Semantic Checking of Different Type Information Sources About Permitted Speeds in Railway Transport

Viktor Shynkarenko and Larysa Zhuchyi

Ukrainian State University of Science and Technologies
2, Lazarian str., Dnipro, Ukraine

Abstract
The infrastructure of railway stations must ensure a high level of safety when trains move at the declared speeds. The operation of various infrastructure elements is carried out in accordance with the normative and technical regulations of the railways, based on their current state. Information about this is stored in electronic documents and databases of various types. Means are proposed to improve the safety of train traffic based on the semantic checking of data from various sources about the permitted speeds on the elements of railway tracks. To formalize the restrictions of the technical operation rules of Ukrainian railways, it is proposed to use the method of semantic annotation. The ontology is formed based on the composition of relations. The methods of conceptualization of the tabular representation of knowledge and multi-level concretization proposed earlier are applied. A modular ontology has been developed for integrating the data of the railway switch and track lists, orders that set the permitted speeds on the elements of the railway infrastructure, technical operation rules and building norms. This approach provides a connection between natural language regulations, information systems and ontologies on the issues of a train speed restriction. Heterogeneous and diverse data sources harmonization will increase the level of their accuracy and, as a result, the reliability of the corresponding subsystems of the railway transport operation.

Keywords
Ontology, permanent speed restriction, railway, natural-language regulations, tabular data, semantic checking, concept

1. Introduction

This paper explores the possibilities of semantic annotation of railway transport legal regulations. The checking of permitted train speeds is carried out based on the integration of data from the lists of switches and tracks, orders establishing permitted speeds on elements of the railway infrastructure, technical operation rules and building norms. The approach is to convert part of the restrictions of legal regulations not into an ontology schema (such as owl class restrictions) but in an annotation in tsv format (and then in RDF format). This allows the subsequent integration of these regulations and track lists and consistency checking. Formalized document annotations provide a link between regulation texts and ontologies.

2. Problem statement and purpose

The permitted speed of a section of a railway track includes many factors, such as rail type of track and the frog type of the switches and their condition. Data on the characteristics of infrastructure

elements and speeds are stored in sources of various formats: drawings, databases, MS documents word.

Compliance checking of the speed and rail type of track is carried out in accordance with the legal regulations, presented in the form of non-formalized restrictions of natural language texts.

The purpose of the work is to improve safety on the railway by linking the railway track speed values of the order establishing permitted speeds on the elements of the railway infrastructure with the characteristics of the track and ensuring the consistency of these tables with formalized restrictions of the technical exploitation rules (TER) and state building norms (SBN) using ontological means.

3. Related works

Much attention is paid to transport ontologies [1-9], as well as the formalization of regulatory documents, for example, the European Union EUR- Lex database [10]. Ontologies allow one to represent constraints as axioms, and data as triples to perform semantic checking. To populate ontologies, automated methods of data extraction from their tabular representation and natural language texts are often used.

In the transport domain, a semantic dataset [11] has been developed by extracting and integrating data on the suitability of transport infrastructure for people with disabilities from various sources to enrich public transport data with it. The Mobility and Accessibility Ontology (MAnto) is based on such models as Transmodel [12] and IFOPT [13]. Structured texts like GTFS data descriptions are annotated to map them onto the MAnto ontology.

Annotation of natural language texts can be performed in the ontology editor Protégé [14] and outside it [15]. The INCEpTION functionality includes means for annotation using ontologies with the entity linking method and tags using the semantic role labelling method. For annotation, we use the INCEpTION web service [16] and the semantic role labelling method.

Some developments use tagsets to perform text annotation using the semantic role labelling method with ontologies concepts, for example [17]. As part of the S-CASE project (Scaffolding Scalable Software Services), software requirements are annotated [17] to check them for consistency. Tools have been developed, the architecture of which includes a module for "translating" requirements into software specifications. Annotation is performed with methods such as feature-based extraction, dependency parsing and semantic role labelling in mate tools and is done automatically. Actor-Action-Object triples can be marked manually or by the parser.

3.1. Entity linking

Entity linking is an annotation method that allows one to assign named entities from the text (mostly proper names) URI of ontology individuals according to their context. The method has become widespread in the domains of biological and historical sciences. Entity linker output is a JSON table that is used, for example, to separate proper names in the text and not search for syntax errors in them, as in [19], where scispacy entity linker is used [20]. Python library scispacy is based on the ontologies such as Human Phenotype Ontology [21], Gene Ontology [22], RxNorm [23], Medical Subject Headings thesaurus [24] and Unified Medical Language System dictionaries [25].

In [26], ontologies of the Irish cultural heritage and Cultural Heritage entity Linker (CHEL) were developed. Ontologies are populated with individuals from The Statute Staple, The Down Survey, The Books of Survey and Distribution, Dictionary of Irish Biography and the Oxford Dictionary of National Biography. A feature of CHEL is the generation of a Globally Unique Identifier for identical instances of different ontologies. Digitized cultural heritage data are used in digital library and museum data integration [27], as in the case of the Europeana project [28].

3.2. Manual data annotation

Manual annotation is actively used in tasks of biological domain DNA sequencing for two reasons: greater reliability of manual annotations [29] and the inability to annotate some genes (for example,

pseudogenes) in automatic mode. The DNA of a biofuel-producing bacterium is manually annotated in [29], where International Protein Nomenclature developed by the National Center for Biotechnology Information is used for naming, as well as bug fixes of various automatic annotation systems are done. Gene Ontology contains millions of manual and automatic annotations [30], is not just a vocabulary, but contains class logical definitions [22], and has been reused in many ontological developments.

Annotation of software requirements is performed to increase the efficiency of its testing [31, 32]. Their own markup language has been developed for annotating software requirements. Software testing is done through simulation. Annotation allows one to develop a model for associating software requirements represented as natural language text and simulation signals. Mapping of signal names and requirement parts is done manually. Annotations have several levels of detail. Annotation is done manually because the engineer must choose the level of detail and, for example, determine under what conditions the car's turn signal should turn on.

3.3. Semantic Annotation of Legal Documents

A domain-oriented software package was developed for automatic annotation of regulatory documents [33], which can export files in RDF format and allows one to execute SPARQL queries on text documents in the construction domain. The task is relevant because the verification of models for compliance with regulations is performed manually. To perform annotations, an ontology was developed that reuses concepts from the dc [34], doco [35], lemon [36] ontologies.

