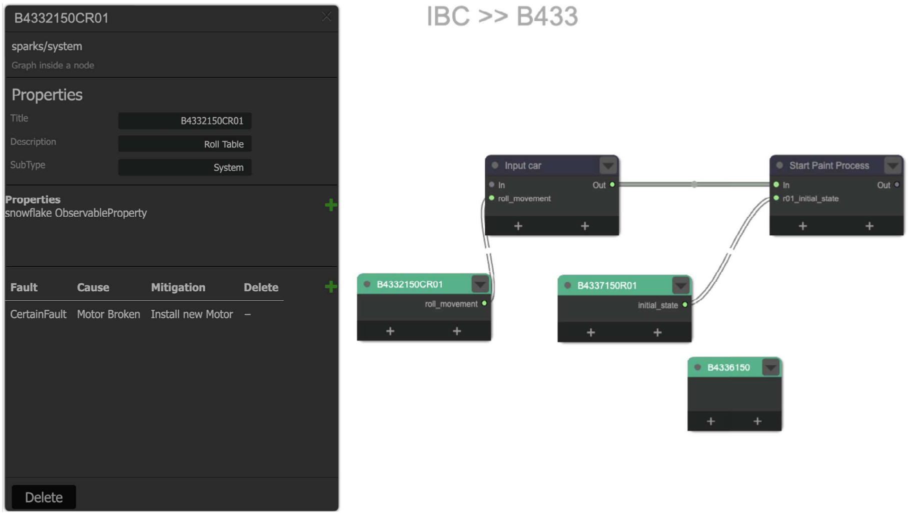

Fig. 4. Example of the information can be specified in the designed GUI. It shows two procedures within the IBC process, which was originally already uniquely defined. Three systems are provided, of which two influence two subsequent procedures following each other. A detail panel on the left is provided for the roll table, responsible for the transport of the car bodies within the station.

2.3 Linking Data to Knowledge

The interaction between data and semantic components in manufacturing environments unlocks the full potential of unique identifiers, especially to monitor and evaluate the different processes [13]. The challenge arises when attempting to align the data generated by different manufacturers' software with the predefined semantic structure. Often, data arrives in various formats and may be stored across multiple databases, complicating the efforts to integrate it seamlessly into a unified semantic representation. In an ideal scenario, a comprehensive overhaul and alignment of data silos would be the optimal solution. However, practical constraints, such as costs and time constraints often limit the feasibility of such endeavors. The solution used here is to semantically describe the data source alongside the device and the properties that the device can observe.

The SPARKS ontology provides a concept to define the link between sensors and observable properties in the form of a `Sensor Property Service`. Subclasses of this concept can work twofold: 1) They provide a service concept that links the sensor and the property that it observes, and 2) They can add additional data properties to provide all the necessary information to acquire the corresponding data, e.g. add data properties to define the database schema, login information and specific location within the database to reach the correct sensor data.

The GUI editor is able to provide this link for each component by specifying the `Sensor Property Services` for each device. This can be seen in the left panel in Fig. 4, where different sensor property services can be added using the

'+' sign. Different types of property services were defined in our KG for this project, based on the different data silo technologies. In the example of Fig. 4, a Snowflake Sensor Property Service was defined for the roll table and an arbitrary Observable property (e.g. Tracking). Snowflake is a SQL-typed database [4], and we defined this Sensor Property Service concept with additional data properties to access this Snowflake database. These data properties can also be specified within the GUI editor directly when the specific Sensor Property Service is instantiated. Under the hood, the GUI uses automated mapping scripts to transform all these operator-provided graphical inputs to semantic representations [19].

