ISSN 2307–3489 (Print), ISSN 2307–6666 (Online)

Наука та прогрес транспорту, 2022, № 2 (98)

ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ТА МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ

UDC 656.2.027:004.65

V. I. SHYNKARENKO[1*], L. I. ZHUCHYI[2*]

[1*]Dep. «Computer and Information Technologies», Ukrainian State University of Science and Technologies, Lazaryana St., 2, Dnipro, Ukraine, 49010, tel. +38 (063) 489 49 15, e-mail shinkarenko_vi@ua.fm, ORCID 0000-0001-8738-7225
[2*]Dep. «Computer and Information Technologies», Ukrainian State University of Science and Technologies, Lazaryana St., 2, Dnipro, Ukraine, 49010, tel. +38 (059) 965 92 36, e-mail larisa_zhuchiy@ukr.net, ORCID 0000-0002-9209-7262

Constructive-Synthesizing Modelling of Ontological Document Management Support for the Railway Train Speed Restrictions

Purpose. During the development of railway ontologies, it is necessary to take into account both the data of information systems and regulatory support to check their consistency. To do this, data integration is performed. The purpose of the work is to formalize the methods for integrating heterogeneous sources of information and ontology formation. **Methodology.** Constructive-synthesizing modelling of ontology formation and its resources was developed. **Findings.** Ontology formation formalization has been performed, which allows expanding the possibilities of automating the integration and coordination of data using ontologies. In the future, it is planned to expand the structural system for the formation of ontologies based on textual sources of railway regulatory documentation and information systems. **Originality.** The authors laid the foundations of using constructive-synthesizing modelling in the railway transport ontological domain to form the structure and data of the railway train speed restriction warning tables (database and csv format), their transformation into a common tabular format, vocabulary, rules and ontology individuals, as well as ontology population. Ontology learning methods have been developed to integrate data from heterogeneous sources. **Practical value.** The developed methods make it possible to integrate heterogeneous data sources (the structure of the table of the railway train management rules, the form and application for issuing a warning), which are railway domain-specific. It allows forming an ontology from its data sources (database and csv formats) to schema and individuals. Integration and consistency of information system data and regulatory documentation is one of the aspects of increasing the level of train traffic safety.

Keywords: constructive-synthesizing modelling; ontology; information system; railway; database; table; speed restriction; ontology learning

Introduction

In Europe, transport ontologies [15] are developed given the information support evolution and heterogeneous databases integration without changing them. Development is performed using complementary software tools. Railway transport data transformation into an ontology is only partially applied. In [3], the RailTopoModel UML model was transformed into an ontology schema. When developing the Rail Core Ontology [15], OpenRefine is used to transform railway train timetables into ontology individuals.

Previously, we developed a railway track ontology formation procedure. This paper considers the railway ontology formation procedure formalization by means and methods of constructive-synthesizing modelling (CSM) [16].

Problem statement and purpose

Automating the formation of ontologies makes it possible to facilitate the laborious ontology development process and is subject to ontology learning (OL).

OL is performed using tabular, textual sources and a variety of models. OL tools are based on logic rules, machine learning, statistical methods, etc.

When developing railway ontological support, it is necessary to consider both information system

doi: https://doi.org/10.15802/stp2022/268001

data and regulatory support to check their consistency. According to the review [11], insufficient attention is paid to OL's data integration.

The purpose of the work is to formalize heterogeneous information sources integration methods of ontology formation. As an example, the railway track information support is considered in terms of the train speed restriction in connection with its technical condition.

Analysis of recent research and publications

Automation of ontology development implies the automation of schema development and data transformation into ontology individuals.

For the ontology scheme, the following methods are used:

– transformation of the XML schema, for example, using XSD2OWL [6] (as well as UML [14], etc.) into an ontology schema;

– using a controlled language for ontology development, for example, OntoDL in Onto2OWL [8];

– automated semantic annotation of documents, for example, using text2onto [4].

For our work, tabular data structure mapping is more relevant. In the case of the ontology schema generation using the SQL-DDL database schema, mapping rules are used. The tables are mapped to the corresponding ontology classes [2]. Other database-based ontology generation tools are presented in the review [11].

The formalization of ontology scheme formation is also carried out based on ontology pattern language (OPL). As part of the approach, ontology design patterns have been developed, and the possibilities for combining them are demonstrated using OPL tools. To represent the ontology pattern language, one uses grammar [13] and OPL languages [12] based on derivation rules.

Automation of data transformation into ontology instances is performed by the following means:

– data wrangling, such as in the Karma [10] workbench and OpenRefine;

– semantic role labelling, for example in Inception [9];

– Virtual Knowledge Graph System, for example in Ontop [21].

Tools such as «RDB to RDF Mapping Language» are used to map table data onto an ontology schema, followed by ontology population to integrate data and check their consistency.

Another way to automate the development of ontology formation using several software tools is platforms for their integration. Integration can be understood as the connection of services in applications based on ontologies, where RabbitMQ and Redis messages are used [15].

For our work, the integration of ontology development applications in the sense of mapping heterogeneous files is more relevant, for example, as in OntoPop [1], where tags for annotation and ontology classes are mapped. First, documents are annotated, rules are developed in the LangText language [5] for mapping tags and ontology, and then the ontology is populated. Ontology individuals are retrieved from the text.

In other platforms, like [7], Java Patterns Engine Annotation rules are used to map annotations to ontology instances as part of text2onto [4].

Methodology

CSM is based on formal grammar and is used to formalize the formation of structures and constructive processes of various natures. The basis is the generic constructor, which is presented in [16].

Here, the CSM is used to formalize the procedure for developing a railway track ontology using the example of a speed restriction warning due to its defects.