Modular Financial Industry Regulatory Ontology [37] was developed to annotate the financial regulatory documents of the Anti-Money Laundering domain to check business processes for consistency. Document annotation is done automatically using machine learning to populate the ontology with instances. The training data is annotated manually in the GATE system. The software package allows one to execute SPARQL queries on text documents to search for relevant restrictions, obligations, prohibitions, etc.

In [38], a domain-specific system was developed for recognizing named entities of legislative acts to perform intellectual analysis and integration of the national and European legal framework. For annotation, the Inter-Active Terminology vocabulary is used for Europe and the Wikipedia knowledge base.

4. Ontology development methods

In the work, the ontology is populated with instances from the INCEpTION output files [16], obtained as a result of annotating the railway legal documents. Annotation is the process of changing the text by adding metadata (tags, ontology concepts) to it.

Structure validation and data transformation of drawings are performed based on the tabular knowledge representation model [39], the modular approach of the ontology development and the method of separating rules and data are carried out according to the integration of the railway information systems ontology framework [40].

The linking of railway switch and track list tables is performed to check the consistency of track names and railway switch frog type. Semantic checking is carried out on the example of the TER restriction stating that only railway switches of the frog type of 1/11 should be installed on the main tracks.

The linking of the tables of the track list and the order establishing permitted speeds on the elements of the railway infrastructure is performed to check the consistency of the rail type of track and the speed of the railway line section. Semantic checking is performed on the example of the SBN restriction on the correspondence of the maximum speed of 120 km/h on the line track of the P65 rail type.

The ontology is developed in the Protégé ontology editor, the vocabulary – in MRcube, drawing tables data extraction – in Tabula, data wrangling – in OpenRefine.

5. Ontology formation

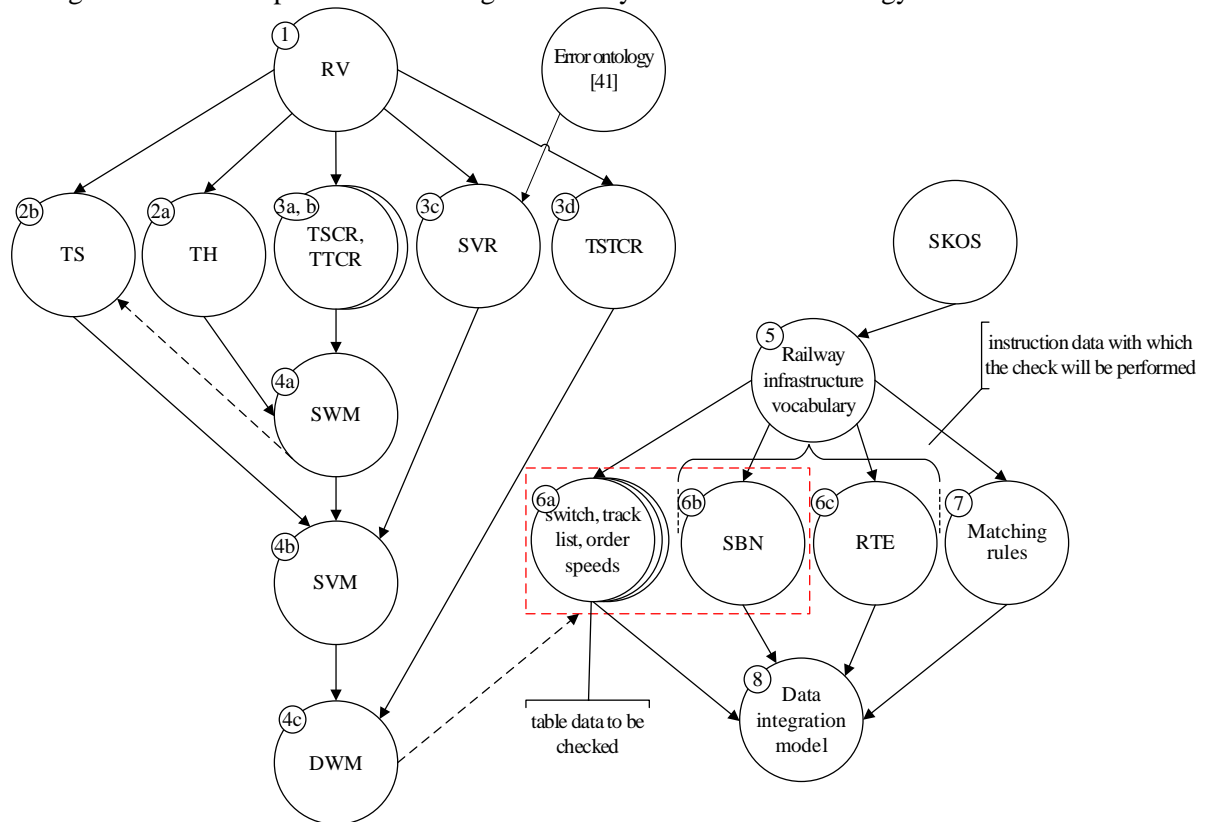Figure 1 shows the process of forming the railway infrastructure ontology.

Figure 1: Modular ontology of railway track characteristics

In Figure 1:
RV – Resources Vocabulary
TH – Table Header
TS – Table Structure
TSCR – Table Software Classification Rules
TTCR – Table Type Classification Rules
SVR – Structure Validation Rules
TSTCR – Table Station Classification Rules
SWM – Structure Wrangling Model
SVM – Structure Validation Model
DWM – Data Wrangling Model
The ontology is formed in the following sequence:

• common vocabularies (ontologies 1, 5 Figure 1) are developed to describe the structure and data of the tables of the track, railway switch lists, order speeds and the table of SBN «Capacity of the upper structure of the main tracks in the design of new railway lines» in the ontology of the abstract model of sources and the ontology of the abstract model of the railway infrastructure;

• rules are formalized for validating the table structure (ontology 3c), classification (properties of the table that it should have for the reasoner to classify it into the class) of tables by type, program (ontologies 3a, b) and station (ontology 3d) in the ontology of a concrete source model that includes rules;

• the header of the tables is described in the ontology of a concrete resources model that includes data (ontology 2a)

