Archives of
Transport System
Telematics

Volume 10

Issue 4

November 2017

RailTopoModel and RailML – data exchange standards in railway sector

T. CISZEWSKI, W. NOWAKOWSKI, M. CHRZAN
UNIVERSITY OF TECHNOLOGY AND HUMANITIES IN RADOM, Faculty of Transport and Electrical Engineering, Malczewskiego 29, 26-600 Radom, Poland
EMAIL: t.ciszewski@uthrad.pl

ABSTRACT

RailTopoModel is a tool for logical define object model of the data related to railway infrastructure. In April 2016, the standard was released by the UIC (International Union of Railways) as International Railway Standard IRS 30100:2016. RailML has been developed since 2002. It is based on the XML - meta-language that is used to define other languages. RailML is used as a data exchange format in railway systems. It includes within its scope data relating to railway traffic management, rolling stock management, stacking timetables, information for passengers, booking and selling tickets. RailML is currently available in version 2.3. Its upcoming version 3.0 has been especially designed to ensure compliance with RailTopoModel and will be the first practical implementation of this standard. Additionally interlocking schema will be included to the RailML 3.0 schema. The authors pay attention to the importance of both RailML and RailTopoModel specification. The current state of the standards and their prospects are discussed. The authors also present their own software which allows edit and validate RailML files.

KEYWORDS: RailTopoModel, IRS 30100, RailML, XML, data exchange standards, railway IT systems

1. Introduction

Nowadays, the operation of modern railway systems without the use of information technology is impossible. Their use allow us not only to optimize operating costs [1, 2, 3], but also enable multidimensional cooperation of companies operating in a wide variety of railway business areas. This necessity of collaboration of many diverse and independent systems [4, 5] that work for numerous railway companies forces to design and maintain a lot of interfaces, which are used only to convert and exchange data between these systems. At the same time the problem of ensuring the interoperability of systems is compounded by the lack of standards of protocols and data structures which are necessary for information exchange [3, 5].

2. The problem of data exchange between systems

The variety of data structures, which are used in the cooperating systems enforces data conversion, whenever they are exchanged between applications (Fig. 1). However, the conversion process is costly and time-consuming and decreases the efficiency of the system. In addition, for each pair of cooperating systems, the model and the format of the exchanged data must be defined as well as the additional technical aspects of information exchange must be specified.

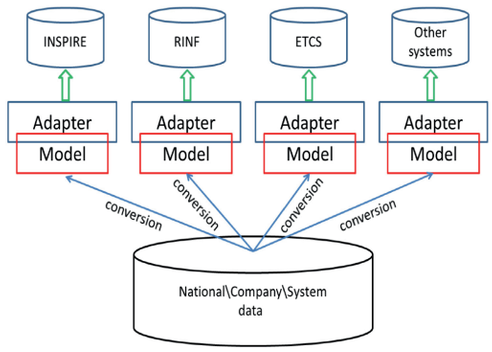

Fig. 1. In case of the lack of data format standardization the cooperation with other system requires each time the data conversion into a new format [own study]

The data model defines objects and their topological structure and determines their attributes. The selected data format is one of the possible model data representation that is used by the particular system. Adapters (interfaces) allow to convert this format of the data into a common data exchange format, thus enabling system cooperation (Fig. 2).

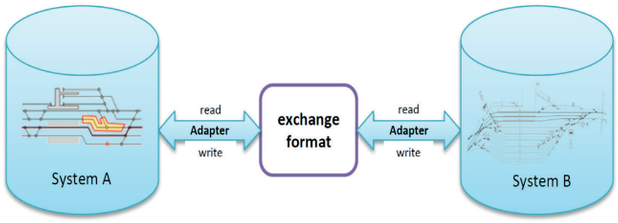

Fig. 2. The data exchange between two cooperating systems [own study]

If we assume that n particular systems need data transfer between them then the number of necessary interfaces will be L [6]:

$$L = n(n-1)$$

For example, if we assume that there is a need to exchange data between 6 various railway systems it means, in accordance with the above equation, the need to design and maintain 30 interfaces. This example clearly shows a great need for unifying data structures, because in general terms, if the number of programs increases linearly, the number of adapters grows quadratically. If we realize that the number of railway applications is steadily increasing the task of unifying data structures becomes even more significant.