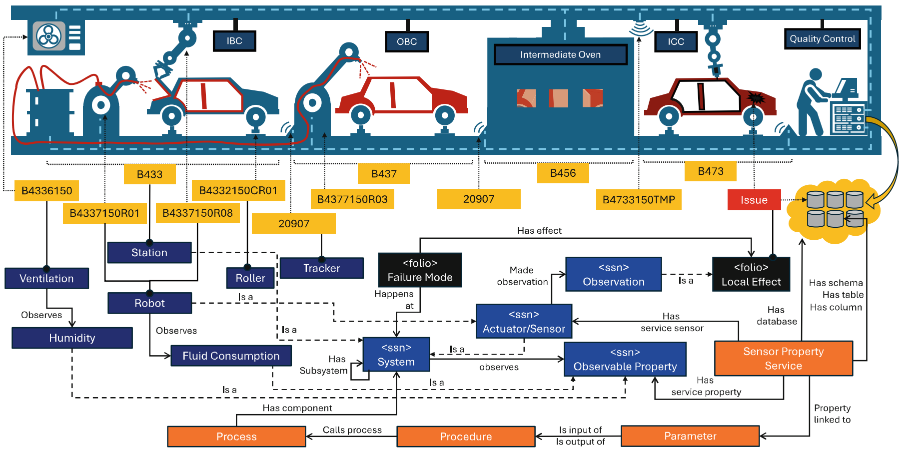

Fig. 5. Example of information about a paintshop production line being semantically annotated and interlinked to data available in data silos using the SPARKS ontology

The data linked to occurring faults of devices or quality issues is also available within different databases and can be linked in a similar way as described above. The overall process of identification, semantic annotation and linking of data resulted in a system where both the knowledge of the system procedures and faults occurring at those systems are uniformly defined and become easily queryable for further analyses. Figure 5 shows this linked knowledge schematically.

In addition to the equipment in each station, car bodies themselves can also be uniquely defined. They are tracked using RFID scanners to follow up on the production configuration. It is based on these body numbers that configuration parameters are defined, such as the paint color, which are in turn directly used as input for the PLC program to start the correct paint processes. While what happens in the production line might affect the quality of the product, all the data and sensor/actuator information available in the data silos about each station does not have a direct link to the car body that was in a specific station at the particular moment these observations were made. The KG can be used to enable this interlinking. The car body, or in more general terms the product, is a so-called feature of interest. The station trackers in our example are responsible for scanning the car body identifiers and providing timestamps when a car body enters the assigned station. These scanning devices are defined in our KG, making it possible to request the timestamps for a car body identifier when it is in a particular station. The Sensor Property Services can be used to query the data of all devices within this station, taking into account the ranges of the obtained timestamps. This results in data that can be linked directly to an instantiated car body concept.

3 Enabling Quality Control Through KG-Enhanced Decision Support

The outcome of Sect. 2 is a KG that uniquely connects and identifies instances within the paintshop production line. These links between data and identifiers present numerous opportunities to assist paintshop operators with their daily tasks. As indicated previously, the operator was not able to directly access the necessary insights or station errors that occurred during manual inspection of the car's paint quality. This also hindered solutions to detect quality issues automatically. To show how the KG can assist in resolving this, we elaborate on three predictive maintenance use cases that were realized.

3.1 Knowledge-Driven AI on Semantic Data

Beyond concepts and links between these concepts in a KG, rules can further encapsulate the domain-specific insights derived from the expertise of operators and quality control engineers. They can play a role in decision-making processes by providing actionable guidelines based on observations and experiences. An illustrative example of such a rule could identify the correlation between temperature fluctuations in a specific station and an increase in paint drip quality issues observed on the car bodies that traversed the station when the temperature fluctuations occurred. Such an expert-driven rule is formally provided in Eq. 1.

$$\begin{aligned}&\text{car}(?x) \wedge \text{hasLog}(?x, ?y) \wedge \text{metricTempValue}(?y, ?v) \wedge \\ &\text{greaterThan}(?v, 18) \rightarrow \text{sparks:Fault}(?x, ?v)\end{aligned} \tag{1}$$

The availability of our KG, the fact that we can combine all data and link it to a car body, and the ability to perform rule-based reasoning upon all this data, make it possible for these cases to come up with a knowledge-driven AI system to automatically inform quality operators of possible faults occurred at the car body they are inspecting. We created a solution based on rules expressed in the Semantic Web Rule Language (SWRL) [8] in combination with the OWL-Ready2 [10] Python package to gain the capability to effortlessly load knowledge