The paper presents only the specialization, interpretation and concretization of the generative constructor for the formation of the speed restriction warning database table. In [18] the following constructors are available:

– formation of railway train management rules (TMR) [20] warning table structure DU-61 and knowledge presentation concepts in csv format;

– converting the structure of the DU-61 warning into the csv format table metadata;

– transforming a database table into a common representation;

– converting a csv table to a generic representation;

– transformation of TMR DU-61 warning table structure and tabular knowledge representation concepts of general representation into ontology vocabulary;

– transformation of the DU-61 form table of the general representation into the ontology individuals;

doi: https://doi.org/10.15802/stp2022/268001

– transformation of the ontology vocabulary and TMR DU-61 warning table structure into ontology rules;

– transformation of schema and instances into ontology.

Findings

Information support of the railway track in terms of defects – a speed restriction warning includes (Figure 1):

– an application for issuing warnings indicating the location and speed restriction;

– the data structure description of the TMR DU-61 warning;

– warning form;

The development of an ontology of data sources for the DU-61 warning involves the following steps:

– filling in the database form DU-61 table according to the request of the track master using SQLite;

– transformation of the database DU-61 form table into a generic representation using OpenRefine;

– filling in csv tables according to the model of tabular knowledge representation [16] and the TMR DU-61 table;

– converting the TMR DU-61 table into a csv metadata table using a text editor;

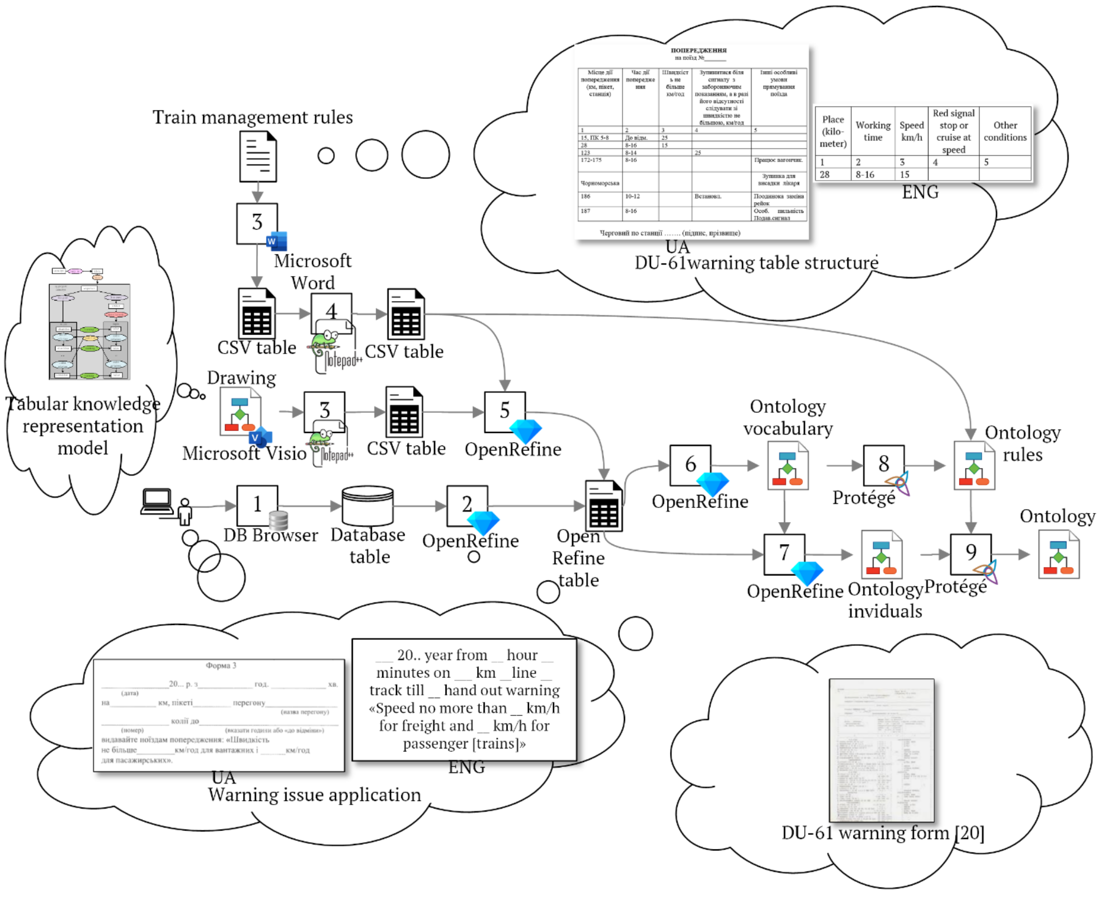
– converting csv tables into a generic representation using OpenRefine;

– transformation of the generic format TMR DU-61 from metadata tables and the tabular knowledge representation model into the ontology vocabulary using the Open Refine RDF extension;

– transformation of the DU-61 form table of the generic format and the ontology vocabulary into ontology instances (ontology Abox) using the Open Refine RDF extension;

– transformation of ontology vocabulary into ontology rules (ontology Tbox) using Protégé;

– integration of schema and ontology instances using Protégé.



