F O U N D A T I O N S O F C O M P U T I N G A N D D E C I S I O N S C I E N C E S
Vol. 47 (2022) No. 3
ISSN 0867-6356
DOI: 10.2478/fcds-2022-0016 e-ISSN 2300-3405

Ontology-Based Semantic Checking of Data in Railway Infrastructure Information Systems

Viktor Shynkarenko*, Larysa Zhuchyi*, Oleksandr Ivanov*

Abstract Semantic checking of railway infrastructure information support data is one of the ways to improve the consistency of information system data and, as a result, increase the safety of train traffic. Existing ontological developments have demonstrated the applicability of description logic for modelling railway transport, but have not paid enough attention to the data resources structure and the railway regulatory support. In this work, the formalization of the tabular presentation of data and the rules of railway transport regulations is carried out using the example of a connection track passport and temporary speed restrictions using ontological means, data wrangling and extraction tools. Ontologies of the various formats data resources and railway station infrastructure, tools for converting and extracting data have been developed. The semantic checking of the compliance of railway information system data with regulatory documents in terms of the connection track passport is carried out on the basis of a multi-level concretization model and integration of ontologies. The mechanisms for implementing the constituent ontologies and their integration are demonstrated by an example. Further research includes ontological checking of natural language normative documents of railway transport.

Keywords: ontology, OWL, semantic checking, data validation, ontology integration, knowledge base, knowledge-based system, concept, information system

1. Introduction

We assume that an information system is a set of procedures for collecting, processing and transmitting information [16]. Information support of railway infrastructure facilities (as well as many other business facilities), as a rule, includes:

- siloed databases that represent the operational state and store the history of changes in various components, for example, financial, accounting, contractual, architectural and construction aspects of the activity;

- planning and reporting system, usually in MS Excel format;

* Ukrainian State University of Science and Technologies, 2 Lazaryana str., Dnipro, 49010, Ukraine, larisa_zhuchiy@ukr.net, shinkarenko_vi@ua.fm, iv_@inbox.ru

- project documentation, guidelines and normative materials in MS Word or Adobe Pdf formats.

National and interstate railway transport information systems have different structures, and there are also many inconsistent databases that provide regulatory, design and operational information.

The Automated Unified Freight Transportation Management System of Ukraine includes station, car, train, locomotive and freight models, which are presented as a set of databases that include relevant regulatory, design and operational documentation. According to the model, specialized automated workstations were implemented.

Directive 2008/57/EC of the European Parliament on the interoperability of the rail system [37] has some railway transport subsystems: infrastructure, control-command and signalling, energy, rolling stock, operation and traffic management, maintenance and telematics applications for passenger and freight services.

Organization for Cooperation of Railways carried out an analysis of the technical requirements and compatibility of the regulatory documentation of the following subsystems: infrastructure, track and track facilities [38]; energy supply; signalling centralization, blocking and communication; passenger cars; locomotives and multiple-unit rolling stock.

In the UK [35] at the beginning of the development of the Rail Core Ontology, they highlighted such railway subsystems as operations (timetable information, route setting & signalling, customer information, environment monitoring, train control, fleet information), planning (timetable planning, track improvement, track/route information), maintenance (infrastructure condition, fleet condition, asset management, scheduling / track ownership), change (infrastructure information, supplier data, design specification).

The emerging need for the integration of railway transport information systems provides for the harmonization of information and increases its reliability.

In this paper, we solve a particular problem of developing a method and instrumental ontological means for coordinating information about the railway station infrastructure. In particular, there is no centralized management of the station model in the information models of the Automated Unified Freight Transportation Management System of Ukraine.

The development of data unification and harmonization procedures is a promising area of research. Automation of the data extraction process, their structuring and checking for compliance with railway regulations is an urgent task.

2. Goals and objectives

The work aims at semantic checking and harmonization of data from various railway transport information systems with the requirements of regulatory documentation based on a multi-level concretization model and integration of ontologies.

We assume that the implementation of the ontological information system involves the solution of several subtasks:

- harmonization of heterogeneous data of different formats and sources using the principles of modularity and the formation of ontologies «from general to particular»;

- formation of multi-level ontologies from abstract to concrete, using several levels of concretization;

- development of tools for linking ontologies based on import, data wrangling and extraction (data extraction, data wrangling);
- performing a consistency check of the developed model based on ontological reasoning in Protégé.

A prototype of an ontology-based information system has been developed, using the example of one of the tasks, namely the semantic control of the data of the connection track passport for compliance with regulatory documentation.

3. Related Works

The concepts of "ontology" and "knowledge base" (KB) have several different definitions. This work relies on a useful one given in [7]: “an ontology defines the common words and concepts (meanings) used to describe and represent an area of knowledge, and so standardizes the meanings. ... An ontology includes the following:

- classes (general things) in the many domains of interest;
- instances (particular things);
- relationships among those things;
- properties (and property values) of those things;
- functions of and processes involving those things;
- constraints on and rules involving those things”.

In [14] “a KB is the knowledge module of a KBS [knowledge-based system]. It contains abstract and specific knowledge of a particular subject in a machine-readable format.”

Guarino [15] and Gómez-Pérez [14] highlight several differences between KB and ontologies, such as the applicability of an ontology to a number of tasks and its reuse by a number of knowledge-based systems, as well as the standardization of the implementation language.

Ontologies are one of the representations of knowledge bases. However, knowledge bases are mainly used as the basis of decision-making systems, and expert systems. In this KBS, the ontology is not associated with decision-making systems but is intended to harmonize the data of information systems and instructional documents, as well as to be an interface for information systems, which will increase the reliability of information and identify cases of non-compliance of the railway infrastructure with standards. As a result, an increase in the train traffic safety level should be expected. The railway management policy supports any effort in this direction.

One of the applications of ontologies is semantic checking. Checking can be performed at the class level and the class instance level. Conceptually, the data is checked against the instructions of the subject area or some developed model. Instrumentally, data validation is carried out using rules or using OWL and Resource Description Framework Schema (RDFS) constructs. Rules can be represented using Semantic Web rule Language (SWRL), Notation 3 Logic (N3Logic), Prolog, etc. For example, RDFS has the rdfs:range construct to limit the range of relationships, while OWL has cardinality limits for classes. One of the ways to develop railway transport ontologies is their constructive-synthesizing modelling [33, 34].

3.1. Checking data compliance with regulations

The compliance of data and normative documents by ontological means is checked taking into account the rules in the N3Logic and SWRL languages.

The construction projects are checked for compliance with the instructions [36] to identify potentially dangerous places. US Occupational Safety and Health Administration guidelines are represented in the form of SWRL rules. The ontology includes Construction Product Model (walls, columns), Construction Process Model (materials and work), and the Construction Safety Model (work and hazards).

In the ontology [27], the construction models were harmonized with the Belgian instruction NBN S 01-400-1. In particular, the EN 12354-3:2000 acoustic comfort calculation formulas are formalized as N3Logic rules for linking construction project instances with, for example, the EN12354:directSoundPowerRatio_4000Hz property.

3.2. Checking data for compliance with models

One of the types of semantic control is checking the data compliance with some ontological model using rules and/or restrictions.

Good examples are modern studies that use rules to check compliance of data and ontological model of data structures [5], performance calculation [2], and XML documents [1].

Formalization of the concepts «set», «list», etc. is implemented in The Collections Ontology [5]. According to their model, a list is an ordered collection of elements. The absence of cycles is checked (the situation in Figure 1).

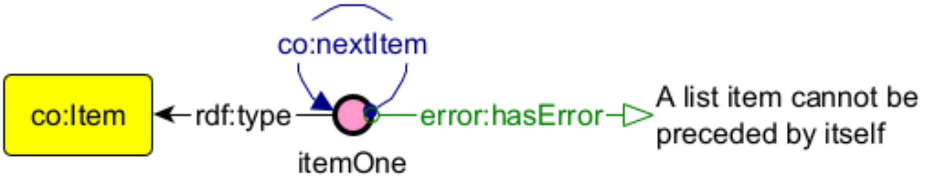


Figure 1. A fragment of an inconsistent ontology having a cyclic order of the list elements

Data are validated according to the SWRL rule (relation co:precededBy is inverse of co:nextItem):

```
precededBy(?x, ?x) -> error:hasError(?x, "A list item cannot be preceded by itself"^^ rdf:PlainLiteral ),
```

where property error:hasError is imported from Error ontology by Silvio Peroni [28]. Error ontology is also used to determine the wrong order of steps in The Publishing Workflow ontology [10].

In [1], a method for the ontological representation of XML documents was developed. The model includes scientific articles and the indication of authors and reviewers. Data matching is performed on the basis of SWRL rules: if the author and reviewer of the article are represented by the same ontology instance, then it has the NoSelfReview type.

With the help of KPIOnto rules [8], the formulas for calculating performance indicators are harmonized [2]. The model formalizes the possibilities of representing formulas. The rules exclude the presence of equivalent and contradictory formulas in the ontology.

By means of OWL and RDFS, the consistency of data and ontology schema can also be checked using constraints. They are implemented by the properties owl:allValuesFrom, owl:someValuesFrom, owl:maxCardinality, rdfs:domain and others.

Semantic control based on constraints for relations is presented in [5, 22, 28].