The data structures standardization would allow to solve data exchange issues. Then the number of adapters L will be equal to the number of cooperating railway applications [6]:

$$L = n$$

In this case, multiple data conversion is not needed (Fig. 3). It should be noticed that in case of standardization the requirements for the model and data format must consider the purpose of data usage and existing tools as well as should be independent of the interfaces.

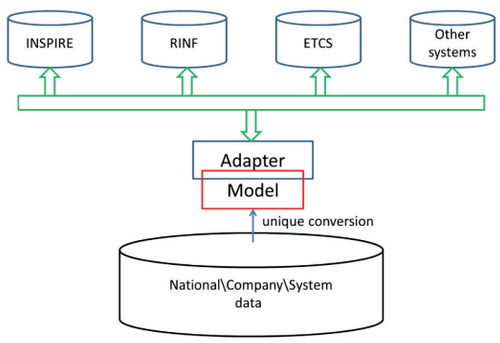

Fig. 3. Cooperation between systems in case of unifying data model and its format [own study]

3. RailML

An attempt to standardise data structures for cooperating railway systems must take into consideration the modern data formats used in information technology. The most important of them are the ASN.1 (Abstract Syntax Notation 1) [7, 8, 9], and XML (Extensible Markup Language) [11]. The last one is an open standard developed by the W3C (World Wide Web Consortium) and is supported by software market leaders.

XML is a language that describes data, in other words, meta-language. In short it can be said that XML is used for describing other languages (XML application) in order to store information. Examples of such applications can be numerous XM-based standards: XHTML, RSS, MathML, CML, FpML, ebXML, GML, SVG, MusicML, SMIL, RDF, OWL, TransXML as well as described in this article Railway Markup Language (RailML). In all these examples, XML as a data exchange mechanism plays a crucial role.

Because XML simultaneously store data and describe their structure this format is very efficient. The XML documents are divided into markup and content, that may be distinguished by the simple syntactic rules. XML itself does not define a set of tags and attributes, so that they can be specifically defined and matched to the particular application. At the same time, we define by ourselves a data structure that can be tabulated, but can also adopt a tree form.

Built-in namespace mechanism provides easy extension of existing documents while maintaining forward and backward compatibility. Several methods that allow to define the syntax, structure and content validation can be used. These include, inter alia: Document Type Definition (DTD) language, the W3C XML Schema (XSD), RELAX NG, which is part of Document Schema Definition Languages (DSDL) specification, Schematron and Namespace Routing Language (NRL). XML Schema Definition is recommended by W3C and currently the most popular standard which describes the elements and their attributes that can appear in an XML files. This definition also specifies child elements, data types and default values for elements and their attributes [11]. This tool (XML Schema) is used by the authors to validate RailML files in dedicated editor. This capabilities of prepared software will be described later in this article.

RailML is open XML-based standard which is being developed since 2002, initially by a group of researchers from German Fraunhofer Institute for Transportation Systems and Infrastructure in Dresden and the Swiss Federal Institute of Technology's Institute for Transportation Planning and Systems [11]. At present, the RailML Consortium was extended to include researchers from several universities, railroad operating companies, private research institutes, and consulting firms. RailML standards are developed in the context of technical discussions that are open to everyone interested in developing applications for the international railroad industry. The Fraunhofer Institute serves as the partnership's technical coordinator providing resources such as the web page and discussion forum [12]. The first stable version 1.0 was released in 2005 for productive usage. A version 2.3 published in 2016 is the latest production version. The publication of the RailML 3 final version which will consider the RailTopoModel standard (International

Railway Standard - IRS 30100) developed under the auspices of UIC has been scheduled for 2017 [3, 12, 13].

RailML was designed to enable cooperation of railway applications. Communication between them that use RailML can be performed using two alternative methods. In the first method RailML can serve as a standardized format for data exchange that was mentioned above. In this case, applications can export and import RailML files, however, internally still use their own data formats. In some cases, some of the applications can use RailML as a native data structure. Alternatively, the direct communication based, for example, on TCP / IP can be used. This method is particularly suitable for those pairs of systems that use RailML as native data format - then no additional data import\export filters do not have to be implemented.

In RailML similarly to all XML documents, the hierarchical - tree structure of the document is used, in which the root is the <railml> tag. The root is the parent of all nested elements that hierarchically may contain subelements (children). Subelements must be in pairs and correctly nested within their parent element. All elements can have specified additional attributes that enable their more precise description. In the RailML 2.3 specification the root may include three types of subelements, <infrastructure>, <rollingstock> and <timetable> (Fig. 4).