• instances of the table header (ontology 2a) and rules for classifying by type and program (ontologies 3a, b) are imported into the ontology of a concrete resources model of the second level (ontology 4a), and the instances are classified into classes like «track list table» by the reasoner;

- in annotations to classes, OpenRefine scripts are obtained to transform the structure of tables into ontology individuals;
- table files with a weak structure (converted from AutoCAD, MS Word to pdf format) go through the data extraction procedure in Tabula (transformation to csv);
- instances generated by OpenRefine, as part of a concrete data source model ontology (ontology 2b), are imported into a second-level concrete resources model ontology (ontology 4b), along with data validation rules (ontology 3c). If the ontology is consistent, one goes to the next step, otherwise, errors are corrected;
- the rules for classifying tables by station (ontology 3d) are imported into the ontology (ontology 4c). Linking of tables related to the same station is performed, and the tables are classified into classes like «abstract station track record» by the reasoner;
- scripts for data transformation are got in annotations of classes;
- in OpenRefine one generates the first part of the ontology of a concrete railway infrastructure model that includes data (ontologies 6a);
- classes and relations of the railway vocabulary ontology (ontology 5) are exported from Protégé in csv format and converted in OpenRefine to INCEpTION tagset in JSON format;
- regulation text is annotated using the semantic role labelling method with tagsets in INCEpTION. Annotated text is exported to a tsv file;
- tsv files are converted to railway infrastructure ontology instances in OpenRefine so that the second part of the ontology of a concrete railway infrastructure model that includes data (ontology 6c) is generated;
- the matching rules formalized in the ontology of a concrete model of railway infrastructure that includes rules (ontology 7);
- instances of the OpenRefine data of the tables of the track list, the list of switches, the table of the order establishing permitted speeds (ontology 6a), SBN tables of the capacity of the superstructure of the main tracks (ontology 6b), the output file of the TER annotation tables (ontology 6c), the rules for linking tables and checking data consistency (ontology 7) are imported into the ontology of a concrete railway infrastructure model of the second level (ontology 8). The consistency of these tables and restrictions to TER and SBN is checked.

6. Checking of factors affecting the speed of the track

Consider the implementation of the ontology of data sources and the ontology of the railway infrastructure.

6.1. Data resources ontology

The abstract source model is developed as a vocabulary for describing the structure of the switch list and the track list tables, the table of the order establishing permitted speeds, the power of the superstructure of the main tracks SBN table, the table of the RTE annotation output file and includes the names of these tables, the names of columns and station attributes.

The population of **ontologies of a concrete data source model** with instances is partially automated by generating instances in OpenRefine, where empty cells are populated with literals like "error" and a station attribute is retrieved from the table with station owners.

A concrete source model with rules is developed in the form of separate ontologies for the classification and validation of the table structure rules. The tables are classified by program, type, and station, so the hierarchy is organized using logical definitions (Figure 2):

The station attribute is retrieved from the station owner drawing table at the table structure transformation step and is associated with all the station tables with a property chain.

Validation is performed using SWRL rules [41] for cells whose content has been replaced with the "error" literal in OpenRefine, in the table structure transformation step.

In the ontology of **a concrete source model of the second level**, tables are validated before being checked for compliance with legal documents and their structure and data are converted into ontology instances.

Let's consider an example of the resources concrete model of the second level on the table of the track list. In the beginning, rules are imported into the ontology for classifying tables by type and program. The table contains columns with sections of the railway line, tracks and speeds and is made in the database, therefore it is classified into the «track list table» class.
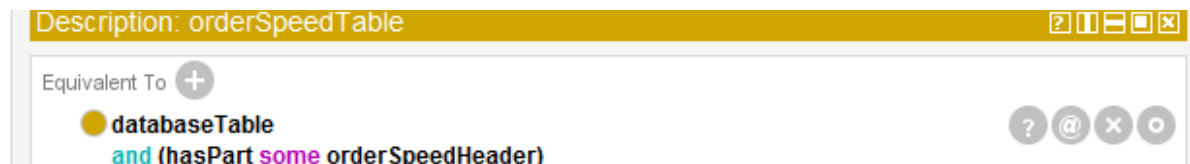
Figure 2: The logical definition of the permitted speed order table

In the annotations, the OpenRefine script is got to convert the structure to RDF triples. Instances of «table», «tuple», «value» etc. are generated and are interconnected by the relations like «has part» and «has element».

6.2. Railway infrastructure ontology

The railway infrastructure abstract model is a vocabulary that integrates concepts of railway tracks, railway switches and orders establishing permitted speeds, technical operation rules and building norms.

The abstract model also contains the relation «word id corresponds to the individual» to process the table received from INCEpTION.

Relationships in INCEpTION are not directed. The direction is indicated by the id of the subjects and objects of the relationship. Relations like «frog type SBN corresponds to track name» and «track name SBN corresponds to frog type» are related by the relationship «inverse of» in the ontology containing rules.

Instances (frog type and track name) are associated in two contexts (regulation and list). Two different relationships are developed: «drawing corresponds to», «frog type SBN corresponds to track name» for the triple generated from the list, and the triple generated from the regulations.

A concrete model of the railway infrastructure with data is developed in the form of five ontologies:

- the track list table ontology;
- the railway switches list table ontology;
- the table of the order establishing permitted speeds ontology;
- the power of the upper structure of the track table ontology;
- the INCEpTION output file table is generated as a result of TER annotation ontology (the annotation process is shown in Figure 3).
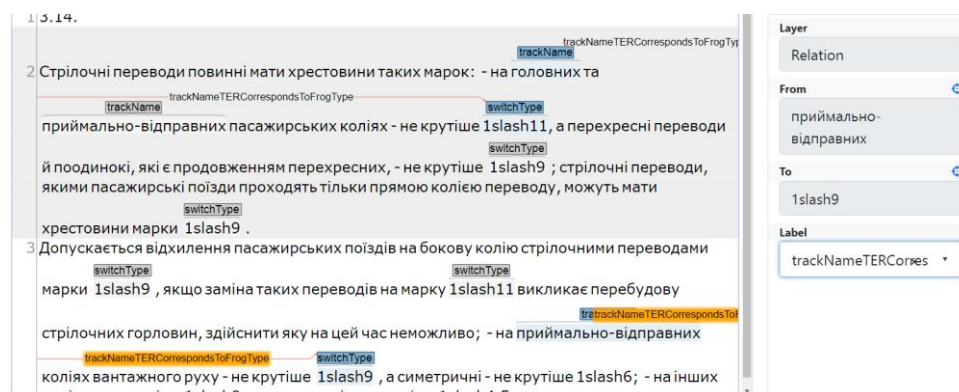

Figure 3: Annotation of RTE with railway infrastructure ontology concepts in INCEpTION

URI for the name of the track and frog type is chosen as their actual names like «main» and «1/11». Instead of «1/11» names like «1slash11» are used, because INCEpTION interprets «1/11» as three separate words, and «1slash11» as one. Therefore, after extracting data from the text and tables, each track name and frog type must be associated with two relationships: for example «frog type TER corresponds to track name» and «list corresponds to» for railway switches frog type and rail type.