The constraints on the domain of the co:index relation in [5] check the data type of the list elements indexes. This is necessary because the swrlb:add() built-in functions are used in the model SWRL rules [5]. This restriction is based on the rdfs:range property and the xsd:positiveInteger data type.

```
co:index Range: xsd:positiveInteger
```

By restriction, each literal associated with the co:index relation is of data type positiveInteger . The ontology becomes inconsistent if the knowledge base contains triples of the type «itemOne co: index 1.5» because the literal «1.5» does not belong to the set of integers.

In [28] they check for the absence of instances associated with the error:hasError relation. Such a constraint is implemented on the basis of the values domain that is not associated with an error:hasError relation with any literal:

```
error:hasError Domain: error:hasError exactly 0
rdfs:Literal
```

According to the model [22], the concepts belong to some scheme and are connected by semantic relations, such as skos:broader. To test for the absence of conceptual schemas linking by the skos:broader relation, a constraint on the domain and range is used:

```
:semanticRelation Range: concept
Domain: Concept
```

The ontology becomes inconsistent if the knowledge base contains triples like «scheme1 rdf:type skos:ConceptSheme» and «scheme1 skos:broader scheme2» because the reasoner classifies the instance of scheme1 into the class skos:Concept, and the classes skos:ConceptSheme and skos:Concept do not overlap.

Semantic checking based on constraints for classes is presented in [5, 6, 35].

In [5], the presence of only one first element of the list is checked (the situation in Figure 2) and only positive ordinal numbers of elements.

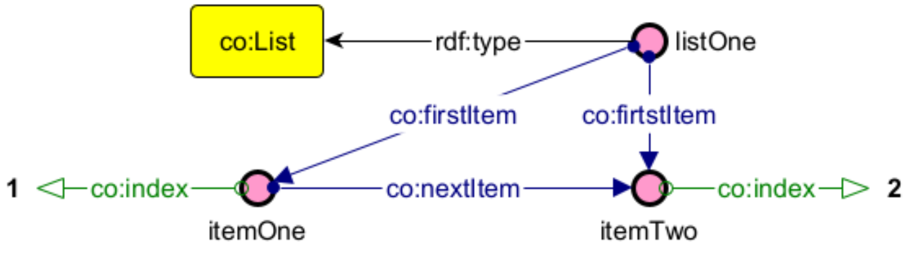


Figure 2. Fragment of inconsistent list ontology that includes two first elements

Data are validated using the following OWL restrictions.

```
co:List SubClass of co:firstItem max 1 owl:Thing
```

```
co:ListItem SubClass of co:index exactly 1 rdfs:Literal
```

The train route can be considered as an instance of a list that contains restrictions on the number of start and end nodes [35].

For the concepts of time, an OWL-Time ontology [6] is developed based on Allen's temporal interval calculus. According to the OWL-Time model, the representation of classes of dates and time intervals is formalized. Based on the class descriptions «:DateTimeDescription» and «:DurationDescription» data format check is performed:

```
:DurationDescription SubClass Of :years only xsd:decimal
:DateTimeDescription SubClass Of :year only xsd:gYear,
```

where xsd:gYear is a data type representing the Gregorian calendar and has values like «2001+02:00».

3.3. Checking data for compliance with software models

Ontologies receive a lot of attention when describing software and data types. Software is characterized by the type of input and output data. This affects the compatibility of the software and its applicability to the processing of user data. Consistency checking is performed by ontological means using universal class restrictions.

The OntoDT ontology [26] presents data types and operations on them. For example, instances of a «table» class and «add record» operation can be associated with the «has operation» relationship, and so on. The ontology also includes a taxonomy of data types such that «tables» are a subclass of «heterogeneous aggregate datatype». The developed classification allows one to perform SPARQL queries to a higher hierarchy class.

The software and data formats used for data analysis in the biological domain are presented in The Software Ontology (SWO) [21]. For example, instances of the program class Microsoft Excel have XLS format output. «XLS class spreadsheet» is a subclass of «data format specification». It should be noted that the structure of the data type is not modelled.

Ontology for software [24] has a high level of abstraction and is designed to describe web services. Its competence question is also the compatibility of the input and output data of the software.

OntoDT is part of the OntoDM ontology development project [26]. According to the OntoDM model, algorithms are applied to data of a certain type. The applicability of an algorithm to data can be described with universal class restrictions.

4. Abstract and Concrete Ontologies

In this work, ontologies of three levels are developed. The abstract level is represented by a vocabulary. The first level concretization models are enriched with rules and data, the Tbox of the ontology scheme is completed, and the dictionary classes are associated with instances. In the second level concretization models, rules are associated with a data set. The presence of the second level of concretization allows one to form several ontologies (based on a single ontology of the first level) using different data.

The abstract level of the ontology determines the conceptual apparatus of the vocabulary of terms (concepts), as well as the concepts of relations. For example, concepts are: railway station, track, signal, etc., while relations are: has gradient, connects station, has algebraic difference (of gradients).

At the first level of concretization, abstract ontologies are enriched in two directions: in one, instances of concepts are formed (for example, a specific station "Kyiv", a specific track No. II of the station "Kyiv"), in the other – rules and restrictions are given (for example, the gradient should be no more than 40 ‰).

At the second level of concretization, abstract and first-level ontologies are connected. The formed concepts have the completeness of properties and relations necessary for semantic checking.

Since information from electronic sources of various formats (databases, spreadsheets and text documents) is subject to harmonization, it is necessary to have a formalized and uniform presentation of data from these sources. The abstract ontological resources model developed by us was presented earlier in [32]. Two-level concretization is performed similarly to the domain ontology described above. The model formed in this way is shown in Figure 3.

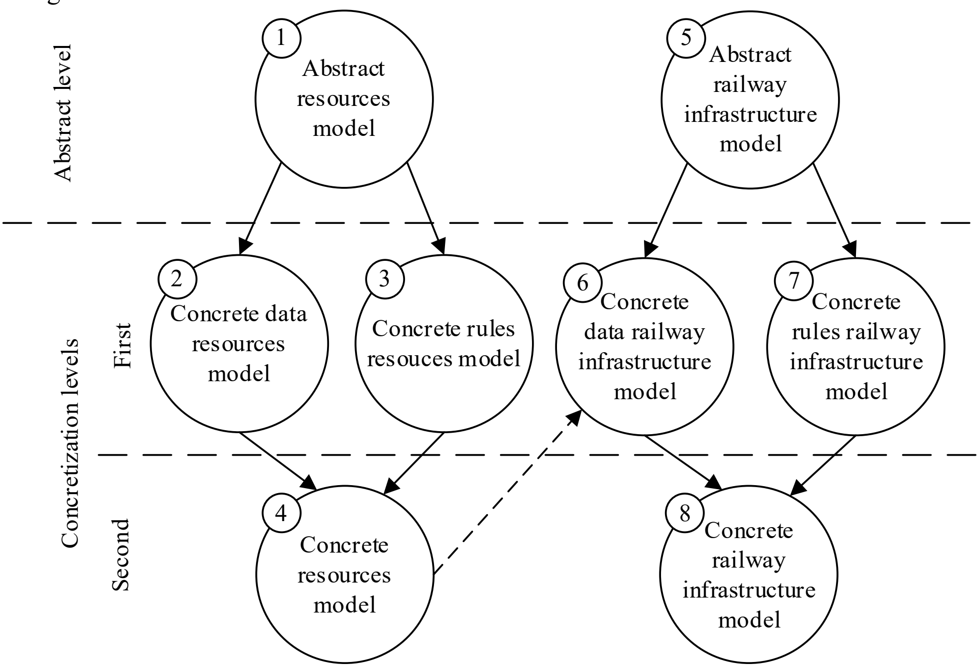


Figure 3. Integration of railway and source models

The models are connected by inclusion to transform the data into an ontological representation (between models 4 and 6).

Let us further consider the possibilities of software implementation of the developed models and the links between them.

5. Ontology development and integration tools

In our research the following software tools were used for the development of ontologies:

- OWL format and RDF format ontology editors Protégé and MRcube;
- data wrangling and data extraction tools OpenRefine and Tabula;
- PDF viewer Sumatra;
- spreadsheets MS Excel and database management system SQLite for the formation of initial data;

Ontologies are processed by various software tools depending on complexity. MRcube supports range constraints. Protégé supports OWL-DL inference.

In Protégé instance checking is executed, i.e., consistency of data and ontology schema verification. The reasoning is done by Hermit using the OWL-DL profile.

The procedure of tools usage for generating models is summarized in Table 1-2.

Table 1. Specification of resources ontological models

Model	Tools/ development way	Formed model realization	Input information sources and formats
Abstract resources model	MRcube/ manual	RDFS/ OWL vocabulary	- the model of tabular data representation [32] - SWO [21]
Specific resources model that includes data	Sumatra/ manual	pdf format resources	Drawings of the station track plan
	Excel/ manual/	xls format resources	Railway track Profile
	SQLite manual	db format resources	List of tracks
	Tabula, OpenRefine, Protégé/ automated	RDF data representing railway infrastructure	Sources in pdf, xls, db format representing the infrastructure of the railway station and Abstract resources model
Concrete resources model that includes rules	Protégé/ manual	The ontology that has data transformation procedures (scripts) and table structure, rules for recognizing the data source, checking the consistency of the table structure	Procedures for original format table data transformation into ontology instances; data resource parameters, Error ontology [28] and Abstract resources model
Concrete resources model	Protégé/ manual	The ontology that is used to perform semantic checking of resource structure	Concrete data resources model and Concrete rules resources model

The process of development of the ontology is illustrated by the example that is given below Figure 4. It shows how to use software tools, as well as importing (using the owl:import mechanism and Protégé «Import ontology wizard») ontologies for their development.