ПОПЕРЕДЖЕННЯ
на поїзд №_______

Місце дії попередження (км, пікет, станція)	Час дії попередження	Швидкість не більше км/год	Зупинитися біля сигналу з забороняючим показанням, а в разі його відсутності слідувати зі швидкістю не більшою, км/год	Інші особливі умови прямування поїзда
1	2	3	4	5
15, ПК 5-8	До відм.	25		
28	8-16	15		
123	8-14		25	
172-175	8-16			Працює вагончик.
Чорноморська				Зупинка для висадки лікаря
186	10-12		Встановл.	Поодинока заміна рейок
187	8-16			Особ. пильність Подав.сигнал

Черговий по станції ……. (підпис, прізвище)

Place (kilometer)	Working time	Speed km/h	Red signal stop or cruise at speed	Other conditions
1	2	3	4	5
28	8-16	15		

Форма 3

_______________20... р. з____________ год. ____________ хв.
(дата)
на___________ км, пікеті_________ перегону_________________
(назва перегону)
_______________ колії до_______________________________
(номер) (вказати години або «до відміни»)
видавайте поїздам попередження: «Швидкість не більше_________км/год для вантажних і ______км/год для пасажирських».

___ 20.. year from __ hour __ minutes on ___ km __line __ track till __ hand out warning «Speed no more than __ km/h for freight and __ km/h for passenger [trains]»

Fig. 1. The procedure for the development of constructors for the railway train speed restriction document management ontological support (the numbering corresponds to the data transformation sequence)

doi: https://doi.org/10.15802/stp2022/268001

ISSN 2307–3489 (Print), ISSN 2307–6666 (Online)

Наука та прогрес транспорту, 2022, № 2 (98)

Generative constructor for forming speed restriction warning database table

Constructors for each stage of development are developed that have Protégé, OpenRefine, Tabula and SQLite actors, as well as input and output data (Figure 1). The procedure for developing constructors is demonstrated. For the generative constructor, the inputs are the application, the TMR, and the tabular knowledge representation model. The inputs of the transformation constructor and the outputs of the generative and transformation constructors are the constructs that are obtained through inference.

The purpose of the constructor is to generate a speed restriction warning database table.

Consider the specialization of the generalized constructor (GC) based on the constructive-synthesizing approach:

$$C = \langle \mathrm{M}, \Sigma, \Lambda \rangle \ {}_{s}{\mapsto}\ C_{tabl} = \langle \mathrm{M}_{tabl}, \Sigma_{tabl}, \Lambda_{tabl} \rangle$$

where C – GC; M – heterogeneous expandable carrier OK; $\sum$– the signature of the relations and the corresponding operations of the GC; Λ– a set of synthesis information support (CIS) assertions of the GC; ${}_{s}{\mapsto}$ – specialization operation; C_{tabl} – a specialized constructor (SC) for generating a speed restriction warning table; M_{tabl} – includes the set of terminals SC; $T = \{\Gamma\}$, where Γ is the set of strings, non-terminals $N = \{TABLE, TUPLE3, ELEMENT1, LIT, HEADER\}$, derivation rules; $\sum_{tabl}$ – relations and operations of the SC; Λ_{tabl} – information support for the construction of the SC may include the semantics of concepts and operations, derivation rules, restrictions, and initial and termination conditions.

$\Sigma_{tabl} = \{\rightarrow, \Xi, \Theta, \Phi\}$ includes the signature of operations: $\Xi = \{\circ\}$ – binding; $\Theta = \{|\Rightarrow, \|\Rightarrow\}$ – derivations; $\Phi = \{\#, @, :=, \cdot, AND\}$ – operations on attributes, $\rightarrow$ – derivation relation.

Semantics Λ_{tabl} includes the following concepts: table, table name, table key column (key), tuple, number of columns (b), number of rows (f), attribute value (LIT), sequence relation ($\circ$), as well as [5] operations of full ($\|\Rightarrow$) and partial ($|\Rightarrow$) derivation, constrIndex, constrIndexPart strings.

Λ_{tabl} includes the semantics of operations on attributes:

– # (a, c, *operation*) – the «*operation*» in the # operation is performed when $a = c$;

– @(a) – input a from an external actor;

– := (a, b) value b assignments to a;

– $\cdot(a,b,c)$ – a concatenation of b and c strings with result in a;

– *AND*(*operation*, *operation)* – logical AND, returns true if both operations return «true».

Λ_{tabl} includes the following constraint: the partial derivation is performed considering the derivation relation attribute (${}_{t}{\rightarrow}$), if t = 0, then the rule is not available.

The derivation rules are defined when the constructor is instantiated.

We interpret the constructor through an algorithmic constructor whose actor is DB Browser SQLite:

$$\langle C_{tabl} = \langle M_{tabl}, \Sigma_{tabl}, \Lambda_{tabl} \rangle \rangle,$$

$$C_{A,L} = \langle M_{A,tabl}, \Sigma_{A,tabl}, \Lambda_{A,tabl} \rangle_{I} \rightarrow$$

$$C_{I,L} = \langle M_{I,tabl}, \Sigma_{I,tabl}, \Lambda_{I,tabl} \rangle,$$

where ${}_{I}{\rightarrow}$ is the interpretation operation, $M_{A,tabl}$ includes a set of algorithms implemented by the actor, as well as sets of input and output data for them, $\Sigma_{A,tabl} = \{\cdot, :\}$ includes the signature of operations of sequential and conditional execution of algorithms [5].

Below are the algorithms $V_{A,tabl}$ that implement the corresponding operations on the attributes: $A_1 |_{a,c}^{A_j}$ – the algorithm A_j is executed if $a = c$, $A_2 |_{\tilde{a}}^{a}$ is the implementation of the @ operation; $A_3 |_{b}^{a}$ – $a = b$; $A_4 |_{b,c}^{a}$ – $\cdot(a,b,c)$, $A_5 |_{A_j,B_j}^{true, falce}$ – *AND*(*operation*, *operation)*, where A_j is the identifier of the algorithm, X_j,Y_j are the domains and values of the algorithm $A_i |_{Y_i}^{X_i}$.

