Zbigniew Łukasik, Waldemar Nowakowski, Tomasz Ciszewski
Uniwersytet Technologiczno-Humanistyczny im. Kazimierza Pułaskiego w Radomiu,
Wydział Transportu i Elektrotechniki

DEFINITION OF DATA EXCHANGE STANDARD FOR RAILWAY APPLICATIONS

The manuscript delivered: March 2016

Abstract: Railway similar to the other branches of economy commonly uses information technologies in its business. This includes, inter alia, issues such as railway traffic management, rolling stock management, stacking timetables, information for passengers, booking and selling tickets. Variety aspects of railway operations as well as a large number of companies operating in the railway market causes that currently we use a lot of independent systems that often should work together. The lack of standards for data structures and protocols causes the need to design and maintain multiple interfaces. This approach is inefficient, time consuming and expensive. Therefore, the initiative to develop an open standard for the exchange of data for railway application was established. This new standard was named railML. The railML is based on Extensible Markup Language (XML) and uses XML Schema to define a new data exchange format and structures for data interoperability of railway applications. In this paper the current state of railML specification and the trend of development were discussed.

Keywords: railway traffic control systems, railML, XML

1. INTRODUCTION

It is hard to imagine the functioning of the modern world without information technologies. It is a result of numerous advantages of the modern IT solutions. One of the most important arguments for using IT systems is cost optimisation [1, 3, 6]. Variety aspects of railway operations as well as a large number of companies operating in the railway market causes that currently we use a lot of independent systems that often should cooperate. The lack of standards for data structures and protocols causes the need to design and maintain multiple individual interfaces that provide interoperability of railway applications. This problem occurs in all countries of the European Union. Therefore, the initiative to develop an open standard for the exchange of data for railway application was established. This new standard was named railML [9]. This standard was developed in cooperation with the railway, traffic and standardization sector e.g. Union Internationale des Chemins de Fer (UIC), European Organisation for the Safety of Air Navigation (EUROCONTROL), European Railway Agency (ERA). In this paper the current state of railML specification and the trend of development were discussed.
2. DATA EXCHANGE FOR RAILWAY APPLICATIONS

Railway similar to the other branches of economy commonly uses information technologies (IT) in its business. It can be considered that this is a requirement of the modern world, which try to build an information society capable of competing on the global market. The growing importance of IT systems in railway transport is caused by large number of companies which offer their specialised software. The lack of standards for data structures and protocols causes the need to design and maintain multiple individual interfaces that provide interoperability of railway applications. The variety of data structures is the characteristic of the system, and cooperation with other system requires data conversion to the new format each time. Data conversion is inefficient, time-consuming and expensive process. If we assume that \( n \) particular systems need data transfer between them then the number of necessary bidirectional interfaces will be \( L \) [9]:

\[
L = \frac{n(n-1)}{2}
\]  

(1)

In general terms, if the number of programs increases linearly, the number of interfaces increases quadratically. The unification of data structures would allow to solve this problem. Then the number of interfaces will be equal to the number of cooperating applications (2).

\[
L = n
\]  

(2)

For example, if we assume that there is a need to exchange data between 6 different railway applications it means, according to equation (1), a need to develop 15 bidirectional interfaces. If we assume that the data structures are standardised the number of new interfaces to be developed is reduced to 6. This example clearly shows a great need for unifying data structures. This becomes even more significant if we notice that the number of railway applications is steadily increasing.

3. XML

The data structure specification for cooperating railway applications should take into consideration modern IT standards. Currently there are several such standards. The most important are XML (Extensible Markup Language) and ASN.1 (Abstract Syntax Notation One) [4, 5, 7, 8]. XML is open W3C (World Wide Web Consortium) standard supported by software industry market leaders, and based on international standards. What is important, XML allows the simultaneous data storage and data structure description. The XML documents are divided into markup and content, that may be distinguished by the simple syntactic rules. XML Schema Definition is recommended by W3C and currently
the most popular standard [10] which describes the elements and the attributes that can appear in an XML document. This definition specifies also the number of child element, data types and default values for elements and attributes. Developing data schema make possible syntax, structure and content validation of XML documents. Based on the XML specification many new markup languages were developed and acknowledged as standards, for example xHTML, RSS, MathML, CML, FpML, ebXML, GML, SVG, MusicML, SMIL, RDF, OWL, TransXML as well as described in this article Railway Markup Language (railML).

4. RAILML STANDARD

RailML is open XML-based standard for railway application. Applications can exchange data via railML either via exporting respectively importing railML files, or as a direct inter-process communication via TCP/IP. This standard 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. 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 railroad industry. The Fraunhofer Institute serves as the partnership’s technical coordinator providing resources such as the web page and discussion forum [11]. 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 3.0 version which will consider the RailTopoModel standard (International Railway Standard - IRS 30100) developed under the auspices of UIC is planned [12].

As mentioned above the railML data structure is defined using XML Schema. Similar to all XML documents hierarchical form is used. Root element is the parent of all other elements. All elements can have sub (child) elements. Sub elements must be in pairs and correctly nested within their parent element. Additionally, all elements can have attributes

![Fig. 1. Root element <railml> and its sub elements](image-url)
for more detailed description. In case of railML the root element is named <railml> and contains three sub elements <infrastructure>, <rollingstock> and <timetable> (Fig. 1).