Let's look at the order with an example. Abstract resources model ontology is developed in the MRcube ontology editor. The table showing the owners of the connection track in Pdf format is extracted from the connection track drawing in the Tabula tool, i.e., converted to csv format. Railway track list in db format, extracted table showing track owners in csv format and profile in xls format are transformed in OpenRefine to Concrete data resources model (2) using Abstract resources model (1).

Table 2. Specification of railway infrastructure ontological models

Model	Tools/ development way	Formed model realization	Input information sources and formats
Abstract railway infrastructure model	Protégé/ manual	Vocabulary OWL	The conceptual apparatus of the station elements under the technical exploitation rules of railways (TER) and state building norms (SBN)
Concrete railway infrastructure model that includes data	Protégé/ manual & automated	Ontology that includes data representing the railway infrastructure	Second level concretization resources and bridging ontology
Concrete railway infrastructure model that includes rules	Protégé/ manual	Ontology that includes formalized restrictions imposed on the relations of the Abstract model vocabulary	TER and SBN restrictions and Abstract railway infrastructure model
Concrete railway infrastructure model	Protégé/ manual	Ontology that is used for semantic checking of source data representing railway infrastructure	Concrete data railway infrastructure model and Concrete rules railway infrastructure model
Bridging ontology	Protégé/ manual	Ontology to map the concepts of data resources and railway infrastructure	Abstract resources model and Abstract railway infrastructure model

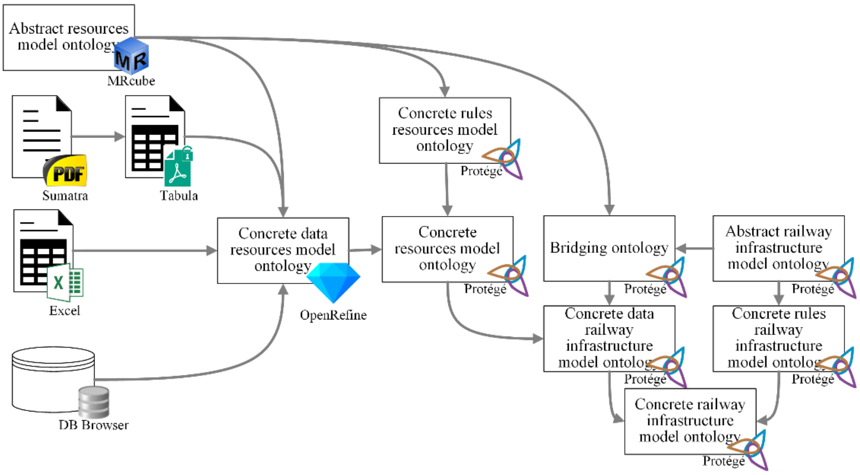


Figure 4. Ontology development workflow

All of the subsequent ontologies are developed in the Protégé ontology editor and imported in the following order:

- Abstract resources model and Abstract railway infrastructure model are imported into the Bridging ontology;
- Abstract resources model is imported into the Concrete data resources model and Concrete resources model that includes rules;
- Concrete resources model that includes data and Concrete resources model that includes rules are imported into the Concrete resources model;
- Concrete resources model and bridging ontology are imported into the Concrete railway infrastructure model that includes data;
- Abstract railway infrastructure model is imported into the Concrete railway infrastructure model that includes rules;
- Concrete railway infrastructure model that includes data and Concrete railway infrastructure model that includes rules are imported into the Concrete railway infrastructure model.

6. The procedure for the formation of the railway infrastructure ontology on the example of the station track

Consider the information support of systems in terms of railway station connection tracks. It includes the following tabular data and regulatory documents:

- railway track lists;
- a table showing the owners of the track;
- longitudinal profile of the track;
- Technical Exploitation Rules of railways at industrial enterprises;
- State Building Norms.

Consider the rules of regulatory documents.

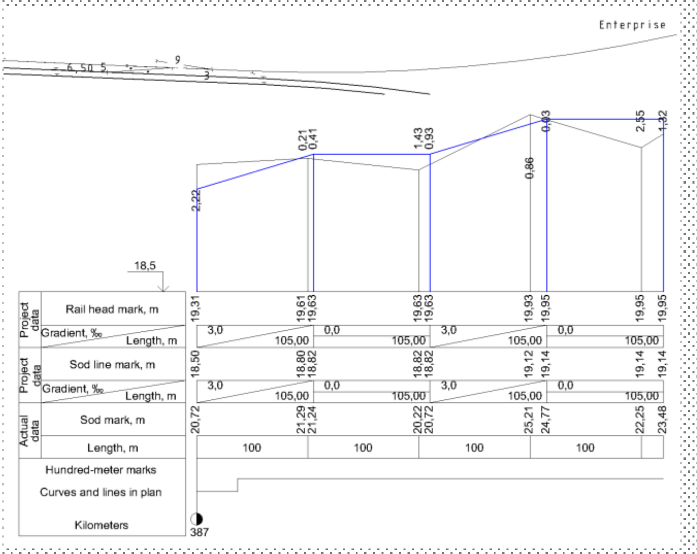
According to the TER, the connection track has the following definition. «Connection tracks – tracks intended for transport service of one or several enterprises, organizations, institutions, connected to the general railway network by a continuous rail track and belonging to the railway or enterprise, organization, an institution».

The definition of the TER has been reformulated into a restriction on the existence of at least one enterprise connected to the station by connection track.

According to SBN, the characteristics of the plan, profile and superstructure of the railway track must satisfy its operating conditions. For the connection track, the algebraic gradients difference should not exceed 13 ppm, and the rail type should be P65.

The initial data (Figure 5) are presented in different formats tables: the track profile is calculated in Excel, the drawing is in AutoCAD [19] (the tabular part of the drawing is converted to pdf format, and the graphic part is presented for illustrative purposes) and the list of tracks is in the database.

A connection track passport includes a track plan and its longitudinal profile. The track plan is an image of the location of railway tracks in space. A longitudinal profile is a projection of a railway track onto a vertical plane. The profile consists of different length gradients and horizontal platforms.



	A	B	C	D	E
1	profile element length	beginning sod line height	ending sod line height	gradient	gradients difference
2	230	434	436	8,7	3,4
3	380	436	438	5,3	3,8
4	220	438	440	9,1	1,9
5	274	440	443	10,9	0,4
6	265	443	446	11,3	2,4
7	336	446	449	8,9	2,6
8	316	449	451	6,3	0,9

Track number	Track name	Track boundary: From switch	Track boundary: Through switches	Track boundary: To switch	Track length: whole	Track length: useful	Rail type
1A	connection	1	-	3	696	650	P65

Ch.	P.n.	sheet	Doc №	signature	data	03-21-TRK.PZH-CH		
						Non-public railway tracks OOO "RostSelMash" of Nyzhnyodniprovsk station of Prydniprovska railway		
created						Non-public railway tracks	stage	sheet	sheets
revised								
examined						Track layout Scale 1:500	OOO "FullCourse"	
Pr. engin.								

Figure 5. Initial data on the example of a connection track passport (profile, list of tracks and a table indicating track owners)

Figure 6 shows an extension of the Figure 3 model considering different format data resources. The numbering of ontologies in Figure 3 and Figure 6 is consistent. The designations of several ontologies of the same type are supplemented with a letter identifier, for example, «2a».

Table 3 summarizes the belonging of ontologies to the Figure 3 concretization levels.

Table 3. Concretization levels of the resources and railway infrastructure ontologies modules

Level	Ontology module	
	Resources ontology	Railway infrastructure ontology
Abstract level	1 Resources	5 Railway station
First concretization level	2a «Table header», 2b «Table structure», 3a «Table software classification rules», 3b «Table type classification rules», 3c «Validation rules»	6a «Track profile», 6b «Track rail type», 6c «Track owners», 7a «TER», 7b «SBN»
Second concretization level	4a «Structure wrangling model», 4b «Table validation model»	8a «Track passport», 8b «Station Model»