Carrier $M_{I,table} = M_{tabl} \bigcup M_{A,tabl}$.

Information support $\Lambda_{A,tabl}$ is presented in [5],

$$\Lambda_{I,tabl} = \Lambda_{tabl} \bigcup \Big\{ \Big(A_1 |_{a,c}^{A_j} \lrcorner \# \Big), \Big(A_2 |_{\tilde{a}}^{a} \lrcorner @ \Big)$$

$$\Big(A_3 |_{b}^{a} \lrcorner := \Big), \Big(A_4 |_{b,c}^{a} \lrcorner \cdot \Big), \Big(A_5 |_{A_j,B_j}^{true, falce} \lrcorner AND \Big) \Big\}$$

doi: https://doi.org/10.15802/stp2022/268001

Consider the concretization of the constructor:
$C_{I,tabl} = \langle M_{I,tabl}, \Sigma_{I,tabl}, \Lambda_{I,tabl} \rangle_K \rightarrow$
$C_{K,tabl} = \langle M_{K,tabl}, \Sigma_{K,tabl}, \Lambda_{K,tabl} \rangle$, , where $_K\!\rightarrow$ operation specification, $M_{K,tabl} = M_{I,tabl}$
$\Sigma_{K,tabl} = \Sigma_{I,tabl}$, $\Lambda_{K,tabl}$ includes the following.

Initial conditions: table header count headerCnt = 0, number of columns (b), number of rows (f), key column number keyColNum = 0, row counter tupleCnt = 0, value counter elementCnt = 0, constructor-constructed value index constrIndex, part of index constructed constrIndexPart, $t_0 = 1$, $t_1 = 0$, $t_2 = 0$, $t_3 = 0$, $t_4 = 0$, $t_5 = 0$, $t_6 = 0$, $t_7 = 0$, $t_8 = 0$, $t_9 = 0$, $t_{10} = 0$, $t_{11} = 0$.

Non-terminals: TABLE, TUPLE, TUPLE1, VECTOR, LIT, HEADER1, TUPLE2, TUPLE3, ELEMENT1.

Terminals: b, f, key, table name, header, value, element, index.

The initial non-terminal is TABLE.

Input data – an application for issuing a warning, for example «at 23 km, give trains a warning to go at a speed of no more than 60 km/h»;

Restrictions described earlier.

Termination condition – all derivation relations are invalid.

The output is a warning database table (Table 1); Derivation rules (include derivation relationships s and attribute operations g):

$s_1 = \langle {}_{lit\ attr}table_{t_0} \rightarrow TUPLE1 \circ VECTOR \circ {}_{lit\ attr}\ TUPLE \rangle$

where $_{LIT}attr = {}_{LIT}b, {}_{LIT}f, {}_{LIT}key, {}_{LIT}table\ \ name.$

$g_1 = \langle := (t_0, 0), := (t_1, 1), @\left(LIT \hookleftarrow b \hookleftarrow TABLE\right),$
$@\left(LIT \hookleftarrow f \hookleftarrow TABLE\right), @\left(LIT \hookleftarrow key \hookleftarrow TABLE\right)$
$@\left(LIT \hookleftarrow table\ name \hookleftarrow TABLE\right)\rangle$

Table 1

Shortened warning form DU-61

Place	Speed
23 km	60

Consider the realization example corresponding to Table 1.

Derivation 1.

$_{LIT}b, {}_{LIT}f, {}_{LIT}key, {}_{LIT}table\ name\,TABLE \overset{s_1}{\Longrightarrow}$
$TUPLE1 \circ VECTOR \circ$

$_{LIT}b, {}_{LIT}f, {}_{LIT}key, {}_{LIT}table\ name\,TABLE \overset{s_2}{\Longrightarrow}$
$\langle t_0 = 0, t_1 = 1, LIT \hookleftarrow b \hookleftarrow TABLE = 2$
$LIT \hookleftarrow f \hookleftarrow TABLE = 1$
$LIT \hookleftarrow key \hookleftarrow TABLE = place$
$LIT \hookleftarrow table\ name \hookleftarrow TABLE = DU61\rangle$
$s_2 = \langle TUPLE1_{t_1} \rightarrow {}_{LIT\ value}header \circ HEADER1 \rangle$
$g_2 = \langle := (t_2, 1), +\left(headerCnt, headerCnt, 1\right)$
$@\left(LIT \hookleftarrow value \hookleftarrow header\right),$
$\#\left(LIT \hookleftarrow value \hookleftarrow header, LIT \hookleftarrow value \hookleftarrow key,\right.$
$\left. := (kelCol, Num, headerCnt))\right)$
$\#\left(headerCnt, b, := (t_2, 0)\right)$
$\#\left(headerCnt, b, := (t_3, 1)\right)\rangle$

Derivation 2.
$_{LIT\ value}header \circ HEADER1 \circ VECTOR \circ$

$_2b, {}_1f, {}_{place}key, {}_{DU61}table\ name\,TABLE \overset{s_3}{\Longrightarrow}$
$\langle t_2 = 1, headerCnt = 1,$
$LIT \hookleftarrow value \hookleftarrow header = place$
$keyColNum = 1\rangle.$
$s_3 = \langle HEADER1_{t_2} \rightarrow {}_{LIT\ value}header \circ HEADER1 \rangle$
$g_3 = \langle +\left(headerCnt, headerCnt, 1\right)$
$\#\left(headerCnt, b, := (t_2, 0)\right)$
$@\left(LIT \hookleftarrow value \hookleftarrow header\right),$
$\#\left(headerCnt, b, := (t_3, 1)\right)$
$\#\left(headerCnt, b, := (t_7, 1)\right)$
$\#\left(LIT \hookleftarrow value \hookleftarrow header, LIT \hookleftarrow value \hookleftarrow key,\right.$
$\left. := (kelCol, Num, headerCnt))\right)\rangle$

doi: https://doi.org/10.15802/stp2022/268001

Derivation 3.