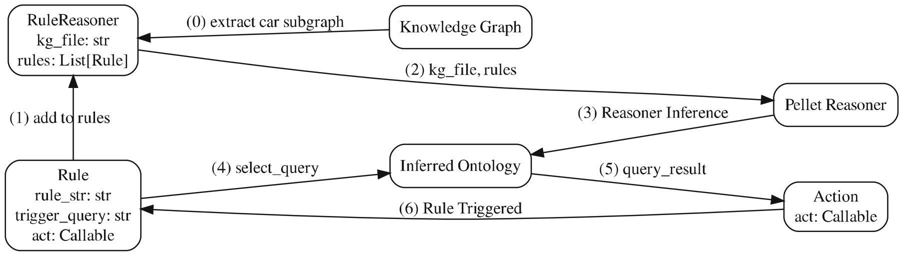

Fig. 6. Flow of the Rule-based Python reasoner created to act upon triggered rules.

graphs in OWLReady, interact with ontology classes and individuals, and execute reasoning tasks, all through a Python syntax that prioritizes convenience.

Our solution consists of the following steps. First, the overall KG is queried to generate a new, smaller KG that only contains all data from a particular car body. This smaller KG connects and transforms the data provisioned by the Sensor Property Service to SSN observations. Next, the newly generated KG and rules provided by the expert are given to our created Python rule-based reasoning framework. The flow of this framework is provided in Fig. 6. A rule class within this framework consists of 3 arguments: the SWRL rule, a SPARQL selection query and a Python callable function. The SWRL rule (or multiple SWRL rules when more than one rule is defined) is added together with the car-specific knowledge graph to the RuleReasoner as seen in step one of Fig. 6. Step two in our framework will trigger the Pellet reasoner to infer the materialized knowledge graph based on the available concepts and the provided rules. After this step, the SPARQL query can be used to retrieve the inferred knowledge that we are interested in, e.g. all the derived faults. If knowledge was inferred that adheres to the query, the Python function linked to this query is called and executed.

A simplified Python example is provided in Listing 1.1 to show the rule-based Python reasoning framework functionality. Car data is queried from the general knowledge graph using the query function (line 4), and the retrieved data is stored in the graph variable. Subsequently, a rule named rule1 is defined (lines 6–7) with conditions specifying Eq. 1. Additionally, a corresponding SPARQL query and a callable function are encapsulated to handle the results of the rule (lines 8–9). The fault function is defined (lines 11–12) to process the results of the rule by printing a message indicating a fault in the car. Finally, a RuleReasoner object named reasoner is initialized (line 14), passing the car data (graph) and the defined rule (rule1). The reason method is called on the reasoner object (line 16), initiating the reasoning process where the rule is evaluated against the provided data, and if new knowledge is inferred, the fault function is called.

Our rule-based framework shows the additional benefits of having a KG to steer this whole process. Without having to define additional rule-specific context, our framework and rules can be applied to multiple stations. New stations providing data linked to the `metricTempValue` can possibly trigger the callback function. This makes dynamically extending the KG and using this rule-based framework even more interesting.

Listing 1.1. Example code to detect possible faults on a car body using our Python rule-based reasoner.

```
from reasoner import *
from cars.query_car import query

# Query car data for analysis
graph = query(3076989)

# Define a rule for detecting faults in cars
rule1 = Rule(
    'car(?x) ^ hasLog(?x, ?y) ^ metricTempValue(?y, ?v) ^
       greaterThan(?v, 18) -> sparks:Fault(?x, ?v)',
    'SELECT ?x ?y WHERE {?x sparks:Fault ?y}',
    lambda res: fault(res))

# Define an action function for the fault rule
def fault(res):
    print(f'FAULT: car {str(res[0]).split(".")[-1]} is faulty
         due to a too high VP413P04BF1 value of {res[1]}')

# Initialize the rule reasoner with the graph and rules
reasoner = RuleReasoner(graph, [rule1])

# Trigger the reasoning process
reasoner.reason()
```