A feature of the table of the output file INCEpTION are also cases and plurals of concepts like «main» instead of «main». OpenRefine searches for all values containing strings of type «гол» and replaces the values with «main».

A concrete model of the railway infrastructure with rules is developed in the form of an ontology, including compositions of relations for linking railway switch and track lists tables, compositions of relations for processing annotations, and restrictions for checking the compliance of the track list and the order speeds tables data with the rules of SBN and TER.

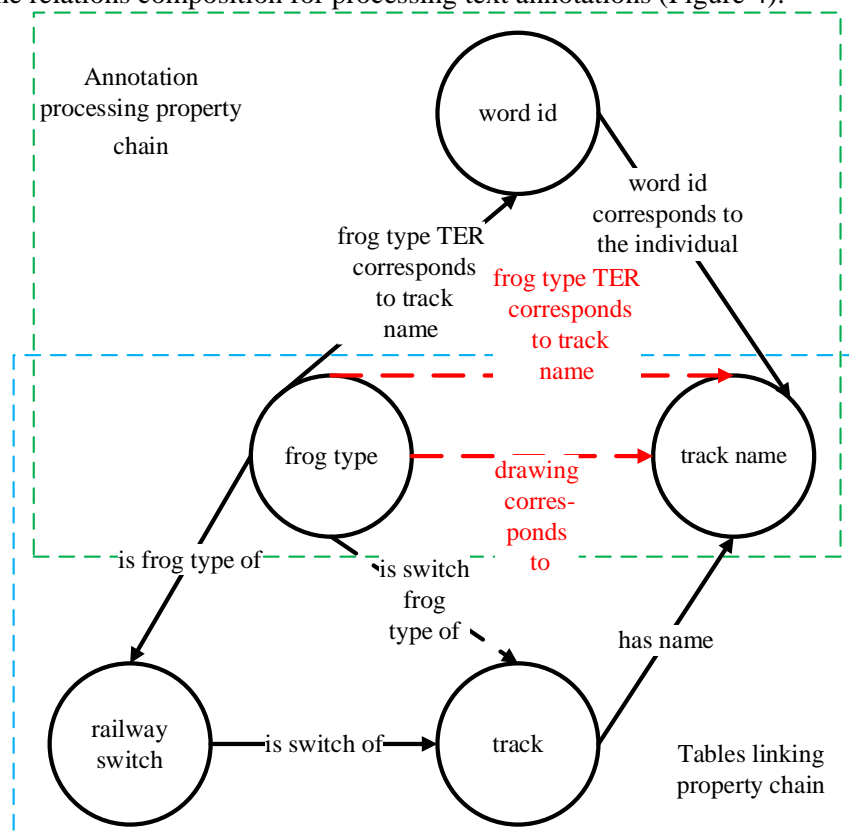Consider the relations composition for processing text annotations (Figure 4).

Figure 4: Relationship compositions for data linking of tables of tracks and railway switches

The red color in Figure 4 represents the relation that connects the same instances as the relation «list corresponding to» in the track list tables.

In the INCEpTION output file table, words are associated not with words, but with ids of other words, for example, «3-34 587-608 siding track name trackNameTERCorrespondsToFrogType 3-41». Linking of concept to the concept is done using relation compositions like in Figure 5.

Let us consider relations compositions linking railway switch list and railway track list tables (Figure 4) utilized to check whether the frog type of the switch corresponds to the name of the railway track. Relations compositions are used to link the railway switch list and the track list with compositions in such a way as to obtain a TER regulation fact, that is, to link the name of the track with the frog type of the railway switch.

To link the values of the tracks and switches lists tables, the following operations are performed:

- linking the station railway track and the railway switch frog type by the relation «is switch frog type» (Figure 6);

- linking the name of the railway station track and the railway switch frog type with the relation «list corresponds to» (Figure 7).
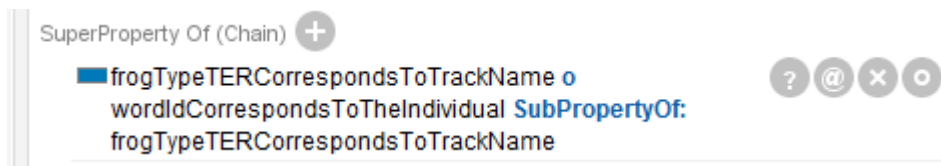

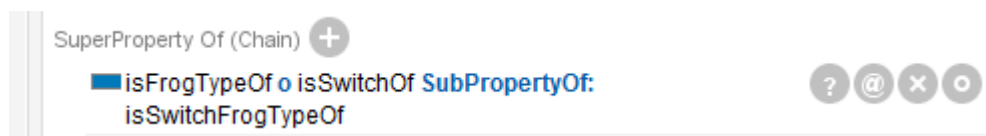**Figure 5:** Relations composition for processing of the railway regulation text annotations

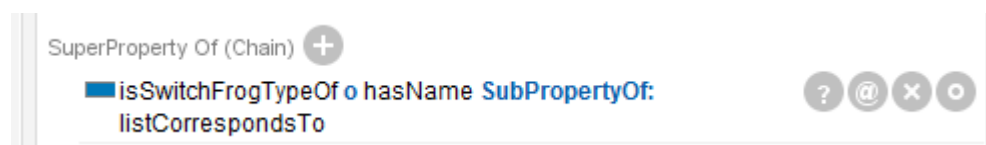**Figure 6:** Relations composition for linking railway track and railway switch lists tables