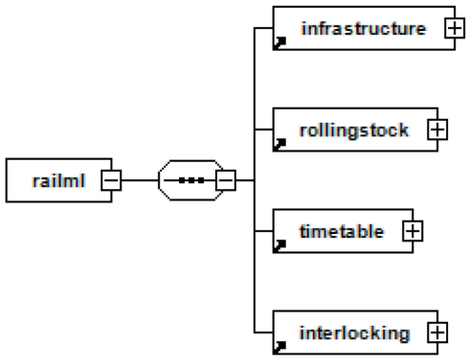

Fig. 4. Main subelements of <railml> root (RailML 3 developer preview version) [own study based on 3, 12]

The <Infrastucture> tag defines, among other things, the following subelements of infrastructure: <infraAttrGroups>, <tracks>, <trackGroups>, <controllers>, <operationControlPoints>, <speedProfiles>, <visualizations>.

The <rollingstock> tag defines, inter alia, the following subelements: <vehicles> and <formations>. They are used to describe the data, the structure and parameters of the rolling stock. A practical example of the trains description using subelements and attributes of the <rollingstock> is shown in Fig. 5.

The next of the elements, that is shown in Fig. 4, it is <timetable>. It contains data structures suitable for describing timetables. The following tags are defined as <timetable> children: <timetablePeriods>, <operatingPeriods>, <categories>, <annotations>, <trainParts>, <trains>, <trainGroups> and <rosterings> [12].

In the next standard version - RailML 3 - the schema extension was announced. The interlocking information will be included. The developer RailML3 version describes, inter alia, following information (Fig. 4): signal plans, blocks and route locking table [14, 15]. However, the main reason for the release the new RailML specification is to ensure compliance with the described below RailTopoModel standard (IRS 30100).

```
<rollingstock id="vn06">
  <vehicles>
    <vehicle id="vn10" name="EN62" length="58.95"
             speed="160" bruttoWeight="107" />
    <vehicle id="vn12" name="ET22" length="19.24"
             speed="125" bruttoWeight="120" />
    <vehicle id="vn14" name="HCP111A" length="24.5"
             speed="160" bruttoWeight="39.5" />
  </vehicles>
  <formations>
    <formation id="vn19" name="EN62">
      <trainOrder>
        <vehicleRef orderNumber="1" vehicleRef="vn10" />
      </trainOrder>
    </formation>
    <formation id="vn27"
               name="ET22 with 4 cars">
      <trainOrder>
        <vehicleRef orderNumber="1" vehicleRef="vn12" />
        <vehicleRef orderNumber="2" vehicleRef="vn14"
                    vehicleCount="4" />
      </trainOrder>
    </formation>
  </formations>
</rollingstock>
```

Fig. 5. Sample <rollingstock > section of RailML file [own study]

4. RailML File Editor

For purposes of their analysis the authors have built their own software RailML File Editor (Fig. 6). It allows to view and edit RailML files (.xml) and XML Schema Definition (.xsd). They can be saved in native format and exported to HTML, Rich Text Format (.rtf) and Adobe Portable Document Format (.pdf) (Fig. 7).

The software also cooperate with the open source tool - XSDDiagram [15] and in the case of viewing or editing XML Schema Definition can generate graphical visualization of the schema diagrams (Fig. 4).

The view mode allows to present a document in the form of a tree, and after selecting one of its nodes allow to view its attributes and values. In edit mode the possibility of RailML syntax highlighting as well as the convenient tools to search and navigate in the document are available. What is important, this software allows to validate RailML documents and XSD schemas.

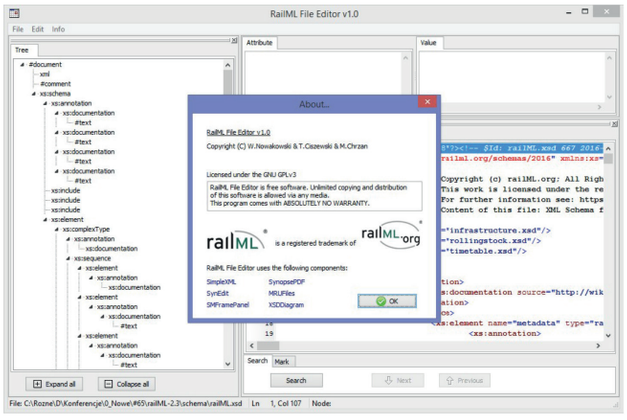

Fig. 6. RailML File Editor main window [own study]