3.2 Semantically Enriched MLOps Pipeline

Besides knowledge-driven AI, data-driven solutions, e.g. machine learning (ML), emerge frequently in the industry 4.0 domain, as they can use the available sensor data to search for unknown patterns or unwanted system behaviour automatically [2]. MLOps, the integration of DevOps principles within ML workflows, revolutionizes this digital transformation [6]. It streamlines the deployment, monitoring, and optimization of ML models, ensuring seamless integration into production environments. A typical ML pipeline involves data ingestion, preprocessing, model training, evaluation, and deployment [5]. MLOps separates these stages, facilitating the reuse of components and iterative improvements, thereby making it possible for industries to leverage ML for the dynamic purposes of Industry 4.0.

Despite the advancements in ML and data science techniques, it remains evident that considerable efforts are still required to ensure the acquisition of correct and high-quality data from within databases. This initial step is fundamental, as the effectiveness of subsequent feature extraction and ML methods heavily relies on the quality and relevance of the available data. Within our paintshop, this first step was extremely challenging due to the missing link between sensor metadata and the provided data. The generated KG resolved this mismatch and data from specific observable properties can now be easily queried using multiple SPARQL queries. The SPARQL Query in e.g. Listing 1.2 can be used to acquire all table, signal and column information from a specific Sensor Property Service concept for all systems within a given station. With this information, a generic component within the ML pipeline can be provided to acquire all the necessary data over different stations for that particular property-sensor combination. Decoupling the ML pipeline from the underlying data structure has the additional benefit of introducing fewer faults when this data structure changes. When e.g. moving to a new database, only the Sensor Property Services and connectors will have to be rewritten, but the links towards the stations remain the same.

Listing 1.2. Example SPARQL query to acquire the Sensor Property Service data properties for a particular station

```
SELECT ?schema ?table ?signal WHERE
{
    ?service a sparks:SnowflakeSensorPropertyService .
    ?service  sparks:hasSensor ?sensor .
    <B433> ssn:hasSubSystem* ?sensor .
    ?service sparks:hasSchema ?schema .
    ?service sparks:hasTable ?table .
    ?service sparks:hasColumn ?signal .
}
```

While data acquisition is one such process within the ML pipeline that clearly benefits from a KG, the proceeding feature engineering and ML training steps can also be linked to the Sensor Property Services and process information within the KG [16]. Such a semantic pipeline could enable ML engineers to easily create, adapt or even directly apply different ML models to different stations in different factories by only specifying and reasoning upon which station identifiers to use within the KG.

3.3 Dynamic Dashboard

In the modern era of data analytics, dashboards have emerged as necessary tools for gaining insights, monitoring systems, and facilitating informed actions [9]. Dashboards provide so-called widgets, visual interfaces that consolidate complex data into interpretable visualizations, enabling operators to derive actionable insights. However, traditional dashboard creation methods often require significant manual effort and expertise in how the data is outputted, posing challenges in adapting to evolving data landscapes and dynamic operational needs. Therefore, in previous research, we created the Dynamic Dashboard, a dashboard solution designed to automate the process of dashboard creation by leveraging Semantic Web technologies [22].

The constructed KG provides hierarchical information, grouping sensors and subsystems as indicated in Fig. 5. A link to the data is also provided using the Sensor Property Service. The Dynamic Dashboard uses these concepts and instances of both the ontology and KG and further extends this semantic knowledge in two parts.

First, for each observable property within the KG, additional metric information is added. This metric information includes details about how the data from that observable property is provided, e.g. the data type and unit of measurement defined by the Ontology of units of Measure (OM) [14]. An example of such a metric is provided in Listing 1.3.

Listing 1.3. Example Code of metric information

```
metrics:quantity a dashb:Metric; a om:Quantity;
        dashb:datatype xsd:double .
```

The different available widgets to visualize the data are also semantically described. Each widget has a component list that defines how many sources it accepts, but also which metrics it can visualize. An example of such a widget description is provided in Listing 1.4.