Figure 7: Composition of relations to obtain TER triple from track lists

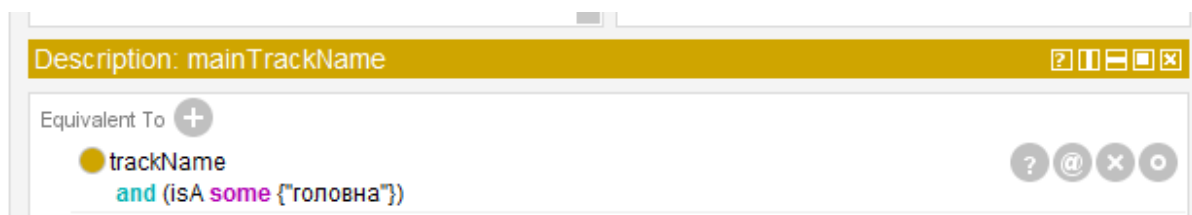The classification of track names is done by the reasoner according to axioms like Figure 8.

Figure 8: Logical definitions of track names

The classification and checking of the railway switch frog types are carried out by the reasoner according to the axioms like the Figure 9.
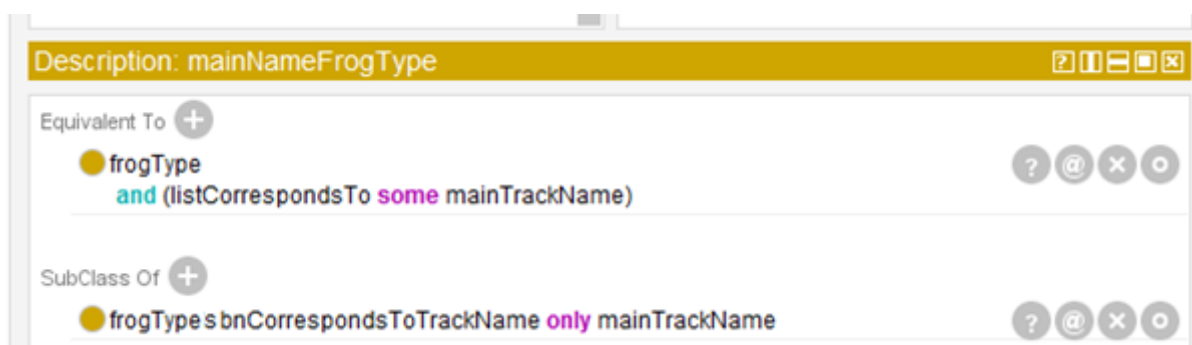

Figure 9: Logical definition and restrictions of the railway switch frog type of the station main track

Compositions of relations for linking the track list tables and the order speed table (Figure 10) are used to check the correspondence between the rail type and the speed and to link the track list and the order speed table to obtain an SBN regulation fact, that is, link the railway track rail type with the speed of the line section.

The difference between the speed and the frog type values is that the frog type is a discrete value, and the speed is continuous. Permitted speed intervals correspond to different rail types and checking

is performed when the order speed is equal to the extreme value of the interval (red arrows in Figure 10) and the intermediate one (green ones in Figure 10).
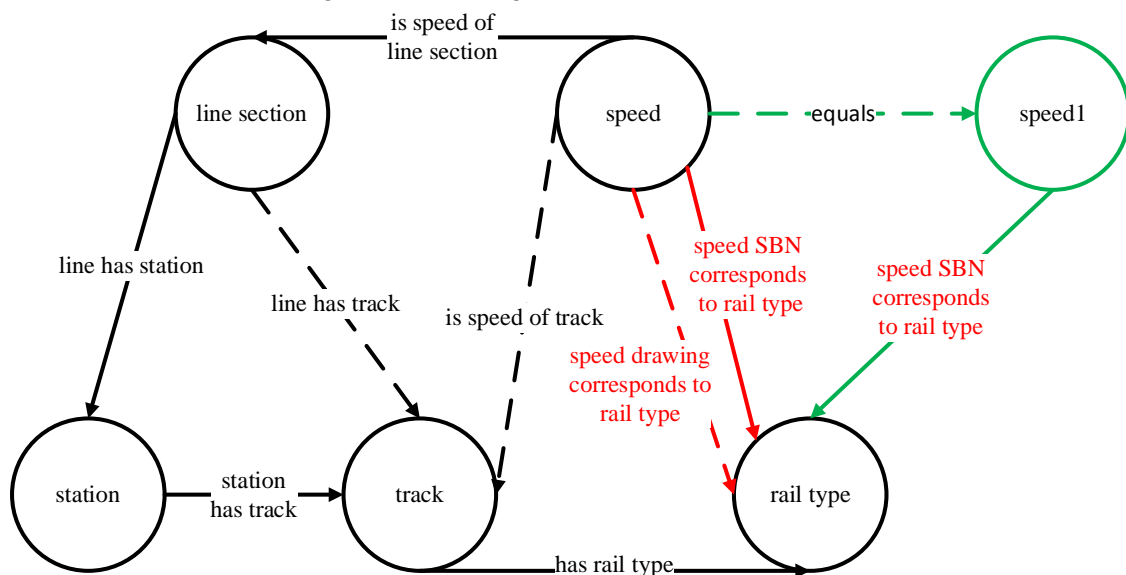

Figure 10: Compositions of relations for linking the list of tracks and the order of permitted speeds

Let us consider the case of equality of the speed of the order to the extreme value of the interval.

To link the values of the tracks list and the order establishing permitted speeds tables, the following operations are sequentially performed:

- linking the station track and the line section by the relation «line has track» (Figure 11);
- linking the track and railway line speed by the relation «is speed of track» (Figure 12);
- linking the rail type of the track and the railway line speed by the relation «speed drawing corresponding to rail type» (Figure 13).
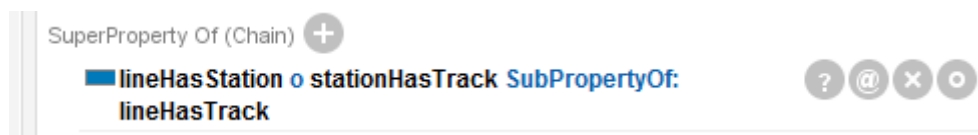

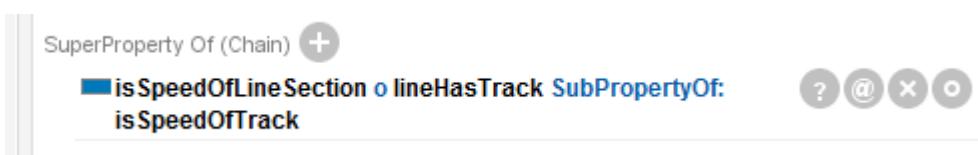**Figure 11:** Relations composition for determining whether a station track belongs to the line section.