$$ {}_{place}value\,header \circ {}_{LIT}value\,header \circ $$
$$ HEADER1 \circ VECTOR \circ $$
$$ {}_{2}b, {}_{1}f, {}_{place}key, {}_{DU\,61}table\ name\,TABLE \overset{s_4, s_8}{\Longrightarrow} $$
$$ \langle headerCnt = 2, t_2 = 0, $$
$$ LIT ↲ value ↲ header = speed $$
$$ t_3 = 1, t_7 = 1 \rangle. $$
$$ s_4 = \langle VECTOR_{t_3} \to TUPLE2 \circ TUPLE3 \rangle $$
$$ g_4 = \langle := (t_5, 1), +(tupleCnt, tupleCnt, 1) $$
$$ \#(tupleCnt, f, := (t_8, 1)) \rangle $$

Derivation 4.

$$ {}_{place}value\,header \circ {}_{LIT}value\,header \circ TUPLE2 \circ TUPLE3 \circ $$
$$ {}_{2}b, {}_{1}f, {}_{place}key, {}_{DU\,61}table\ name\,TABLE \overset{s_6, s_9}{\Longrightarrow} $$
$$ \langle t_5 = 1, tupleCnt = 1, t_8 = 1 \rangle. $$
$$ s_5 = \langle TUPLE3_{t_3} \to TUPLE2 \circ TUPLE3 \rangle $$
$$ g_5 = \langle := (t_5, 1), +(tupleCnt, tupleCnt, 1) $$
$$:= (t_4, 1), \#(tupleCnt, f, := (t_8, 1)) \rangle $$
$$ s_6 = \langle TUPLE2_{t_5} \to {}_{LIT}value\,element \circ $$
$$ ELEMENT1 \rangle, $$

where ${}_{LIT}attr\,element = {}_{LIT}type, {}_{LIT}index, {}_{LIT}value\,element.$

$$ g_6 = \langle := (elementCnt, 1), $$
$$ @(LIT ↲ value ↲ element), $$
$$ \#(elementCnt, headerCnt, := (t_9, 1)), := (t_6, 1), $$
$$ \#(elementCnt, headerCnt, := (t_{10}, 1)), $$
$$ \#(elementCnt, keyColNum, $$
$$:= (constrIndexPart, LIT ↲ value ↲ element)), $$
$$ \#(elementCnt, keyColNum, $$
$$ \cdot(constrIndex, constrIndexPart, $$
$$ LIT ↲ table\ name ↲ element)) $$
$$ \#(elementCnt, keyColNum, := (elementCnt, 0))) \rangle $$

Derivation 5.

$$ {}_{place}value\,header \circ {}_{speed}value\,header \circ $$
$$ {}_{LIT}type, {}_{LIT}index, {}_{LIT}value\,element \circ ELEMENT1 \circ $$
$$ {}_{2}b, {}_{1}f, {}_{place}key, {}_{DU\,61}table\ name\,TABLE \overset{s_7}{\Longrightarrow} $$
$$ \langle elementCnt = 1, LIT ↲ value ↲ element = 23km $$
$$ t_6 = 1, j = 23km, constrIndex = 23kmDU\,61 \rangle. $$
$$ s_7 = \langle ELEMENT1_{t_6} \to {}_{LIT}value\,element \circ $$
$$ ELEMENT1 \rangle, $$
$$ g_7 = \langle +(elementCnt, elementCnt1), $$
$$ \#(elementCnt, headerCnt, := (t_6, 0)) $$
$$ @(LIT ↲ value ↲ element), $$
$$ \#(elementCnt, headerCnt, := (t_{10}, 1)), $$
$$ \#(elementCnt, headerCnt, := (t_{10}, 1)), $$
$$ \#(elementCnt, keyColNum, $$
$$:= (constrIndexPart, LIT ↲ value ↲ element)), $$
$$ \#(elementCnt, keyColNum, $$
$$ \cdot(constrIndex, constrIndexPart, $$
$$ LIT ↲ table\ name ↲ element)) $$
$$ \#(elementCnt, keyColNum, := (elementCnt, 0))) \rangle $$

Derivation 6.

$$ {}_{place}value\,header \circ {}_{speed}value\,header \circ $$
$$ {}_{string}type, {}_{LIT}index, {}_{23km}value\,element \circ $$
$$ {}_{LIT}type, {}_{LIT}index, {}_{LIT}value\,element \circ ELEMENT \circ $$
$$ {}_{2}b, {}_{1}f, {}_{place}key, {}_{DU\,61}table\ name\,TABLE \overset{s_{11}}{\Longrightarrow} $$
$$ \langle elementCnt = 2, t_6 = 0, $$
$$ LIT ↲ value ↲ element = 60, t_{10} = 1, $$
$$ constrIndex = 23kmDU\,61, elementCnt = 0 \rangle. $$
$$ s_8 = \langle HEADER1_{t_7} \to \varepsilon \rangle, $$
$$ g_8 = \langle \varepsilon \rangle. $$
$$ s_9 = \langle TUPLE3_{t_8} \to \varepsilon \rangle, $$
$$ g_9 = \langle \varepsilon \rangle. $$
$$ s_{10} = \langle ELEMENT1_{t_9} \to \varepsilon \rangle, $$

$$g_{10} = \Big\langle \#\big(AND\big(\big(tupleCnt, f\,\big),$$
$$\big(elementCnt, headerCnt\big)\big), := \big(t_{11}, 1\big)\big),$$
$$:= \big(t_4, 1\big), := \big(t_5, 0\big), := \big(t_8, 0\big), := \big(t_9, 0\big)\Big\rangle$$