Listing 1.4. Example Code of widget description

```
<time-series-line-chart> a dashb:LineChart;
    rdfs:label "Line chart"@en;
    dashb:component [
        dashb:accepts [ dashb:datatype xsd:double ];
        dashb:min 1;
        dashb:max 1 ].
```

The above combination of metric information and widget description enables the Dynamic Dashboard to reason upon which visualizations are valid and interesting for a provided observable property. More information about this reasoning procedure and the overall architecture of the Dynamic Dashboard can be found in the corresponding research papers [12,22]. Within the paintshop, the Dynamic Dashboard offered the operators two distinct functionalities for visualizing insights: user-driven and event-driven dashboards [12].

In the user-driven dashboards, operators can create visualizations tailored to their specific needs and preferences. Through an intuitive interface, operators can explore the available sensors and systems available within the KG and can select the properties for visualization. As discussed, semantic reasoning suggests suitable visualizations based on the selected observable properties and registered metrics, streamlining the process of dashboard creation.

The event-driven dashboard, on the other hand, automatically generates dashboards in response to detected anomalies or events present in the data. Analogously to observable properties and widgets, semantic descriptions can be provided for events. These events can originate from a wide range of applications. Within this paintshop setting, these events include the outcome of the

Listing 1.5. Example Code of detected anomaly event

```
<events/1234> a dashb:Event;
dcterms:description "4271656 is predicted to be faulty";
sosa:resultTime "2023-09-05T20:28:11"^^xsd:dateTime ;
ssn:wasOriginatedBy [
    fromObservation [
    sosa:resultTime "2023-09-05T20:28:11"^^xsd:dateTime];
    toObservation [
    sosa:resultTime "2023-09-05T20:28:11"^^xsd:dateTime];
    observedProperty </body_visualization/195/83>].
```

rule-based module or the outcome of a ML model as described in Sects. 3.1 and 3.2. By providing a semantic description of the detected faults or detected quality issues, the Dynamic Dashboard is able to recommend the necessary visualization for further inspection by the operators. An example semantic description for a detected anomalous event is provided in Listing 1.5. In addition to the observable properties, the ranges of when the fault occurred are also defined in these events.

Based on this description, the correct widget is automatically created to visualize the anomaly. As shown in Fig. 7, the dashboard for this example event concerns a ML detected quality issue. It displays the general information about the event (right) and provides a visualization to display the faults on a car diagram (left). The red dots shown in the visualization are indications of where the detected faults occurred on the car.

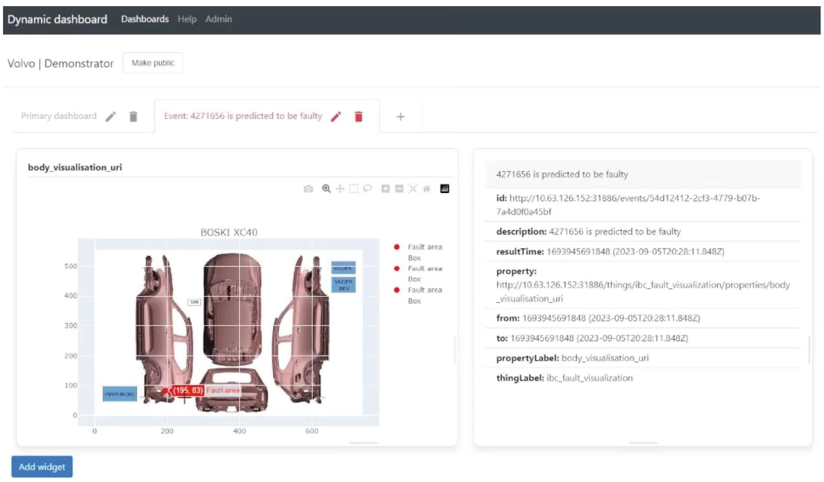

Fig. 7. Expert-driven Dynamic Dashboard example. This example shows a potential predicted quality issue detected by a ML algorithm.