Figure 12: Relations composition for linking track and railway line section speed
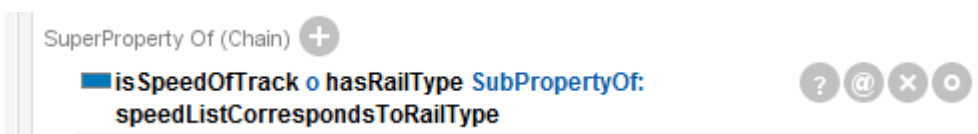

Figure 13: Relations composition for linking rail type and railway line section speed

The classification of track names is done by the reasoner according to axioms like Figure 14.

Classification and speed checking are performed according to axioms like Figure 15.

Let us consider the case of equality of the speed of the permitted speed order to the intermediate value of the gap. Since the speeds are not equal, they are related by the «equals» relations composition of Figure 16.
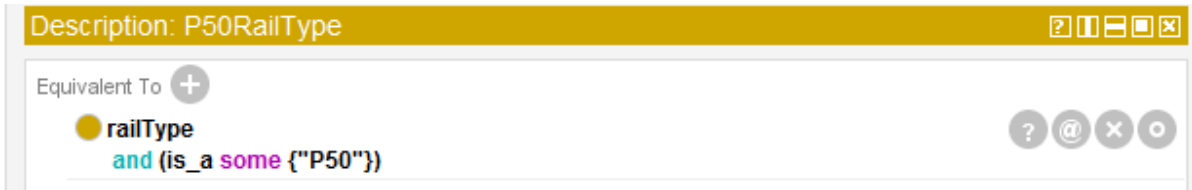
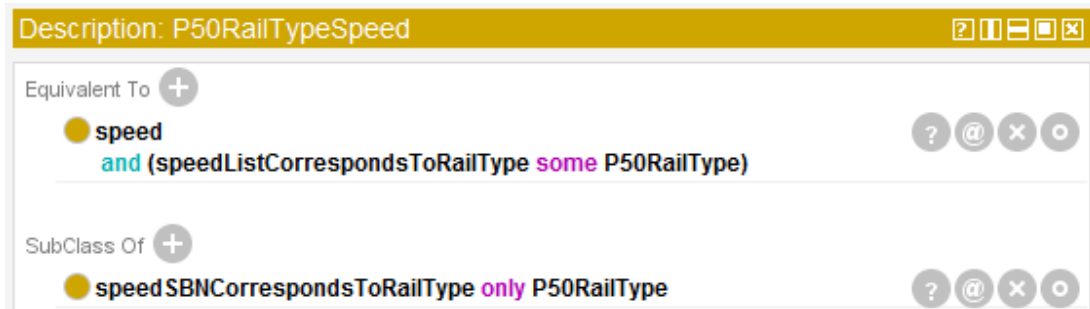**Figure 14:** Logical definitions for classifying rail types

Figure 15: Logical definition and restriction of the permitted speed of the P50 rail type track class
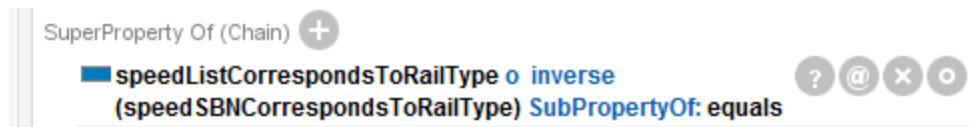

Figure 16: Relationship composition to determine if a station track belongs to a segment

The axioms like in Figure 17, Figure 18 are developed for the second classification of speed by its value and a restriction of each interval such that the order speed corresponding to the rail type of the railway track (by Figure 15 axiom) can be related by the relation «equals» only to the speed in the range of speeds of this rail type.
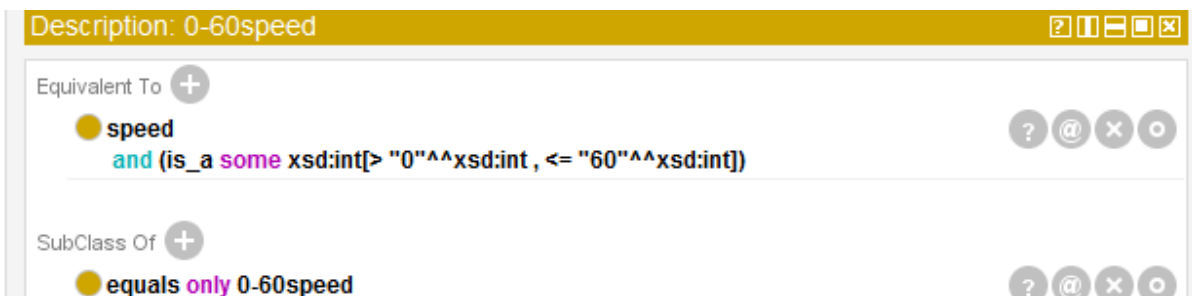

Figure 17: Logical definitions for the speed classification by value
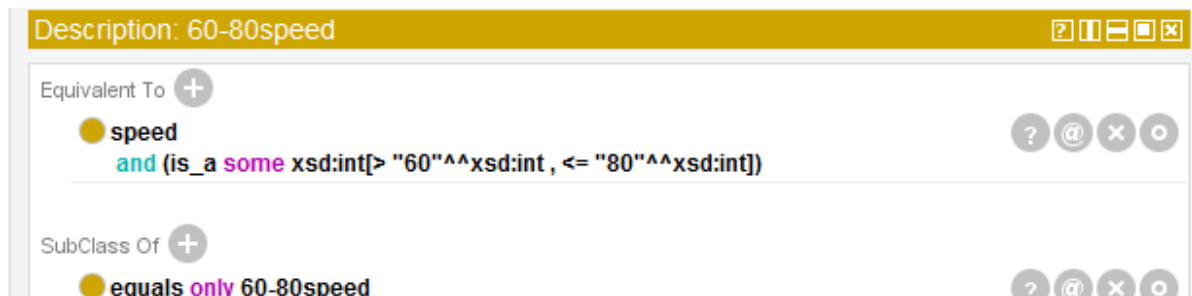

Figure 18: P50 rail type track speed logical definition and restriction

Consider the reasoning path of the main track test instance of **the railway infrastructure concrete model of the second level** ontology, in which there is a 1/9 frog type railway switch corresponding to the siding track.