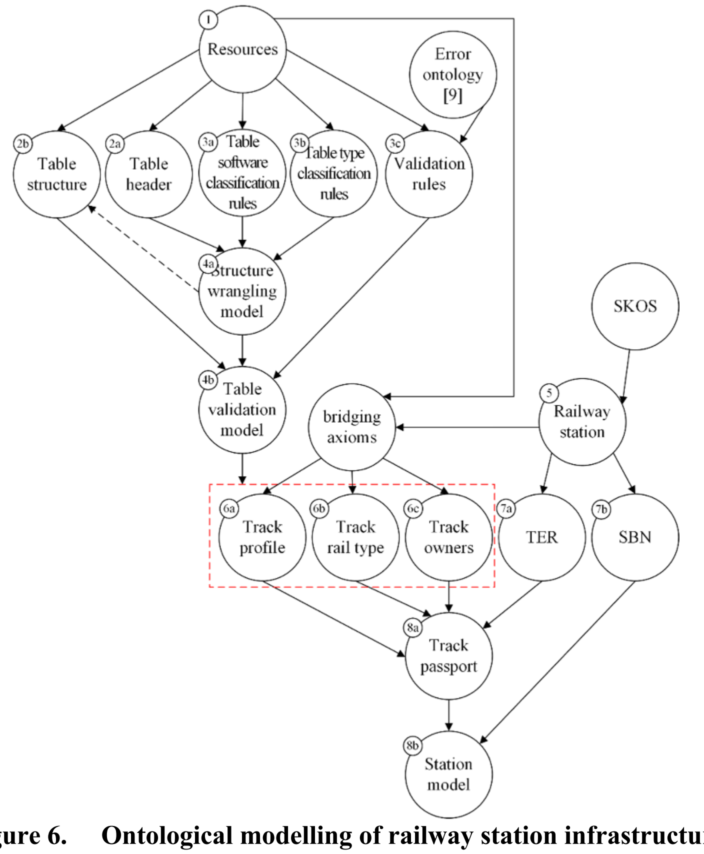


Figure 6. Ontological modelling of railway station infrastructure

The model is formed in the following sequence:
1) abstract ontologies are formed (vocabulary models 1 and 5);
2) model 1 is imported in the (models 3a-3c) ontology:
- in ontology 3a, logical definitions like «if the table is made in Microsoft Excel, then this is a table of the excelTable class» and descriptions of script classes with universal constraints such as «the script has only a railway track profile table as input data» are developed;
- in ontology 3b, logical definitions like «if the table is made in Excel and the table has header <gradient, sod line height…>, then the table is of type «profileTable» are developed, the procedures for transforming the table structure links are attached;
- in ontology 3c, restrictions are formed, in the form of a rule, for example, «if the value individual is represented by the literal «error», then it has the Error ontology property [28]»;

3) model 1 is imported into the ontology (model 2a) and a description of the header, the program for filling the table and the script for its processing are formed;
4) models 2a and 3a, 3b are imported into the ontology (model 4a) and an instance is checked for compliance with a specific table and transformation procedure, if correspondence is established, then a script for converting the table structure can be obtained upon request;
5) model 1, the script obtained in step 4, and a table are imported into OpenRefine (profile calculation in xls format, list of tracks in db, track owners drawing table in csv. The latter one is extracted from the pdf file by the Tabula tool) and the table transformation is performed. Model 2b is formed;
6) models 4a, 2b and 3c are imported into the ontology (model 4b) and the validity of the table data is checked;
7) ontological reasoning is performed, and in case of invalid data, user should correct table errors and go to steps 4-6;
8) resources (model 1) and railway infrastructure (model 5) vocabularies are imported in the bridging ontology to integrate the resources and railway infrastructure ontologies and rules of the following type are formed: «if an instance of the table value is associated with the trackNumber column identifier, then this instance has the railwayTrack type»;
9) steps 2-8 are performed for each type of table (for example, types such as a railway station track list) separately;
10) ontology models 6a-6c are formed including the facts of specific tables: a table indicating the owners of the track (model 6a), a list of tracks (model 6b), and a track profile (model 6c). For each table, a separate ontology is created, where model 4b and bridging ontology are imported. Ontologies of resources and railway infrastructure are integrated;
11) the railway station ontology-vocabulary (model 5) is imported into the first concretization level Concrete railway infrastructure model ontology (models 7a, 7b), where rules and restrictions are formed according to SBN and TER (for example, a restriction on the gradient of the connection track);
12) ontologies including railway track data (models 6a-6c) and the TER ontology (model 7a) are imported into the second concretization level Concrete railway infrastructure model

ontology (model 8a), tables are linked to each other, and the connection track is checked for compliance with TER enterprise standards;
13) ontological reasoning is performed and one should correct table errors and repeat steps 10-12, in case of inconsistencies;
14) track passport ontology (model 8a), SBN ontology (model 7b), are imported into the second concretization level Concrete railway infrastructure model ontology (model8b) to check the track for compliance with Ukrzaliznytsia (UZ) standards (operator of the public railway network). The enterprise and UZ station model databases are integrated.

Next, we will consider the implementation of each model separately.

7. Implementation of individual components of the ontological model

The prototype of the developed ontology for the semantic checking of the connection track passport includes a knowledge base comprising 33 classes, 24 relations and 72 instances. Instances are connected to one railway station connection track. The scale of the UZ information system can be estimated on the basis of the following: the Ukrainian railways consist of 6 divisions, each of which includes about 300 railway stations. Big railway stations can include dozens of connection tracks.

Section 7.1 demonstrates the mechanisms for implementing source ontologies using the example of Figure 5 tables, 7.2 - a railway infrastructure while section 7.2 concerns the railway infrastructure.

7.1. Resources model implementation

Abstract resources model 1 ontology represents a unified scheme for describing various formats tables and satisfies the definition of the [13] lightweight ontology in that it contains concepts and simple relations connecting them. Each data format has its characteristics, which are taken into account when specifying them.

Figure 7 shows the resources ontology vocabulary using the Graffoo notation [9]. «Yellow rectangles represent classes (solid border) and restrictions (dotted border), green parallelograms represent datatypes, arrows starting out of a filled circle refer to object property definitions, arrows starting out of an open circle refer to data property definitions, while other arrows represent assertions between resources » [5].

The ontology reuses concepts from Information artifact ontology [3], Relational Ontology [39], which is part of The Open Biological and Biomedical Ontology (OBO) Foundry, and SWO [21]:

- rdfs: <http://www.w3.org/2000/01/rdf-schema#>
- iao: <http://purl.obolibrary.org/obo/iao.owl#>
- obo: <http://purl.obolibrary.org/obo/>
- swo: <http://www.ebi.ac.uk/swo/>
- table: <http://www.semanticweb.org/larisk0/ontologies/2022/2/table#>

The «Filling software» outline of Figure 7 highlights the concepts of software for filling tables, the «Data wrangling scripts» outline is for scripts for converting resource data into ontology instances, the «Table components» outline is for components of a tabular knowledge representation, the «Table types» outline is for types of resource data tables (for example, track lists and profiles).

The conceptualization of the tabular representation of knowledge was developed in [32]. In the «Table components» outline of Figure 7, only concepts for representing the structure of tables are given: the classes «table», «tuple», «identifier» (in the case of a table, this is the name of a column) and «value». The hasPart relation associates a table with tuples. This relationship is used because the table does not exist without tuples from an epistemological point of view. The hasElement relation associates a tuple with values and identifiers, the «naming» relation associates identifiers with values. The «before» relation is a special case of an order relation, which is necessary because a tuple is an ordered set. It is instantiated for each table and each pair of columns, for example, for a profile and a track list tables, the relationship hierarchy is as follows (Figure 8).

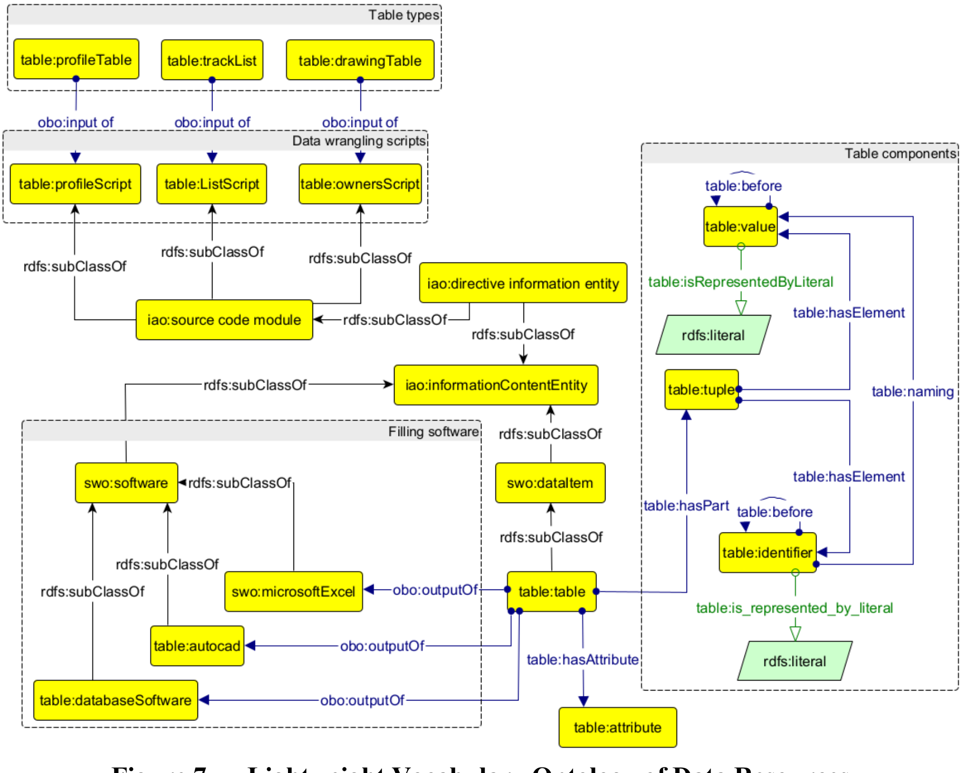


Figure 7. Lightweight Vocabulary Ontology of Data Resources

The general model [32] is specialized for Excel tables with the profileTable classes, as well as the hasAttribute relation and the «attribute» class for linking values of different tables. The relationship is instantiated as enterpriseAttribute and trackAttribute to associate the track list with the track owner and track profile tables, respectively. Microsoft Excel software class is reused from software ontology [21].

The «Filling software» outline of Figure 7 shows Table 1-2 and Figure 4 software. They are typical programs (Excel, AutoCAD, and some DBMS) used in the railway data processing. We use the relation OBO «output of» to link output data and software like in [21].