The fact that the Dynamic Dashboard provides this dual functionality enhances situational awareness, enabling both proactive monitoring through user-driven customization and reactive response to system anomalies via event-driven analysis.

4 Impact and Lessons Learned

The impact of this work on the transition of manufacturing operations from traditional knowledge management approaches to a knowledge graph-based system is multifaceted. The primary influences lie in the realm of efficiency, easy transferability to other sites or processes, and traceability. By devising a solution that integrates data, metadata information, and process insights into a KG, this work effectively tackled the uniformity challenges prevalent in manufacturing processes. A key aspect of this is the establishment of unique identifiers and the incorporation of expert knowledge using ontologies.

This work also provided ways to foster collaboration and consensus-building among domain experts. By developing upper ontologies, such as the SPARKS ontology, and tools to easily extend, adapt and maintain available expert knowledge, this work shows the benefits of designing such frameworks for domain experts to collectively understand and optimize manufacturing processes. Those responsible for paint application quality were often unaware of the production line issues. The proposed collaborative approach eventually led to more effective QC measures.

Through the standardization of identification systems across different systems, the data from diverse sources became operable for multiple applications. Within the paintshop pipeline, this provided operators with Dynamic Dashboards and rule-based fault indicator methods. Our rule-based system was developed as a proof of concept to demonstrate the advantages of using reasoning for root cause analyses. The Volvo Cars Group is conducting a cost-benefit analysis to determine the need for more advanced reasoning systems like RDFox[4] or Stardog[5] for more advanced functionalities.

Additionally, the creation of anomaly detection pipelines based on ML could be more automated. This resulted in the applicability of these models for a wide range of stations and driving the continuous improvement initiatives within evolving production environments. Once a unique identifier for new equipment is defined, it can be easily added and linked to the existing KG using the SPARKS GUI. This demonstrates that our approach can grow and adapt to new requirements without significant redevelopment. The integration of the KG within the MLOps pipeline further enhances the system's flexibility, potentially reducing long-term maintenance costs.

In essence, this work can be seen as a prelude to the use of manufacturing intelligence, where the seamless integration of knowledge and data results in innovation, enhances competitiveness and drives the industry to new heights of

[4] https://www.oxfordsemantic.tech/.
[5] https://www.stardog.com/.

productivity. This is also indicated by how the feedback through the Dynamic Dashboard operations could further influence e.g. the ML behaviour. The MLOps use case demonstrated how the Volvo Cars ML group could use the KG to access relevant station data. This enabled them to develop ML pipelines that are easily transferable to other, similar stations, as the data locations for sensors and actuators in these stations were also queryable through the KG. Future work could use the KG and the available data to work towards KG embeddings and enhance the ML models themselves based on the provided domain-specific knowledge for tasks like anomaly detection [20]. This offers a future where manufacturing processes are not just explainable for the operator, but also adaptive to new unknown situations [17].

5 Conclusion

In conclusion, this work exemplifies the transformative potential of KGs and semantic technologies in automating manufacturing operations within an automotive paintshop. By standardizing identification systems, facilitating collaboration among domain experts, and enabling seamless data integration, the showed applications not only address current challenges but also lay the groundwork for future advancements in Industry 4.0. The lessons and impact of this research serve as valuable insights for the continuous adaptation and improvement of various manufacturing processes.

Supplemental Material Statement: The full Volvo Cars KG, station-specific PLC code and QC-related issues cannot be made available as they incorporate company-specific data. However, small examples (including an example KG) are made available, for the sole purpose of replicating the applications in this paper on a GitHub at https://github.com/predict-idlab/SPARKS. The SPARKS ontology and the source code for both the GUI editor and Python rule-based framework are available on this same GitHub repository.

Acknowledgement. This work was executed as part of the O&O HBC.2020.3072 project, funded by Volvo & VLAIO.