From the track list, the name instance "main" of the track №1 instance is extracted and associated with the literal "main". The track name instance is classified by the reasoner into the «main track name» class by the axiom of Figure 8.

The track instance is also linked to railway switch №1 in the track list. Switch №1 is associated with the frog type of 1/9 in the switch list. By the Figure 5 composition of relations, the track list and the switch list are linked, the railway switch frog type of track №1 and the railway track №1 are connected by the relation «is switch frog type of».

In the Figure 6 relations composition, №1 track name («main») and railway switch frog type 1/9 are linked by the relation «list corresponds to». The reasoner then classifies the 1/9 frog type into the «main track name frog type» class according to the Figure 9 logical definition.

Individual of the frog type 1/11 that were extracted from the TER text annotations, is linked with the word identifier «3-41» of the word «siding» by the relation «frog type TER corresponds to track name» and the name of the track «siding» is linked with the identifier «3-41» by the relation «word id corresponds to the individual». According to the Figure 7 composition of relations, the frog type 1/9 is associated with the name of the track «siding». An instance of a track name is classified by the reasoner into the class «siding track name» by the axiom like in Figure 8.

Since the frog type of 1/9 is classified by the reasoner into the class of frog types of the main name of the track and is connected by the composition of relations in Figure 4 with the «siding» name of the track, the ontology becomes inconsistent by the restriction of Figure 9.

Consider the reasoning path of a test instance of a track with a P50 rail type and the permitted speed 120 km/h, i.e. equal to the extreme value of the speed range corresponding to the P65 rail type.

In the table of the order, the station «some station 1» is associated with №1 track by the relation «station has track» and with the railway line section «some station 1 - some station 2» by the relation «line has station». Track №1 is linked with the section of the railway line by the relation «line has track » by the Figure 11 composition of relations.

In the order speed table, speed 120 is associated with a section of the railway line and «some Station 1 - some Station 2» by the relation «is speed of line section». Track №1 is linked with speed 120 by the relation «is speed of track» by Figure 12 relations composition.

In the track list, track №1 is linked with the P50 rail type, and track №1 is linked with the literal «P50». Speed 120 is linked with the P50 rail type by the relation «speed drawing corresponding to rail type» according to the Figure 13 composition of relations. Reasoner classifies the P50 instance into the class «P50 rail type», and the speed 120 into the speed class «P50 rail type speed» according to the logical definition of Figure 14-Figure 15.

In the SBN ontology, the speed 120 individual is linked with the individual of the P65 rail type. The ontology becomes inconsistent with the Figure 15 constraint.

7. Conclusions and future work

Improving the safety of train traffic can be achieved by information reliability enhancement of information systems through the integration of heterogeneous data sources. An approach is proposed for formalizing the restrictions of legal regulations by annotating natural language texts.

A modular ontology of railway line section permitted speeds has been developed using the composition of relations. This ontology allows one to check the consistency of the speeds and characteristics of the railway infrastructure in the relevant information systems with TER and SBN.

In the future, it is planned to combine the permanent and temporary speed restrictions due to infrastructure element failure.

8. References

[1] V. Skalozub, V. Ilman, V. Shynkarenko, Development of ontological support of constructive synthesizing modeling of information systems, Eastern-European Journal of Enterprise Technologies 6 (2017) 58-69. doi:10.15587/1729-4061.2017.119497

[2] V. Skalozub, V. Ilman, V. Shynkarenko, Ontological support formation for constructive-synthesizing modeling of information systems development processes, Eastern-European Journal of Enterprise Technologies 5 (2018) 55–63. doi:10.15587/1729-4061.2018.143968

[3] R. Lewis, A semantic approach to railway data integration and decision support, Ph.D. thesis, University of Birmingham, United Kingdom, Electrical and Computer Engineering, 2015.

[4] J. Tutcher, Development of semantic data models to support data interoperability in the rail industry, Ph.D. thesis, University of Birmingham, United Kingdom, Electronic, Electrical, and Systems Engineering, 2016.

[5] S. Bischof, G. Schenner, Rail Topology Ontology: A Rail Infrastructure Base Ontology, in: International Semantic Web Conference, Springer, Cham, 597-612, pp. 2021. doi:10.1007/978-3-030-88361-4_35.

[6] M. Katsumi, M. Fox iCity Transportation Planning Suite of Ontologies, 2020

[7] D. Corsar, M. Markovic, P. Edwards, The transport disruption ontology, in: International Semantic Web Conference, Springer, Cham, 329-336, pp. 2015. doi:10.1007/978-3-319-25010-6_22.

[8] L. Zhao, R. Ichise, S. Mita et al., An ontology-based intelligent speed adaptation system for autonomous cars, in: Joint International Semantic Technology Conference, Springer, Cham, 397-413, pp. 2014. doi:10.1007/978-3-319-15615-6_30.

[9] S. Verstichel, F. Ongenae, L. Loeve et al., Efficient data integration in the railway domain through an ontology-based methodology, Transportation Research Part C: Emerging Technologies (2011) 617-643. doi:10.1016/j.trc.2010.10.003.

[10] F. Benvenuti, C. Diamantini, D. Potena et al., An ontology-based framework to support performance monitoring in public transport systems, Transportation Research Part C: Emerging Technologies (2017) 188-208. doi:10.1016/j.trc.2017.06.001.

[11] R. Balakrishnan, M. A. Harris, R. Huntley, K. Van Auken et.al., A guide to best practices for Gene Ontology (GO) manual annotation. Database, 2013. DOI: 10.1093/database/bat054

[12] P. Cáceres, A. Sierra-Alonso, B. Vela et al., Adding semantics to enrich public transport and accessibility data from the Web, Open Journal of Web Technologies 1 (2020) 1-18.