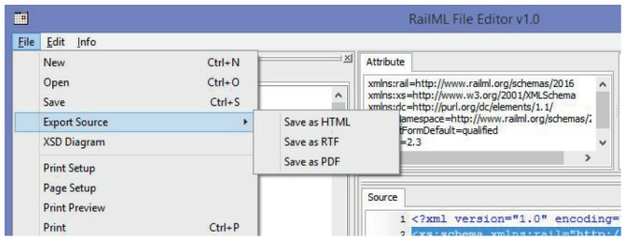

Fig. 7. Export formats of RailML File Editor [own study]

The software is also equipped with the ability to print both kinds of documents and the convenient print preview (Fig. 8).

5. RailTopoModel

According to [17] topology is the study of geometrical properties and spatial relations unaffected by the continuous change of shape or size of figures. Topology is usually we used to abstract the inherent connectivity of objects while ignoring their detailed forms. In technology, topology is a tool that allows to show how the system constituent parts are interrelated or arranged, which is very useful in the transformation the reality into a model of the system (Fig. 9). The proposed model is based on graph theory and therefore is independent from the type of the represented objects and their properties. The objects in the designed graph are represented as nodes or edges and called NetElements.

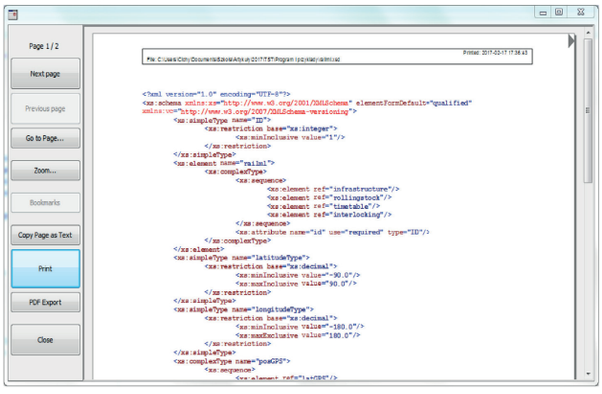

Fig. 8. Print preview window in RailML File Editor [own study]

The main purpose of modelling is to define and describe railway objects as well as system events in order to show how they can be used, and how they cooperate and interact.

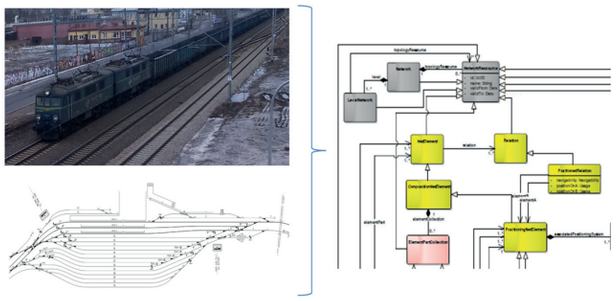

Fig. 9. Railway topology modelling [own study]

In 2013, under the aegis of ERIM (European Rail Infrastructure Masterplan) Task Force of the UIC feasibility study of international infrastructure model was conducted. The research projects checked also the possibility of creating an universal data exchange format [18]. In this document the analysis of models used by companies operating in the railway market and available data exchange formats were conducted. Special attention was paid to available at that time and matured the RailML specification.

This analysis [11, 18] led to the fallowing conclusions:

- because iron network is similar in every country most of the features in analysed topological models are compatible,
- previously designed topological models were created not as universal, but in view of their specific applications,
- systemic and scalable core model would the most appropriate,
- the amount of available data and its accuracy may vary significantly depending on the railway company and the country,
- the designed model should support data with varying degrees of detail (Tab. 1). The description can be as general as corridors; it can be brought at line level, track level, down to physical components such as switches, lineside signals or balises [19].
-

Table 1. RailTopoModels - predefined levels of details [11, 18]

Level of details	Supported data
micro (detailed level)	track geometry, signalling, ETCS
meso (track level)	train dispatching
macro (line level)	timetabling, stations
Corridor (international level)	cartography, economical analysis

Structuring of functional and organizational requirements as well as content of the model were proposed. As a modelling tool UML diagrams were selected [20, 21], and using them has allowed to define RailTopoModel . This standard according to the authors of the above-mentioned feasibility study should take into account, inter alia, following recommendations [11, 18, 19]:

- the UIC RailTopoModel should be a minimal core topology model,
- model should be extensible,
- only standardized model and data exchange format can ensure the interoperability of the railway IT systems,
- the model should take into account and apply the existing standards [22],
- designed standard should enable the possibility of gradual development and the layers of the model should be enriched in stages in order to keep up with needs of the railway industry.