The «Table types» outline of Figure 7 shows Figure 5 resources data tables (profile and list of tracks, as well as a table indicating the owners). The «Data wrangling scripts» outline in Figure 7 shows the scripts for converting data for each type of table of the «Table types» outline (ownersScript, profileScript, listScript). Just as in OntoDM [25], the relation «input of» is used to link the script program and the input data.

A **Concrete resources model that includes data 2** is developed in the form of two ontologies 2a and 2b. The Abstract resources model 1 ontology is imported into ontologies 2a and 2b.

Triples with the tool for filling the table and about its «header» are entered manually into ontology 2a in Protégé. Ontology 2b is generated automatically in OpenRefine.

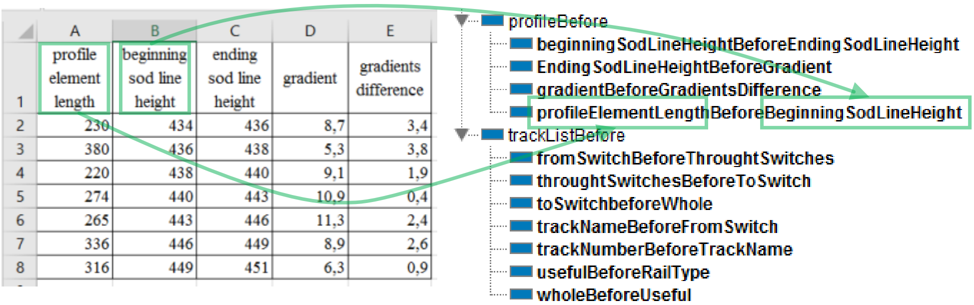


Figure 8. Specification of the order relation for profile tables and list of railway tracks

The table is transformed in two steps in OpenRefine. New «values *n*» columns (cell identifiers) are added. They are created based on existing columns «gradient» and «gradientsDifference». After that, the table is converted to RDF format using the model 1 vocabulary. Thus, the ontology automates ontology population for the table structure instance checking in a Concrete resources model.

A **Concrete resources model 3 that includes rules** is developed in the form of three ontologies 3a-3c. The ontology imports the Abstract resources model 1 ontology and Error ontology [28].

Ontology 3a has logical definitions for classifying a table instance according to its software (for example, the excelTable class in Figure 9), as well as the data wrangling script class restrictions (for example, the profileScript class in Figure 9). And ontology 3b has logical definitions for classifying a table instance according to header identifiers (for example, the profileTable class in Figure 9).

The 3c ontology contains a rule by which a table is given the property of structural errors based on instances of the table that were transformed in OpenRefine. The rule (Figure 9) is made in SWRL.

The **Concrete resources model 4** is developed in the form of two ontologies 4a, 4b.

In ontology 4a, the table represented by vocabulary is classified according to its header and checked against a script to transform the table structure into instances of ontology 2b according to script class restrictions. The Concrete data resources model (ontology 2a of

Figure 6) and Concrete resources model that includes rules (ontologies 3 a and 3b of Figure 6) ontologies are imported into the ontology.

For example, the reasoner infers the triple «table1 rdf:type trackList». The table1 instance is associated with the listScript script instance, so the constraint for the script class of Figure 9 is met. The ontology is consistent.

Semantic checking of the table is performed in ontology 4b using the imported Error ontology [28] property «has error» as in [5]. Concrete data resources model (ontology 2b of Figure 6), Concrete rules resources model (ontology 3c of Figure 6) and Concrete resources model (ontology 4a of Figure 6) ontologies are imported into the ontology.

Let's consider a test instance to demonstrate the processing of a table which structure is not consistent with the model. OpenRefine filters cells that contain a null value and fills them with an «error» value. When such instances are imported into Protégé, the ontology becomes inconsistent because the Error ontology property [28] has zero domain.

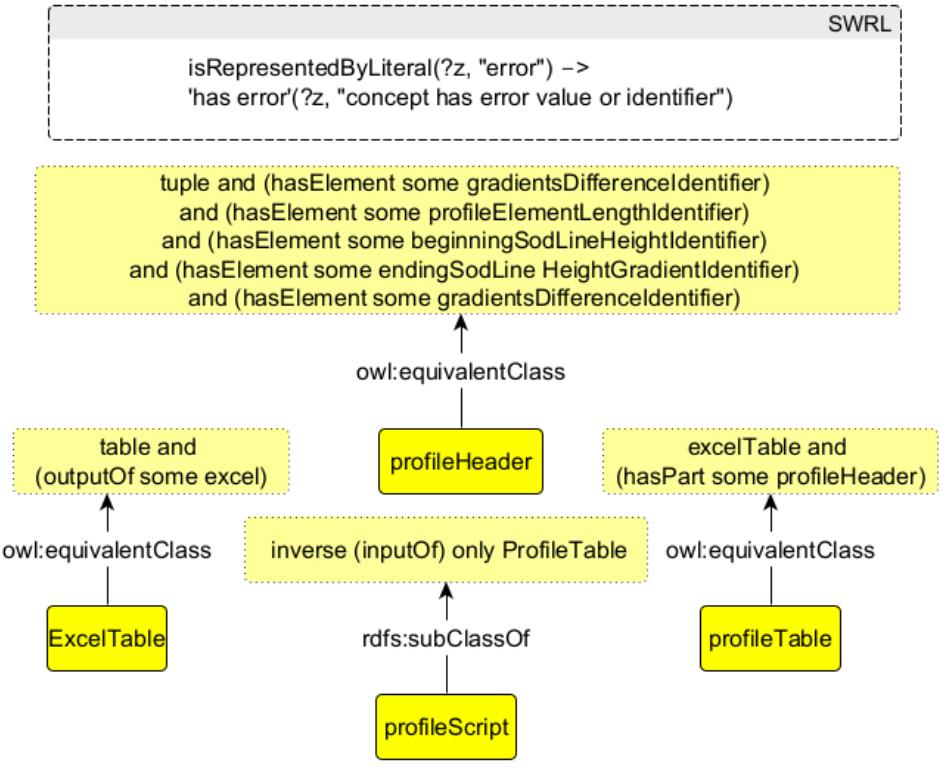


Figure 9. Examples of designs from a specific source model sources that includes rules 3

Figure 10 illustrates the ontology of a Concrete resources model 2 in terms of the first two columns of the track list.

The ontology includes relationships and instances, either added manually or generated in OpenRefine. For example, the triple «tuple0 hasElement identifier1» is added manually, and the triple «tuple 1 hasElement value1» is generated by OpenRefine.

7.2. Integration of resources model and railway infrastructure model

Ontology schemas are mapped to integrate resources and railway infrastructure ontologies (Figure 11). The Concrete resources model that includes data and Concrete resources model that includes rules ontologies (ontologies 5 and 1 of Figure 6) are imported into the bridging ontology.

In Figure 11 construction f) doesn't correspond to Graffoo [9] notation and is developed according to recommendations for Findable, Accessible, Interoperable and Reusable linked data [11].

Classes are linked by owl:equivalentClass relations (Figure 11 construction d)) and logical definitions (Figure 11 construction e)).

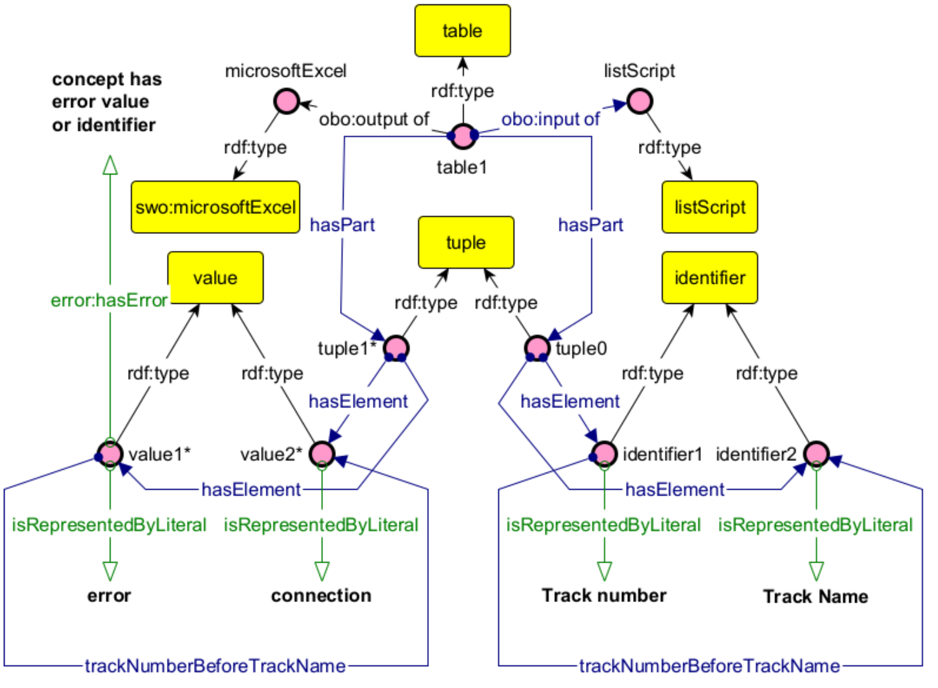


Figure 10. Concrete resources model that includes data 2 in the part of the first two columns of the track list

Relationships are connected by relations owl:equivalentProperty (Figure 11 construction f)) and relations compositions (Figure 11 constructions b)-c)).