Derivation 9.

$${}_{place}value\;header \circ {}_{speed}value\;header \circ$$
$${}_{string}type, {}_{23kmDU\,61}index, {}_{23km}value\;element \circ$$
$${}_{string}type, {}_{23kmDU\,61}index, {}_{60}value\;element \circ$$
$${}_{2}b, {}_{1}f\,, {}_{place}key, {}_{DU\,61}table\ name\;TABLE \overset{s`_{12}}{\Longrightarrow}$$
$$s_{11} = \big\langle LIT \lrcorner index_{t_7} \to n \big\rangle,$$
$$g_{11} = \big\langle +(elementCnt, elementCnt1),$$
$$\#\big(elementCnt, headerCnt, := (t_{10}, 0)\big)$$
$$\#\big(elementCnt, headerCnt, := (t_9, 0)\big)\big\rangle$$

Derivation 7.

$${}_{place}value\;header \circ {}_{speed}value\;header \circ$$
$${}_{string}type, {}_{23kmDU\,61}index, {}_{23km}value\;element \circ$$
$${}_{string}type, {}_{LIT}\,index, {}_{60}value\;element \circ ELEMENT1 \circ$$
$${}_{2}b, {}_{1}f\,, {}_{place}key, {}_{DU\,61}table\ name\;TABLE \overset{s`_{11}}{\Longrightarrow}$$
$$\big\langle elementCnt = 1 \big\rangle$$

Derivation 8.

$${}_{place}value\;header \circ {}_{speed}value\;header \circ$$
$${}_{string}type, {}_{23kmDU\,61}index, {}_{23km}value\;element \circ$$
$${}_{string}type, {}_{23kmDU\,61}index, {}_{60}value\;element \circ ELEMENT1 \circ$$
$${}_{2}b, {}_{1}f\,, {}_{place}key, {}_{DU\,61}table\ name\;TABLE \overset{s`_{10}}{\Longrightarrow}$$
$$\big\langle elementCnt = 1, t_9 = 1, t_{10} = 0 \big\rangle$$
$$s_9 = \big\langle TABLE_{t_{11}} \to eof \,\big\rangle,$$
$$g_{12} = \langle \varepsilon \rangle.$$

Derivation 10.

$${}_{place}value\;header \circ {}_{speed}value\;header \circ$$
$${}_{string}type, {}_{23kmDU\,61}index, {}_{23km}value\;element \circ$$
$${}_{string}type, {}_{23kmDU\,61}index, {}_{60}value\;element \circ eof$$

Originality and practical value

The basis for automating the formation of the ontology of the railway domain by the constructive-synthesizing method has been laid.

The authors laid the foundations of using constructive-synthesizing modelling in the railway transport ontological domain to form the structure and data of the railway train speed restriction warning tables (database and csv format), their transformation into a common tabular format, vocabulary, rules and ontology individuals. As well as ontology population.

Ontology learning methods have been developed to integrate data from heterogeneous sources (the structure of the table DU-61 TMR, the form and application for issuing a warning), and is also railway-oriented. It allows forming an ontology from its data sources (database format and csv) to schema and instances.

Integration and consistency checking of data from information systems and regulatory documentation is one of the aspects of increasing the train traffic safety level.

Conclusions

The formalization of the ontology formation has been performed, which allows expanding the possibilities of automating the data integration and checking its consistency using ontologies.

In the future, it is planned to expand the structural system for the formation of ontologies based on textual sources of regulatory documentation and information systems of the railway.

LIST OF REFERENCE LINKS

1. Amardeilh F. OntoPop or how to annotate documents and populate ontologies from texts. *ESWC 2006 Workshop on Mastering the Gap : From Information Extraction to Semantic Representation.* 2006. P. 1–16.
2. An J., Park Y. B. Methodology for Automatic Ontology Generation Using Database Schema Information. *Mobile Information Systems.* 2018. Vol. 2018. P. 1–13. DOI : https://doi.org/10.1155/2018/1359174