It was also considered [18] that the RailML is the appropriate format, which allows to store and exchange the necessary data and meets all the requirements in this regard. At the same time limitations of the format were noted, especially in terms of the topology model and support tools.

Finally RailTopoModel specification was published as an International Standard Railway IRS 30100:2016 in September 2016. Within its framework, in the form of UML diagram, the objects occurring at different levels of the railway network were defined as classes. All object reference to the topology of the railway network at the appropriate level of aggregation. The components that are not

directly elements of the railway network, but have an impact on it were also included in this standard. In the scope of the specification the varied methods of objects positioning were included [18]. Functional objects, dimensions and properties in RailTopoModel were presented in Fig. 10.

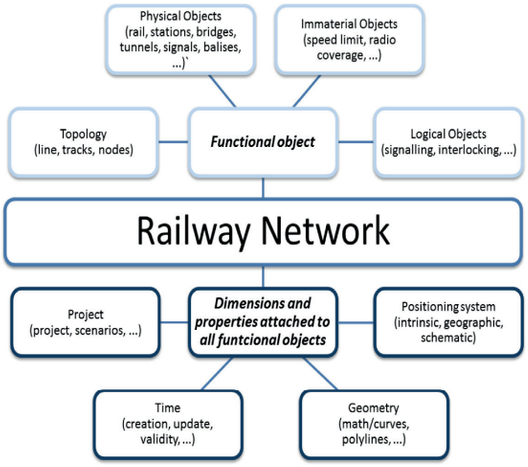

Fig. 10. RailTopoModel - functional objects, dimensions and properties [own study based on 11, 19]

6. Conclusion

In railway applications of IT systems the proprietary and mutually incompatible solutions have been used by manufacturers for many years. These IT systems have worked in different areas of the companies' activities. Although their application delivers considerable benefits in the areas of their usages, the mutual incompatibility and closed formats cause a lot of problems.

The holistic way of looking at the railway operating, the use of universal standards for the description of the railway infrastructure and the use of open data formats which are producer-independent and application-independent, may transfer the operation of IT systems to a whole new level of cooperation. The very first versions of the RailML showed that such universal approach to certain tasks, such as traffic management, rolling stock management, stacking timetables, information for passengers, booking and selling tickets is very practical. Further versions of the standard progressively increases the number of possible applications (for example, interlocking data). The number of implementations of RailML standard in software used in railway operations is still growing, especially in this software which is produced by the members of RailML consortium.

Inspired by the growing popularity of the standard the authors began work on their own software to edit RailML files. The result of this work is shown in this text the first version of RailML File Editor. This software still requires intensive development, but for now, it is useful for users who need view\edit RailML files or XSD schemas. An interesting possibility for further development of this software, which is considered by the authors, is to allow the import and export data to and from various formats used by the railway software. The possibility of implementation of these functions is determined by accessibility to specifications of proprietary data formats used in such software.

RailTopoModel specification as well as IRS standard 30100: 2016, which is final outcome of international work on response to the specific needs of railway companies related to the necessity of multi-level description of the structure and topology of the railway network. As the standard defines only the model and provides the documentation it remains completely independent of its possible applications and its practical implementation. The very first implementation of this model will be probably, announced for the current year, RailML 3 specification, which will be released especially to ensure compliance with the requirements of IRS 30100: 2016. It will define a new, universal data exchange format which is based on RailTopoModel, and is eagerly awaited on the market. The availability of a common data model will facilitate the development of software, processes and services for the railway industry. Unify data structures used by these systems and availability to a common data exchange format (RailML) in authors' opinion will probably revolutionize the data flow in the railway industry.

Acknowledgment

This material is based upon work supported by National Centre for Research and Development under Grant No. PBS3/A6/29/2015 entitled „The system for maintenance data acquisition and analysis of reliability and safety of traffic control systems".

Bibliography

[1] HANNING, W., WEIXIANG, X., CHAOLONG J.: A data structure model supporting railway distributed system integration, in ASME/ASCE/IEEE Joint Rail Conference, Pueblo, Colorado, USA, 2011.