The classes and relations of the resources and railway infrastructure ontologies are linked so that the reasoner re-classifies instances of the Concrete resources model ontology (4) as instances of the Abstract railway infrastructure model ontology (5).

Mapping performs the following tasks:

- reclassification of the railway infrastructure ontology valueClass instances depending on the table column. This is done by linking the table identifiers and values (Figure 7) using the «naming» relation of the resources ontology and is demonstrated in the railwayTrack class of constructs e) in Figure 11;

- linking the reclassified instances with railway infrastructure ontology relations. This is done using the «before» order relation linking the tuple values individuals (Figure 8) when linking value instances belonging to the same table and is demonstrated in the hasRailType relation of Figure 11 construction c).

For instances belonging to different tables (infrastructure owner table and track profile), the hasTrackAttribute and hasEnterpriseAttribute relationships are used (Figure 7). When linking the list of tracks and the table with the indication of the owners, the track instance is associated with an instance of the enterpriseAttribute class by the relation hasEnterpriseAttribute according to the composition of the relations (Figure 11 construction a)). Therefore, it suffices to link them as the equivalent relationships of hasEnterpriseAttribute and connectsEnterprise, as shown in Figure 11 construct f).

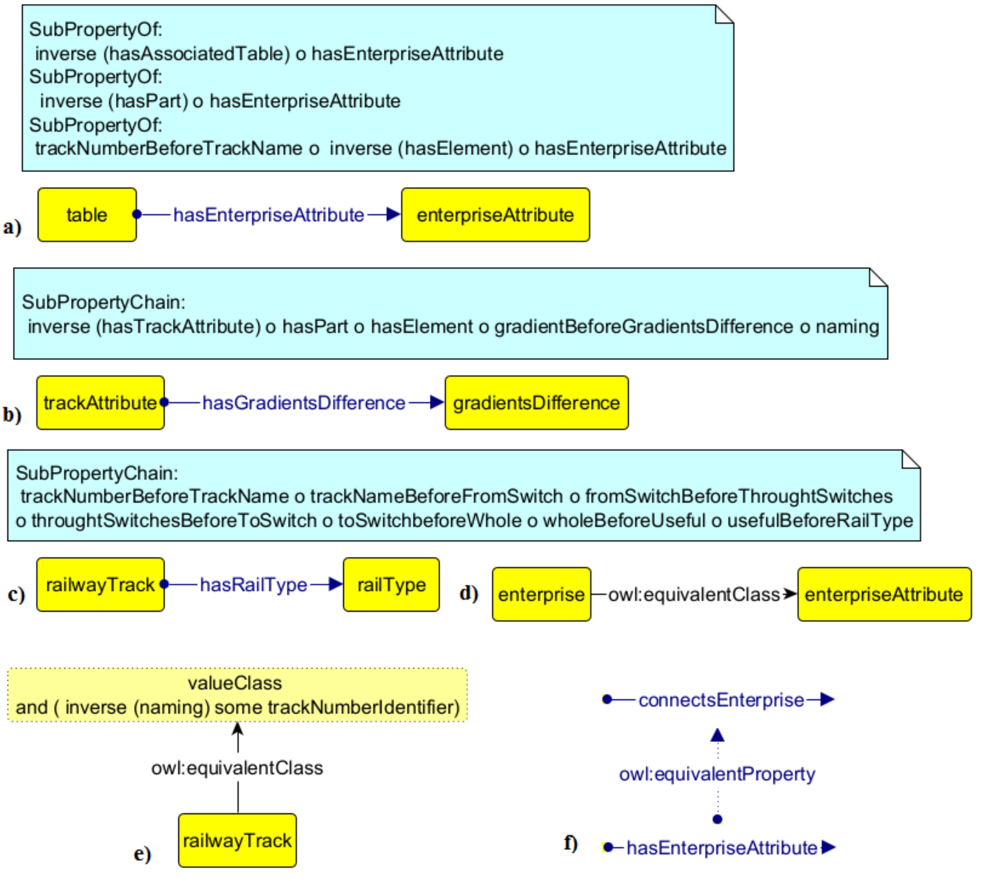


Figure 11. Examples of axioms for ontology integration

In the case of a profile table, the relation gradientBeforeGradientsDifference is used as a pointer to the «Gradients difference» column, which is shown in Figure 11 construction b).

7.3. Implementation of the railway infrastructure model

The Abstract railway infrastructure model 5 ontology-vocabulary is a unified scheme for describing the railway infrastructure of various resources (for example, resources describing the track profile, rail types and owners). Vocabulary includes classes: railwayTrack, railType, enterprise, gradientsDifference and properties: hasRailType, connectsEnterprise, hasGradientsDifference.

The vocabulary also includes definitions of terms from the Traffic Management Rules of Ukraine associated with classes by the property skos:definition.

The Concrete railway infrastructure model that includes rules 7 is developed in the form of two ontologies 7a- 7b. The Abstract railway infrastructure model ontology (Figure 6 ontology 5) is imported into the ontology.

The framework of the modular ontology [31] was developed to integrate railway customer enterprises and UZ databases. Two different regulatory documents packages have been developed for UZ and enterprise railway stations. Therefore, the enterprise track and the UZ track must comply with different restrictions of the TER for enterprises and SBN, respectively. Integration of database ontologies is performed using the logical definitions presented in Figure 13.

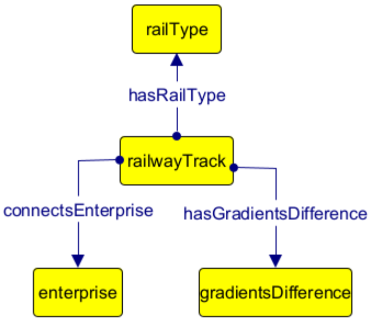


Figure 12. The vocabulary of the railway station infrastructure in the part of the connection track

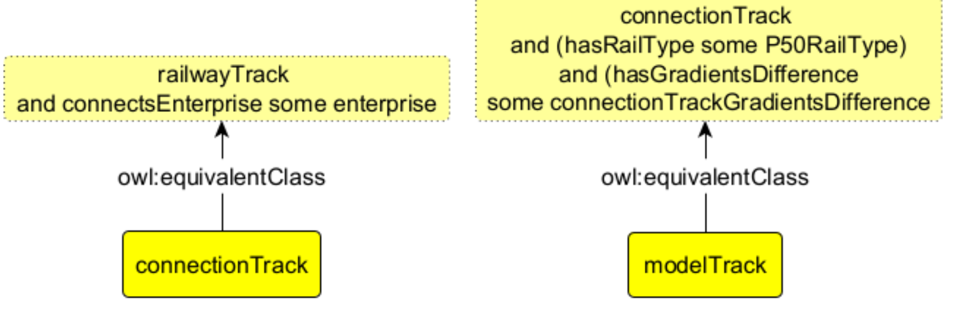


Figure 13. Examples of railway station infrastructure ontology rules

In the ontology of model 7a, TER restrictions on the number of enterprises connected by the connection track are formalized. In the ontology of model 7b, the formalization of three types of SBN constraints is performed.

In Figure 13, the enterprise track is represented by the model 7a «connectionTrack» class, the track added to the station model of the UZ information system is represented by the model 7b «modelTrack» class.

The Concrete railway infrastructure model that includes data 6 is developed in the form of three ontologies 6a-6c using the example of the connection track. Concrete resources model ontologies (Figure 6 ontology 4b) and bridging ontology are imported into the ontology. Resources and railway infrastructure ontologies are integrated.

In ontologies 6a-6b valueClass class instances are reclassified according to the axioms of Figure 11 of type e), and in ontology 6c according to the axiom of type d). Reasoner infers, for example, the triple «value1 rdf:type railwayTrack» in ontology 6b.

In the ontology of model 6b, a rail type instance of the database track list table is associated with the connection track with the property hasRailType by the reasoner according to Figure 11 rule c) of the bridging ontology. Reasoner infers triples like «value1 hasGradientsDifference profileValue5.1» for each 1A track profile element.

The **Concrete railway infrastructure model 8** is developed in the form of two ontologies 8a, 8b.

Semantic checking of data about the track owners is performed in ontology 8a. Concrete railway infrastructure model that includes data (ontologies 6a-6c of Figure 6) and Concrete railway infrastructure model that includes rules ontologies (Figure 6 ontology 7a) are imported into the ontology.

table instances are linked manually. The track list and track owner tables are associated with the hasAssociatedTable relation, and the track list table railway track value and its profile are associated with the hasTrackAttribute relationship. For example, such triples are formed as «table1 hasEnterpriseAttribute attribute1» and «table3 hasTrackAttribute value1».

CSV data about the track owners (extracted from pdf format) RostSelMash and connection track are associated with the relation connectsEnterprise by the reasoner according to the bridging ontology rules of Figure 11 a) and f). Reasoner infers the triple «value1 connectsEnterprise attribute1».

Gradients difference values of the xls table for calculating the track profile and connection track are associated with the relation hasGradientsDifference by the reasoner according to the Figure 11 rule b) of the bridging ontology. Reasoner infers the triple «value1 hasRailType value8».

The reasoner infers the triple «value1 rdf:type connectionTrack» according to the Figure 13 TER logical definition.