[13] CEN European reference data model for public transport information. URL: https://www.transmodel-cen.eu/

[14] IFOPT, "Identification of Fixed Objects in Public Transport" Standard CEN/TC 278, EN 28701, European Committee for Standardization, 2012.

[15] P. Ogren, Knowtator: a protégé plug-in for annotated corpus construction, in: Proceedings of the Human Language Technology Conference of the NAACL, Association for Computational Linguistics, USA, 2006, 273-275. doi:10.3115/1225785.1225791

[16] J. C. Klie, M. Bugert, B. Boullosa, The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation, in: 27th International Conference on Computational Linguistics, USA, 2018.

[17] T. Thongkrau, P. Lalitrojwong, Ontopop: An ontology population system for the semantic web, in: IEICE TRANSACTIONS on Information and Systems, CEUR-WS Team, Montenegro, 2012, 921-931.

[18] T. Diamantopoulos, M. Roth, A. Symeonidis et.al. Software requirements as an application domain for natural language processing, Language Resources and Evaluation, 51, (2017) 495-524. doi:10.1007/s10579-017-9381-z

[19] S. Karthikeyan, A. G. S. de Herrera, F. Doctor, et al., An OCR Post-Correction Approach Using Deep Learning for Processing Medical Reports, IEEE Transactions on Circuits and Systems for Video Technology (2021). doi:10.1109/TCSVT.2021.3087641

[20] M. Neumann, D. King, I. Beltagy, ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing, in: Proceedings of the 18th BioNLP Workshop and Shared Task, Association for Computational Linguistics, Italy, 2019, 319-327. doi:10.18653/v1/W19-5034

[21] P. N. Robinson, S. Mundlos, The human phenotype ontology, Clinical genetics (2010) 525-534. doi:10.1111/j.1399-0004.2010.01436.x.

[22] R. Balakrishnan, M. A. Harris, R. Huntley, A guide to best practices for Gene Ontology (GO) manual annotation, Database (2013). doi:10.1093/database/bat054

[23] bioportal.bioontology.org, RxNorm Vocabulary, 2021. URL:
    https://bioportal.bioontology.org/ontologies/RXNORM.

[24] nlm.nih.gov, Medical Subject Headings. URL: https://www.nlm.nih.gov/mesh/meshhome.html.

[25] nlm.nih.gov, Unified Medical Language System. URL:
    https://www.nlm.nih.gov/research/umls/index.html.

[26] G. Munnelly, Entity Linking for Text Based Digital Cultural Heritage Collections, Ph.D. thesis,
    Trinity College Dublin, Ireland, School of Computer Science & Statistics, 2020.

[27] G. Skevakis, EUROMUSE: A web-based system for the management of MUSEum objects and
    their interoperability with EUROpeana, Ph.D. thesis, Technical University of Crete, Greece,
    Electronic and Computer Engineering, 2011.

[28] A. Isaac, B. Haslhofer, Europeana linked open data–data. europeana. eu, Semantic Web 3 (2013)
    291-297. doi:10.3233/SW-120092

[29] C. M. Humphreys, S. McLean, S. Schatschneider, Whole genome sequence and manual
    annotation of Clostridium autoethanogenum, an industrially relevant bacterium, BMC genomics
    16 (2015) 1-10. doi:10.1186/s12864-015-2287-5

[30] J. A. Blake, K. R. Christie, M. E. Dolan, The gene ontology resource: 20 years and still GOing
    strong, Nucleic acids research D1 (2019) D330-D338.

[31] F. Pudlitz, F. Brokhausen, A. Vogelsang, What am I testing and where? Comparing testing
    procedures based on lightweight requirements annotations, Empirical Software Engineering 4
    (2020) 2809-2843. doi: 10.1007/s10664-020-09815-w

[32] F. Pudlitz, A. Vogelsang, F. Brokhausen, A lightweight multilevel markup language for
    connecting software requirements and simulations, in: International Working Conference on
    Requirements Engineering: Foundation for Software Quality, Springer, Cham, 2019, 151-166.
    doi:10.1007/978-3-030-15538-4_11

[33] D. I. Mouromtsev, I. A. Shilin, D. A. Pliukhin et al., Building knowledge graphs of regulatory
    documentation based on semantic modeling and automatic term extraction, Scientific and
    Technical Journal of Information Technologie 21 (2021) 256–266. doi:10.17586/2226-1494-
    2021-21-2-256-266

[34] E. G. Hernández, J. M. Piulachs, Application of the Dublin Core format for automatic metadata
    generation and extraction, in: Proceedings of the 5th International Conference on Dublin Core
    and Metadata Applications, 2005, 213–216.

[35] A. Constantin, S. Peroni, Pettifer S. et al., The document components ontology (DoCO),
    Semantic Web 7 (2016) 167–181. doi:10.3233/SW-150177

[36] M. Villegas, N. Bel, PAROLE/SIMPLE 'lemon' ontology and lexicons, Semantic Web 6 (2015)
    363–369. doi:10.3233/SW-140148

[37] K. Asooja, G. Bordea, G. Vulcu, L. O'Brien, Semantic annotation of finance regulatory text
    using multilabel classification, in: Proceedings of the International Workshop on Legal Domain
    and Semantic Web Applications, Springer, Slovenia, 2015.

[38] R. Nanda, G. Siragusa, L. Di Caro et.al., Concept Recognition in European and National Law, in:
    JURIX, IOS Press, Luxembourg, 2017, 193-198. doi:10.3233/978-1-61499-838-9-193

[39] V. Shynkarenko, L. Zhuchyi, O. Ivanov, Conceptualization of the tabular representation of
    knowledge, in: IEEE 16th International Conference on Computer Sciences and Information
    Technologies, IEEE, Lviv, 2021.

[40] V. Shynkarenko, L. Zhuchyi, Ontological harmonization of railway transport information
    systems, in: 5th International Conference on Computational Linguistics and Intelligent Systems,
    CEUR-WS Team, Lviv, 2021, 541–554.

[41] S. Peroni, The Error Ontology, 2010. URL:
    https://sparontologies.github.io/error/current/error.html