References

1. Akafuah, N.K., Poozesh, S., Salaimeh, A., Patrick, G., Lawler, K., Saito, K.: Evolution of the automotive body coating process-a review. Coatings **6**(2), 24 (2016)
2. Bousdekis, A., Lepenioti, K., Apostolou, D., Mentzas, G.: A review of data-driven decision-making methods for industry 4.0 maintenance applications. Electronics **10**(7), 828 (2021)
3. Compton, M., et al.: The SSN ontology of the W3C semantic sensor network incubator group. J. Web Semant. **17**, 25–32 (2012)
4. Dageville, B., et al.: The snowflake elastic data warehouse. In: Proceedings of the 2016 International Conference on Management of Data, pp. 215–226 (2016)

5. Elshawi, R., Maher, M., Sakr, S.: Automated machine learning: state-of-the-art and open challenges. arXiv preprint arXiv:1906.02287 (2019)
6. Francois Regis, D.: MLOps paradigm-a game changer in machine learning engineering? (2023)
7. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing? Int. J. Hum Comput Stud. **43**(5–6), 907–928 (1995)
8. Horrocks, I., et al.: SWRL: a semantic web rule language combining OWL and RuleML. W3C Member submission **21**(79), 1–31 (2004)
9. Kumar, N., Prajapati, S.: Challenges for interface designers in designing sensor dashboards in the context of industry 4.0. Int. J. Ind. Manuf. Eng. **13**(8), 539–542 (2019)
10. Lamy, J.B.: Owlready: ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies. Artif. Intell. Med. **80**, 11–28 (2017)
11. Margherita, E.G., Bua, I.: The role of human resource practices for the development of operator 4.0 in industry 4.0 organisations: a literature review and a research agenda. Businesses **1**(1), 18–33 (2021)
12. Moens, P., Vanden Hautte, S., De Paepe, D., Steenwinckel, B.: Event-driven dashboarding and feedback for improved event detection in predictive maintenance applications. Appl. Sci. **11**(21), 10371 (2021)
13. Raja Santhi, A., Muthuswamy, P.: Industry 5.0 or industry 4.0? Introduction to industry 4.0 and a peek into the prospective industry 5.0 technologies. IJIDeM **17**(2), 947–979 (2023)
14. Rijgersberg, H., Van Assem, M., Top, J.: Ontology of units of measure and related concepts. Semant. Web **4**(1), 3–13 (2013)
15. Simperl, E., Luczak-Rösch, M.: Collaborative ontology engineering: a survey. Knowl. Eng. Rev. **29**(1), 101–131 (2014)
16. Soete, C., Steenwinckel, B., Mussche, J., Ongenae, F., Van Hoecke, S.: Enabling ontologies for semi-automated feature extraction and traceability of quality issues in production line monitoring. Information Fusion **under review** (2024)
17. Steenwinckel, B., De Paepe, D., Hautte, S.V., Heyvaert, et al.: Flags: a methodology for adaptive anomaly detection and root cause analysis on sensor data streams by fusing expert knowledge with machine learning. Future Gener. Comput. Syst. **116**, 30–48 (2021)
18. Steenwinckel, B., Heyvaert, P., De Paepe, D., et al.: Towards adaptive anomaly detection and root cause analysis by automated extraction of knowledge from risk analyses. In: SSN@ ISWC, pp. 17–31 (2018)
19. Steenwinckel, B., Moens, P., Mussche, J., Van Hoecke, S., Ongenae, F.: Systematized procedure for automating the retrieval of knowledge within smart industries. IEEE Trans. Ind. Inform. **under review** (2024)
20. Steenwinckel, B., Vandewiele, G., Weyns, M., Agozzino, T., Turck, F.D., Ongenae, F.: Ink: knowledge graph embeddings for node classification. Data Min. Knowl. Disc. **36**(2), 620–667 (2022). https://doi.org/10.1007/s10618-021-00806-z
21. Tudorache, T.: Ontology engineering: current state, challenges, and future directions. Semant. Web **11**(1), 125–138 (2020)
22. Vanden Hautte, S., Moens, P., Van Herwegen, J., et al.: A dynamic dashboarding application for fleet monitoring using semantic web of things technologies. Sensors **20**(4), 1152 (2020)