Gradients and rail types are checked in ontology 8b. Concrete railway infrastructure model (Figure 6 ontology 8) and Concrete rules railway infrastructure model ontologies (Figure 6 ontology 7b) are imported into the ontology.

The reasoner infers the triple «value1 rdf:type modelTrack» (Figure 14) according to the Figure 13 SBN logical definition.

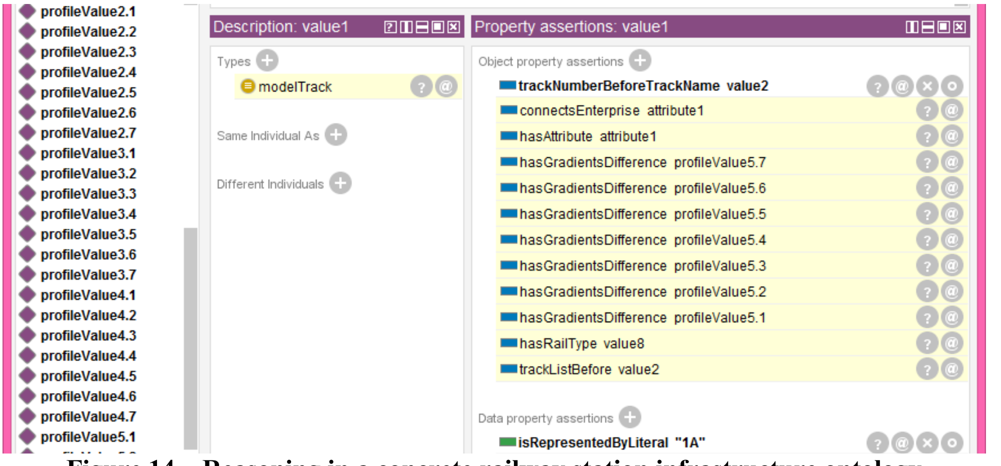


Figure 14. Reasoning in a concrete railway station infrastructure ontology

The enterprise's and UZ's railway track databases are integrated.

7.4. Checking the structure and content of a tabular representation of railway transport data

Data corruption arises from inconsistencies with the model [2], [5] or with the regulation [27], [36]. The paper considers a model of tabular data presentation and railway transport regulations, in particular, related to the construction of the connection track.

There are several approaches to automate data harmonization – by finding errors and assigning additional properties corresponding to the case described by the domain regulations [36]. There are several constructs in OWL for describing constraints, as well as SWRL for rules. Separate ontology modules have been developed to describe errors [28].

Our ontological model (Figure 6) semantic and format checking of data is carried out in a distributed way in models 4 and 8, the rest of the models are designed to prepare the data under study.

The data is validated (in model 4) for compliance with the table structure using the rdfs:domain constraints for the Error ontology module error:hasError property and SWRL rules. So, for example, the test instance of the table containing empty cells, and the value instances are assigned an error property. As soon as the OWL language is based on the principle of an open world, checking restrictions on the minimum cardinality is a non-trivial task, and in the work, it is solved by transforming the table in OpenRefine.

The railway infrastructure ontology (model 8) inconsistencies are searched using logical definitions. Classes conforming to the documentation norms are designed as «defined classes», which imposes normative documentation restrictions on instances and automates their classification. This ensures the semantic consistency of many resources data presented in different electronic formats.

8. Discussion

We emphasize that the work presents only one example of using our prototype for a connection track passport. Further integration is carried out with the Automated Unified Freight Transportation Management System of the Ukraine train model and the railway infrastructure defects data management subsystem.

The gradient of the railway track (Abstract railway infrastructure model ontology class Figure 12) also affects the traction calculations performed when developing a train model timetable. In [31] the railway train model is integrated with the station model by a logical definition such as «a model train is a train that has a weight no greater than that calculated for the train timetable and is located at the station model railway track». Traction calculations are carried out in Excel. For them, checking the structure of resources is an urgent task. The model track definition is shown in Figure 13.

Further, the railway train model can be integrated with wagon and freight ones and speed restriction warnings because the speed value is affected not only by track defects but also by the presence of corresponding wagons and freight in the train.

In the existing ontological developments, little to no attention is paid to the regulatory and legal support of the railway transportation process. Also, restrictions on the applicability of the software (OpenRefine script) for data depending on their type (table, set) and format (xls, csv) have been merged, which was previously only partially implemented.

This section provides an example of using our approach to control data on railway transport speed restriction warnings and compares the study with existing developments.

8.1. Application of the approach for the station model temporary speed restrictions

During the processing of data about track defects and railway track maintenance, there is a manual exchange of data, which can contribute to the occurrence of errors. Checking the structure of speed limits resources and their compliance with regulations is an urgent task, as it has a direct impact on train traffic safety.

The approach shown in Figure 6 remains unchanged, only the types of tables, railway transport regulations and the cause of possible errors change. Data resources can be railway switch and track lists of, a record book and a train driver handout form of speed restriction warnings, as well as books of railway track and switch inspection by representatives of the track facilities. Regulatory documents include traffic management rules, TER, guidelines for the automated issuance of speed restrictions warnings, instructions for the current maintenance of the railway track and safety instructions for performing railway track maintenance.

The manual chain of data exchange includes the following steps:

- track workers identify a railway track defect, make a record in the book of railway tracks inspection and request a speed restriction and a defect application to the track facilities dispatcher by the telephone;

- the track facilities dispatcher makes a record in their warnings record book and conveys data on the speed limit, location and type of defect to a representative of the traffic control system;
- a representative of the traffic control department makes a record in their warning book, prints out a speed restriction warning handout form, checks it against the book and gives it to the driver.

The ontology allows one to check the compliance of defect values and speed restrictions both in different data sources (for example, a railway track inspection book and a speed restriction warnings record book) and according to regulatory documentation. The rules of the instruction for the current maintenance of the railway track include speed restrictions for infrastructure elements depending on their characteristics and defects and are formalized in such OWL restrictions as in Figure 15.

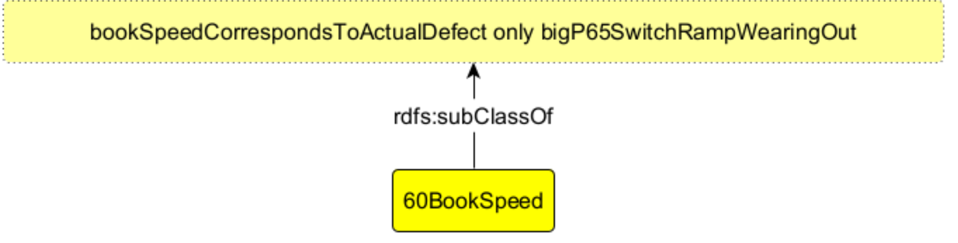


Figure 15. The Concrete railway infrastructure model ontology rules in terms of temporary speed restrictions

The switch list is also a part of the railway station drawing, and the value of the speed limit depends on the degree ramp wearing out and the rail type. Figure 16 shows instances of a Concrete railway infrastructure model in terms of temporary speed restrictions.

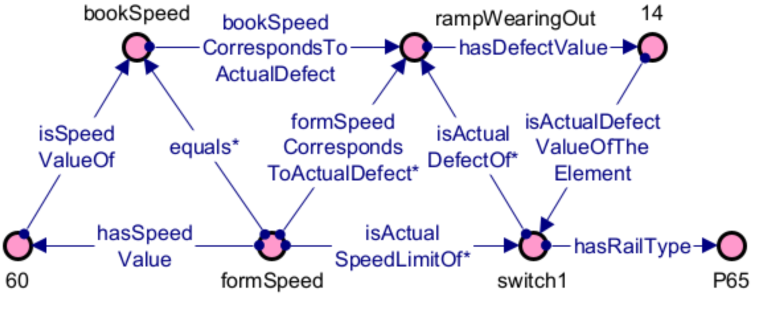


Figure 16. The Concrete railway infrastructure model ontology instances in terms of temporary speed restrictions for one railway switch

The asterisk highlights the relations derived by the reasoner according to the compositions of the relations.

8.2. Comparison of the Resources and Railway Infrastructure Models ontologies with existing knowledge bases

Consider a comparison of the source ontology with existing developments.

In [21, 26,] general software and data types are presented. The Abstract resources model ontology contains the specialization of tables and programs specific to railway software and information support, in addition to the table structure classes.

It can be seen from Table 4 which contains a generalization of the representation software tools by ontological means [21], [24-26] that the linking of the pairs «output data format, software» and «input data type, the algorithm» have already been implemented, which makes it possible to check the compatibility of the triple «format, data type, software».

Table 4. The representation of the software to data processing applicability in existing ontologies

Ontology	Software	Data
OntoDT [26]	Operation	Data type
OntoDM [25]	Algorithm	Data type
SWO [21]	Program	Data format
An ontology for software [24]	Plan	Data

JSON Scripts are used in the OpenRefine environment to transform data of various formats into RDF ontology instances. The script can only be applied to one type of table data. Tables can be in formats such as db, xls, and csv, having weak or strong structures, such as invalid or missing data. Each script is a mapping of a certain set of columns (specializations of a tabular data type, for example, a list of railway tracks) onto an ontology schema.

Abstract resources model and Concrete rules resources model ontologies include scripts for the Concrete resources model ontology population and the tables (profile and list of paths, as well as tables with its owners), which are the source data for the script. Thus, it becomes possible to use constraints to determine the applicability of the script to the table. The script is needed to facilitate the manual entry of instances into the ontology.