[2] ŁUKASIK, Z., NOWAKOWSKI, W., CISZEWSKI, T.: Definition of data exchange standard for railway applications, Prace Naukowe Politechniki Warszawskiej - Transport, no. 113, pp. 319-326, 2016.

[3] ŁUKASIK, Z., NOWAKOWSKI, W.: Automatyczne pozyskiwanie informacji z systemów sterowania ruchem kolejowym do systemu SEPE, Logistyka, vol. 3, pp. 4006-4010, 2014.

[4] MONTIGEL, M.: Time-triggered exchange of train route data between train control systems, in 8th International Conference on Computer Aided Design, Manufacture and Operation in the Railway and Other Advanced Mass Transit Systems (COMPRAIL) Lemnos, Greece, Southampton, 2002.

[5] KORNASZEWSKI M., ŁUKASIK, Z.: The interoperability of European rail transport from Polish authority of messages of view, in International Conference Globalization – Social and Economic Impacts '08, Rajecké Teplice, Slovak Republic, 2008.

[6] NASH, A., et al.: RailML - a standard data interface for railroad applications, in Proc. of the Ninth International Conference on Computer in Railways (Comprail IX), Dresden, Germany, 2004.

[7] ŁUKASIK, Z., NOWAKOWSKI, W.: Application of TTCN-3 for testing of railway interlocking systems, in Communications in Computer and Information Science, vol. 104, Berllin Heiidelberg, Springer Science & Business Media, 2010, pp. 447-454.
[8] ŁUKASIK, Z., NOWAKOWSKI, W.: ASN. 1 notation for exchange of data in computer-based railway control systems, Transport Problems, vol. 4, no. 2, pp. 111-116, 2009.
[9] ŁUKASIK, Z., NOWAKOWSKI, W.: Designing communication software for computer railway control systems with the use of ASN.1, in Computer Systems Aided Science and Engineering Work in Transport, Mechanics and Electrical Engineering, vol. 122, Radom, Technical University of Radom, 2008, pp. 391-398.
[10] HAROLD, E.R.: XML 1.1 Bible, 3rd Edition ed., Indianapolis: iley Publishing, Inc., 2004.
[11] CISZEWSKI T., NOWAKOWSKI, W.: Interoperability of it systems in the international railways, in Proceedings of the 16th International Scientific Conference Globalization and its socio-economic consequences, Rajecke Teplice, Slovak Republic.
[12] railML.org, 2015. http://www.railml.org. [date of access: 17.02.2017].
[13] RailTopoModel, 2016. [Online]. Available: http://www.railtopomodel.org. [date of access: 17.02.2017].
[14] BOSSCHAART, M.: Lean Engineering Design of Rail Interlocking Systems with RailML, Thesis Report Master TIL ed., Delft: Delft University of Technology, 2013.
[15] BOSSCHAART, M., et al.: Efficient formalization of railway interlocking data in RailML, Information Systems, vol. 49, pp. 126-141, April 2015.
[16] COSNIER, R.: XSD Diagram, 2006. [Online]. Available: http://regis.cosnier.free.fr /?page=XSDDiagram. [January 30, 2017].
[17] English Oxford Living Dictionares, [Online]. Available: https://en.oxforddictionaries.com/definition/topology. [date of access: 19.01.2017].
[18] UIC, Feasibility study. UIC RailTopoModel and data exchange format, 2013. [Online]. Available: http://www.railtopomodel.org/files/download/RailTopoModel/ 270913_trafIT_FinalReportFeasibilityStudyRailTopoModel.pdf. [date of access: 30.01.2017].
[19] "IRS 30100, RailTopoModel Railway infrastructure topological model," 2016. [Online]. Available: http://www.railtopomodel.org/en/homepage.html. [February 15, 2017].
[20] "UML," 2017. [Online]. Available: http://www.omg.org/spec/UML/2.5/. [date of access: 30.01.2017].
[21] DENNIS, A., WIXOM, B.H., RORH, R.M.: System Analysis Design UML Version 2. An object-oriented approach., 6th ed., John Wiley & Sons, 2015.
[22] DOPPELBAUER, J.: Regulation and standardisation in the European rail sector, Elektrotechnik und Informationstechnik, vol. 132, no. 7, pp. 384-388, 2015.