3. Bischof S., Schenner G. Rail Topology Ontology : A Rail Infrastructure Base Ontology. *The semantic web - ISWC 2021.* 2021. P. 597–612. DOI: https://doi.org/10.1007/978-3-030-88361-4_35
4. Cimiano P., Völker J. A Framework for Ontology Learning and Data-Driven Change Discovery. *NLDB'05 : Proceedings of the 10th international conference on Natural Language Processing and Information Systems*. 2005. Vol. 3513. P. 227–238. DOI: https://doi.org/10.1007/11428817_21
5. Crispino G. *Une Plateforme Informatique de l'Exploration Contextuelle: Modelisation, Architecture et Realisation (ContextO). Application au Filtrage Semantique de Textes* : Thesis or Dissertation. Universite De Paris Iv – Sorbonne, 2003. 241 p.
6. Cruz C., Nicolle N. Ontology enrichment and automatic population from XML data. *Proceedings of the 4th International VLDB Workshop on Ontology-based Techniques for DataBases in Information Systems and Knowledge Systems*. 2008. P. 1–5.
7. Dinşoreanu M., Salomie I., Pop C. B. Integrated System for Developing Semantically-Enhanced Archive Econtent. *Revista Romana de Informatica si Automatică*. 2011. Vol. 21, No. 4. P. 67–77.
8. Fonseca J. M. S. *Converting ontologies into DSLs* : Master Dissertation. Universidade do Minho Escola de Engenharia Departamento de Inform´atica, 2014. 80 p.
9. Klie J. C., Bugert M., Boullosa B., de Castilho R. E., Gurevych I. The inception platform: Machine-assisted and knowledge-oriented interactive annotation. *Proceedings of the 27th International Conference on Computational Linguistics : System Demonstrations*. 2018. P. 5–9.
10. Knoblock C. A., Szekely P. Exploiting semantics for big data integration. *AI magazine.* 2015. Vol. 36, No. 1. P. 25–38. DOI: https://doi.org/10.1609/aimag.v36i1.2565
11. Ma C., Molnar B. Ontology Learning from Relational Database : Opportunities for Semantic Information Integration. *Vietnam Journal of Computer Science*. 2022. Vol. 9, No. 1. P. 31–57. DOI: https://doi.org/10.1142/s219688882150024x
12. Quirino G. K. S., Barcellos M. P., Falbo R. A. OPL-ML : A Modeling Language for Representing Ontology Pattern Languages. *Lecture Notes in Computer Science*. 2017. P. 187–201. DOI: https://doi.org/10.1007/978-3-319-70625-2_18
13. Ruy F. B., Guizzardi G., Falbo R. A., Reginato C. C., Santos V. A. From reference ontologies to ontology patterns and back. *Data & Knowledge Engineering.* 2017. Vol. 109. P. 41–69. DOI: https://doi.org/10.1016/j.datak.2017.03.004
14. Tilakaratna P., Rajapakse J. Conceptual and System Modeling with UML : Guidelines. *International Journal of Digital Content Technology and its Applications.* 2012. Vol. 6, No. 22. P. 90–97. DOI: https://doi.org/10.4156/jdcta.vol6.issue22.9
15. Tutcher J. *Development of semantic data models to support data interoperability in the rail industry :* Thesis of Dissertation. University of Birmingham, 2016. 355 p.
16. Shynkarenko V., Ilman V. M. Constructive-synthesizing structures and their grammatical interpretations. i. Generalized formal constructive-synthesizing structure. *Cybernetics and Systems Analysis.* 2014. Vol. 50. Iss. 5. P. 655–662. DOI: https://doi.org/10.1007/s10559-014-9655-z
17. Shynkarenko V., Zhuchyi L., Ivanov O. Conceptualization of the tabular representation of knowledge. *2021 IEEE 16th International Conference on Computer Sciences and Information Technologies.* 2021. P. 248–251. DOI: https://doi.org/10.1109/CSIT52700.2021.9648761
18. Shynkarenko V., Zhuchyi L. Constructive-synthesizing modeling of ontological document management support for the railway train speed restrictions. URL: https://tinyurl.com/2p9eeand
19. *Ukrainian railway train management rules.* 2005. URL: https://zakon.rada.gov.ua/rada/show/v0507650-05#Text
20. *Ukrainian railway train speed restriction form*. URL: https://images.app.goo.gl/mgrtncgyDKTsHbYF7
21. Xiao G., Lanti D., Kontchakov R., Komla- Ebri S., Güzel-Kalaycı E., Ding L., … Botoeva E. The Virtual Knowledge Graph System Ontop. *Lecture Notes in Computer Science.* 2020. P. 259–277. DOI: https://doi.org/10.1007/978-3-030-62466-8_17

doi: https://doi.org/10.15802/stp2022/268001

ISSN 2307–3489 (Print), ISSN 2307–6666 (Online)

Наука та прогрес транспорту, 2022, № 2 (98)

В. І. ШИНКАРЕНКО[1*], Л. І. ЖУЧИЙ[2*]

[1*]Каф. «Комп'ютерні інформаційні технології», Український державний університет науки і технологій, вул. Лазаряна, 2, Дніпро, Україна, 49010, тел. +38 (063) 489 49 15, ел. пошта shinkarenko_vi@ua.fm, ORCID 0000-0001-8738-7225
[2*]Каф. «Комп'ютерні інформаційні технології», Український державний університет науки і технологій, вул. Лазаряна, 2, Дніпро, Україна, 49010, тел. +38 (050) 965 92 36, ел. пошта larisa_zhuchiy@ukr.net, ORCID 0000-0002-9209-7262

Конструкційно-продукційне моделювання онтологічного забезпечення документообігу з обмеження швидкості руху поїздів

Мета. Під час розробки онтологій залізничного домену необхідно враховувати як дані інформаційних систем, так і нормативне забезпечення для перевірки їхньої узгодженості. Для цього виконують інтеграцію даних. Основною метою роботи є формалізація методів інтеграції різнорідних джерел інформації та формування онтології. **Методика**. Розроблено конструкційно-продукційну модель формування онтології та її джерел. **Результати**. Виконано формалізацію формування онтології, що дозволяє розширити можливості автоматизації інтеграції та узгодження даних засобами онтологій. Надалі заплановано розширити конструкційну систему формування онтологій на основі текстових джерел нормативної документації та інформаційних систем залізниці. **Наукова новизна**. Обґрунтовано можливості використання в онтологічному домені на залізничному транспорті конструкційно-продукційного моделювання для формування структури й даних таблиць попереджень про обмеження швидкості (форма бази даних і csv), їх перетворення в загальний табличний формат, словник, правила та екземпляри онтології, а також заповнення схеми екземплярами. Набули розвитку методи ontology learning для інтеграції даних різнорідних джерел. **Практична значимість**. Розроблені методи дозволяють виконати інтеграцію різнорідних джерел даних (структури таблиці ДУ–61 інструкції з руху поїздів, бланка та заявки на видачу попередження), що предметно орієнтовані на залізницю, сформувати онтологію від її джерел даних (форматів бази даних і csv) до схеми та екземплярів. Інтеграція та узгодженість даних інформаційних систем і нормативної документації є одним з аспектів підвищення рівня безпеки руху поїздів.