The resource structure, considering of the data exchange manual chain and their weak structure (Excel tables), is validated during the development of linked data portals [12] and software for mapping tables onto an ontology schema [4]. In the first case, using SHACL [18], in the second case, using a CSVW vocabulary [17]. All these tools are either lightweight ontologies [17] or do not contain tools adapted to the railway transport needs [5, 12, 29].

Consider the comparison of railway infrastructure model ontologies with existing transport ontological developments.

As a rule, specialists of, for example, track facilities subsystems or systems for primary processing of data collected by systems, for example, axle box heating control, as in the ontology of RaCoOn [35], are engaged in the control of the conformity of the infrastructure characteristics and the transportation process technology. Ontologies [20, 35] integrate data from maps, train timetables, axle box heating sensors, and GPS. Technological processes for the speed restrictions request are not modelled and are not associated with data on the failure of the railway infrastructure according to regulations. More attention is paid to the development of railway network representation patterns.

Existing ontological developments have demonstrated the applicability of description logic for railway transport modelling, but have not paid due attention to the structure of data sources and the railway normative support.

9. Conclusion

The work was carried out to improve the data consistency of railway transport information systems and, as a result, improve the safety of train traffic.

Ontology models of resources of various data formats, ontologies of railway infrastructure and a method for linking them have been developed, which allows performing data validation and semantic checking, as well as reasoning based on railway regulations. As an example, we consider the harmonization of information support during the connection track commissioning, inspections and maintenance.

The railway infrastructure ontology is used to check the data of the connection track passport and speed restrictions warnings with the regulations. It includes classes and relationships for modelling railway infrastructure elements, their characteristics and defects. An auxiliary role is played by ontologies of data resources. They include classes for modelling the resources data structure, software for extracting and transforming them, and the corresponding concretization.

A method is proposed for automating the heterogeneous documents describing station tracks data extraction and wrangling. The knowledge contained in the data of the track passport tables and their structure, as well as the applicability of multi-level concretization to the problem of railway transport various formats data resources integration are used.

The developed model can be extended to other subsystems of railway transport, as well as web pages and open data tables.

References

[1] Almendros-Jiménez JM, Validation of XML Documents with SWRL, in: Bertino E., Tjoa A M. (eds.), *International Conference on Availability, Reliability, and Security*, Springer, Berlin, Heidelberg, 2012, 44-57.

[2] Benvenuti F., Diamantini C., Potena D., Storti E., An ontology-based framework to support performance monitoring in public transport systems, *Transportation Research Part C: Emerging Technologies*, **81**, 2017, 188-208.

[3] Ceusters W., Smith B., Aboutness: Towards foundations for the information artifact ontology, in: *Proceedings of the Sixth International Conference on Biomedical Ontology (ICBO)*, CEUR, 2015, 1-5.

[4] Chaves-Fraga D., Pozo-Gilo L., Toledo J., Ruckhaus E., Corcho Ó., Morph-CSV: Virtual Knowledge Graph Access for Tabular Data, in: *ISWC (Demos/Industry), CEUR Workshop Proceedings*, 2020, 11-16.

[5] Ciccarese P., Peroni S., The Collections Ontology: creating and handling collections in OWL 2 DL frameworks, *Semantic Web*, **5**, 7, 2014, 515-529

[6] Cox S., Little C., Time Ontology in OWL https://www.w3.org/TR/owl-time/. [Accessed 19 April 2022].

[7] Daconta MC, Obrst LJ, Smith KT, The Semantic Web: a guide to the future of XML, Web services, and knowledge management, John Wiley & Sons, 2003.

[8] Diamantini C., Potena D., Storti E. Sem PI, A Semantic Framework for the Collaborative Construction and Maintenance of a Shared Dictionary of Performance Indicators, *Future Generation Computer Systems (FGCS)*, **54**, 2016, 352-365.

[9] Falco R., Gangemi A., Peroni S., Shotton D., Vitali F., Modeling OWL ontologies with Graffoo , in: European Semantic Web Conference, Springer, Cham, 2014, 320-325.

[10] Gangemi A., Peroni, S., Shotton D., Vitali F., The publishing workflow ontology (PWO), *Semantic Web*, **8**, 5, 2017, 703-718.

[11] Garijo D., Poveda- Villalón M., Best practices for implementing fair vocabularies and ontologies on the web, *Applications and Practices in Ontology Design, Extraction, and Reasoning*, **49**, 2020, 39.

[12] Gayo JEL, Prud'hommeaux E., Solbrig HR, Boneva I., Validating and describing linked data portals using shapes, *CoRR*, abs/1701.08924, 2017.

[13] Giunchiglia F., Zaihrayeu I., Lightweight Ontologies, in: Liu L., Özsu MT (eds.), *Encyclopedia of Database Systems*, Springer, Boston, 2009. https://doi.org/10.1007/978-0-387 -39940-9_1314

[14] Gómez-Pérez A., From knowledge based systems to knowledgesharing technology: Evaluation and assessment, Stanford University, 1994.

[15] Guarino N., Giaretta P. Ontologies and knowledge bases, in: Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, 1995, 25-32.

[16] Hall J.A., *Accounting information systems*, Cengage Learning, 2015

[17] Kellogg G., CSVW Namespace Vocabulary Terms https://www.w3.org/ns/csvw. [Accessed 12 April 2022].

[18] Knublauch H., Kontokostas D., Shapes Constraint Language (SHACL) https://www.w3.org/TR/shacl/. [Accessed 19 April 2022].

[19] Kozachenko D., Berezovyi N., Malashkin V., Arbuzov M., Skovron I., Development of the typical passport structure of industrial railways, *Transport systems and transportation technologies*, **14**, 2017, 42-49.

[20] Lewis R., A semantic approach to railway data integration and decision support, University of Birmingham, 2015.

[21] Malone J., Brown A., Lister AL, Ison J., Hull D., Parkinson H., Stevens R., The Software Ontology (SWO): a resource for reproducibility in biomedical data analysis, curation and digital preservation, *Journal of biomedical semantics*, **5**, 1, 2014, 1-13

[22] Miles A., Bechhofer S., SKOS Simple Knowledge Organization System Reference https://www.w3.org/TR/2009/REC-skos-reference-20090818/. [Accessed 12 April 2022].

[23] Noy NF, McGuinness DL, Ontology Development 101: A Guide to Creating Your First Ontology", Stanford Knowledge Systems Laboratory and Stanford Medical Informatics, 2001.

[24] Oberle D., Grimm S., Staab S., An ontology for software, in: *Handbook on ontologies*, Springer, Berlin, 2009, 383-402.

[25] Panov P., Džeroski S., Soldatova LN, Representing entities in the OntoDM data ontology, in: *Inductive mining Databases and Constraint-Based Data Mining*, Springer, New York, 2010, 27-58.

[26] Panov P., Soldatova LN, Džeroski S., Generic ontology of datatypes, *Information Sciences*, **329**, 2016, 900-920.

[27] Pauwels P., Van De ursen D., Verstraeten R., De Roo J., De Meyer R., Van de Walle R, Van Campenhout J., A semantic rule checking environment for building performance checking, *Automation in construction*, **20**, 5, 2011, 506-518

[28] Peroni S., Error Ontology https://github.com/SPAROntologies/error. [Accessed November 19, 2021].

[29] Roman D., Alexiev V., Paniagua J., Elvesæter B., von Zernichow BM, Soylu A., Simeonov B., Taggart C., The euBusinessGraph ontology: A lightweight ontology for harmonizing basic company information, *Semantic Web*, **13**, 1, 2022, 41-68.

[30] Rosen G., *Analysis of Tabula: A PDF-Table extraction tool*, Uppsala University, Uppsala, 2019.

[31] Shynkarenko V., Zhuchyi L., Ontological Harmonization of Railway Transport Information Systems, *International Conference on Computational Linguistics and Intelligent Systems*, CEUR-WS Team, Aachen, Germany, 2021, 541–554.

[32] Shynkarenko V., Zhuchyi L., Ivanov O., Conceptualization of the tabular knowledge representation, *International Conference on Computer Science and Information Technologies*, Lviv Polytechnic National University, Lviv, 2021.

[33] Skalozub V., Ilman V., Shynkarenko V., Development of ontological support of constructive synthesizing modeling of information systems, *Eastern-European Journal of Enterprise Technologies*, **6**, 4, 2017, 58-69.

[34] Skalozub V., Ilman V., Shynkarenko V., Ontological support formation for constructive-synthesizing modeling of information systems development processes, *Eastern-European Journal of Enterprise Technologies*, **5**, 4(95), 2018, 55-63

[35] Tutcher J., *Development of semantic data models to support data interoperability in the rail industry*, University of Birmingham, 2016.

[36] Zhang S., Bou kamp F., Teizer J., Ontology-based semantic modeling of construction safety knowledge: Towards automated safety planning for job hazard analysis (JHA), *Automation in Construction*, **52**, 2015, 29-41.

[37] *Directive 2008/57/E C of the European Parliament and of the Council of 17 June 2008 on the interoperability of the rail system within the Community*

[38] EZhDA Contact Group. *Analysis of parameters defining to maintain the technical and operational compatibility of the 1520 mm track railway system at the cis-EU border. subsystem:infrastructure. Railroad and travel facilities*

[39] Relation Ontology https://obofoundry.org/ontology/ro.html. [Accessed 12 April 2022].

Received 6.12.2021, Accepted 12.07.2022