Element <infrastructure> allows to save in XML files railway infrastructure parameters, among others, the following elements (Fig. 2, 3):
- <infraAttrGroups> (owner, operationMode, trainProtection, electrification, powerTransmission axleWeight gauge, clearanceGauge, speeds, epsgCode, trainRadio, generalInfraAttributes),
- <tracks> (states, trackDescr, trackTopology, trackElements, ocsElements, infraAttrGroupRefs),
- <trackGroups> (line, locallyControlledArea),
- <operationControlPoints> (propOperational, propService, propEquipment, propOther, tsi, area, geoCoord, designator),
- <controllers> (states),
- <speedProfiles> (tilting, braking, path).

![Fig. 2. Element <infrastructure> and its sub elements](image)

<infrastructure id="d2e4">
  <tracks>
    <track id="d2e8">
      <trackTopology>
        <trackBegin pos="0" id="d2e12">
          <macroscopicNode ocpRef="d2e311" />
        </trackBegin>
        <trackEnd pos="300" id="d2e24">
          <macroscopicNode ocpRef="d2e313" />
        </trackEnd>
      </trackTopology>
    </track>
  </tracks>
  <operationControlPoints>
    <ocp id="d2e12" name="FD" description="Warszawa" />
  </operationControlPoints>
</infrastructure>

![Fig. 3. Sample <infrastructure> section in RailML format](image)
Another element is the `<rollingstock>` which defines XML data structures designed to save rolling stock data. It includes inter alia the following elements (Fig. 4, 5):
- `<vehicles>` (classification, engine, wagon, vehicleBrakes, loadLimitMatrix, curvingLimitation, maintenanceIntervals),
- `<formations>` (trainOrder, categoryRef, trainEngine, trainBrakes, trainResistance).

![Fig. 4. Element <rollingstock> and its sub elements](image)

```
<rollingstock id="d2e16">
  <vehicles>
    <vehicle id="d2e20" name="EU07" length="15.92" speed="120" bruttoWeight="83.5" />
    <vehicle id="d2e22" name="EP09" length="16.74" speed="160" bruttoWeight="83.5" />
  </vehicles>
  <formations>
    <formation id="d2e29" name="EP07">
      <trainOrder>
        <vehicleRef orderNumber="1" vehicleRef="d2e20" />
      </trainOrder>
    </formation>
    ...
  </formations>
</rollingstock>
```

![Fig. 5. Sample <rollingstock> section in RailML format](image)

The last element is `<timetable>`, which specifies the timetables data structures. It includes, inter alia (Fig. 6, 7):
- `<timetablePeriods>` (holidays),
- `<operatingPeriods>` (operatingDay, specialService),
- `<categories>`,
- `<annotations>` (text),
- `<trainParts>` (formationTT, operatingPeriodRef, ocpsTT, organizationalUnitBinding, annotationRef),
- `<trains>` (trainPartSequence, tafTapTsiTrainID),
- `<trainGroups>` (trainRef),
- `<rosterings>` (blockParts, blocks, circulations).
Fig. 6. Element <timetable> and its sub elements

<timetable id="d2e49">
  [...]  
  <trains>
    <train id="d2e1161" name="651 RB 303" trainNumber="303" type="operacyjny" description="Lajkonik">
      <trainPartSequence sequence="1">
        <trainPartRef ref="d2e119"/>
      </trainPartSequence>
    </train>
  [...]  
  </trains>
  <rosterings>
    <rostering id="d2e1517" formationRef="d2e29">
      <blockParts>
        <blockPart id="d2e1529" name="148" formationRef="d2e29" mission="rozklad" startOcpRef="d2e6" endOcpRef="d2e10" trainPartRef="d2e119" runLength="302.7" begin="05:40:18" end="08:55:24"/>
      [...]  
      </blockParts>
      <blocks>
        <block id="d2e1628" blockGroupNumber="1">
          <blockPartSequence sequence="1">
            <blockPartRef ref="d2e1529"/>
          </blockPartSequence>
        </block>
      [...]  
      </blocks>
      [...]  
    </rostering>
    [...]  
  </rosterings>
  [...]  
</timetable>

Fig. 7. Sample <timetable> section in RailML format
5. CONCLUSIONS

The application of IT systems on the railway is related to number of problems and issues. This includes, inter alia, issues such as railway traffic management, rolling stock management, stacking timetables, information for passengers, booking and selling tickets. Variety aspects of railway operations causes that we use a lot of independent IT systems that often should cooperate. Additionally each of the producers offers their own railway applications what increase the number of solutions. This variety of solutions enforces the need to develop an open and universally usable standards for defining data structures. Unified data structures, which are application-independent and producer-independent, should simplify data exchange between railway systems and reduce number of interfaces which are required for systems cooperation. The railML, described in this paper, is an example of the standard which meets these requirements. This standard is being still developed by the railML Consortium and supported by a large number of companies operating on the European market e.g. ABB, Alstom, Bombardier Transportation, Siemens, DB Engineering & Consulting, Thales, Toshiba Corporation. These companies use railML standard in own solutions offered on the market or allow to import/export data using railML format. Nowadays railML Schema includes three subschemas: infrastructure, rollingstock and timetable. Interlocking subschema is currently intensively developed [2].

References

8. Łukasik Z., Nowakowski W.: Struktury danych i ich kodowanie w systemach sterowania ruchem kolejowym z uwzględnieniem notacji ASN.1, Transport Miejski i Regionalny, Nr 12, Grudzień 2009r.
11. https://www.railml.org
DEFINICJA STANDARDU WYMIANY DANYCH DLA ZASTOSOWAŃ KOLEJOWYCH


Słowa kluczowe: systemy sterowania ruchem kolejowym, railML, XML

ACKNOWLEDGEMENT
The paper was carried out as a part of the research project of the National Centre for Research and Development NCBiR NR PBS3/A6/29/2015, titled „The system for maintenance data acquisition and analysis of reliability and safety of traffic control systems".