Ключові слова: конструкційно-продукційне моделювання; онтологія; інформаційна система; залізниця; база даних; таблиця; обмеження швидкості; онтологічне навчання

REFERENCES

1. Amardeilh, F. (2006). OntoPop or how to annotate documents and populate ontologies from texts. *ESWC 2006 Workshop on Mastering the Gap: From Information Extraction to Semantic Representation*, 1-16. (in English)
2. An, J., & Park, Y. B. (2018). Methodology for Automatic Ontology Generation Using Database Schema Information. *Mobile Information Systems*, *2018*, 1-13. DOI: https://doi.org/10.1155/2018/1359174 (in English)
3. Bischof, S., & Schenner, G. (2021). Rail Topology Ontology: A Rail Infrastructure Base Ontology. In *The semantic web - ISWC 2021* (pp. 597-612). DOI: https://doi.org/10.1007/978-3-030-88361-4_35 (in English)
4. Cimiano, P., & Völker, J. (2005). A Framework for Ontology Learning and Data-Driven Change Discovery. In *NLDB'05: Proceedings of the 10th international conference on Natural Language Processing and Information Systems* (Vol. 3513, pp. 227-238). DOI: https://doi.org/10.1007/11428817_21 (in English)
5. Crispino, G. (2003). *Une Plateforme Informatique de l'Exploration Contextuelle: Modelisation, Architecture et Realisation (ContextO). Application au Filtrage Semantique de Textes* (PhD dissertation). Universite De Paris Iv –Sorbonne. (in French)
6. Cruz, C., & Nicolle, N. (2008). Ontology enrichment and automatic population from XML data. In *Proceedings of the 4th International VLDB Workshop on Ontology-based Techniques for DataBases in Information Systems and Knowledge Systems* (pp. 1-5). (in English)
7. Dinşoreanu, M., Salomie, I., & Pop, C. B. (2011). Integrated System for Developing Semantically-Enhanced Archive Econtent. *Revista Romana de Informatica si Automatică, 21*(4), 67-77. (in English)
8. Fonseca, J. M. S. (2014). *Converting ontologies into DSLs* (PhD dissertation). Universidade do Minho Escola de Engenharia Departamento de Inform´atica. (in English)

doi: https://doi.org/10.15802/stp2022/268001

9. Klie, J. C., Bugert, M., Boullosa, B., de Castilho, R. E., & Gurevych, I. (2018). The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations* (pp. 5-9). (in English)
10. Knoblock, C. A., & Szekely, P. (2015). Exploiting semantics for big data integration. *AI magazine, 36*(1), 25-38. DOI: https://doi.org/10.1609/aimag.v36i1.2565 (in English)
11. Ma, C., & Molnar, B. (2022). Ontology Learning from Relational Database: Opportunities for Semantic Information Integration. *Vietnam Journal of Computer Science, 9*(1), 31-57. DOI: https://doi.org/10.1142/s219688882150024x (in English)
12. Quirino, G. K. S., Barcellos, M. P., & Falbo, R. A. (2017). OPL-ML: A Modeling Language for Representing Ontology Pattern Languages. In *Lecture Notes in Computer Science* (pp. 187-201). DOI: https://doi.org/10.1007/978-3-319-70625-2_18 (in English)
13. Ruy, F. B., Guizzardi, G., Falbo, R. A., Reginato, C. C., & Santos, V. A. (2017). From reference ontologies to ontology patterns and back. *Data & Knowledge Engineering, 109,* 41-69. DOI: https://doi.org/10.1016/j.datak.2017.03.004 (in English)
14. Tilakaratna, P., & Rajapakse, J. (2012). Conceptual and System Modeling with UML: Guidelines. *International Journal of Digital Content Technology and its Applications, 6*(22), 90-97. DOI: https://doi.org/10.4156/jdcta.vol6.issue22.9 (in English)
15. Tutcher, J. (2016). *Development of semantic data models to support data interoperability in the rail industry* (PhD dissertation). University of Birmingham. (in English)
16. Shynkarenko, V., & Ilman, V. M. (2014). Constructive-synthesizing structures and their grammatical interpretations. i. Generalized formal constructive-synthesizing structure. *Cybernetics and Systems Analysis, 50*(5), 655-662. DOI: https://doi.org/10.1007/s10559-014-9655-z (in English)
17. Shynkarenko, V., Zhuchyi, L., & Ivanov, O. (2021). Conceptualization of the tabular representation of knowledge. In *2021 IEEE 16th International Conference on Computer Sciences and Information Technologies* (pp. 248-251). DOI: https://doi.org/10.1109/CSIT52700.2021.9648761 (in English)
18. Shynkarenko, V., & Zhuchyi, L. Constructive-synthesizing modeling of ontological document management support for the railway train speed restrictions. (n. d.). URL: https://tinyurl.com/2p9eeand (in English)
19. *Ukrainian railway train management rules.* (2005). URL: https://zakon.rada.gov.ua/rada/show/v0507650-05#Text (in English)
20. *Ukrainian railway train speed restriction form*. URL: https://images.app.goo.gl/mgrtncgyDKTsHbYF7 (in English)
21. Xiao, G., Lanti, D., Kontchakov, R., Komla- Ebri, S., Güzel-Kalaycı, E., Ding, L., … & Botoeva, E. (2020). The Virtual Knowledge Graph System Ontop. *Lecture Notes in Computer Science*, 259-277. DOI: https://doi.org/10.1007/978-3-030-62466-8_17 (in English)

Received: 18.02.2022
Accepted: 20.06.2022