railML® – In OpenTrack And In Other Tools

IT15.rail, OpenTrack Userworkshop
June 12th, 2015
Dr. Bernhard Seybold, trafIT solutions
Outline

• Introduction to railML
• railML in OpenTrack
• Other railML tools

© railML is a registered trademark of railML.org
Some slides courtesy of Vasco Kolmorgen, railML.org
About trafIT solutions

Founded 2009, Zurich

We write software
- OnTime, the timetable quality tool
- Tools for OpenTrack (OTH, OTD, IA2OT)
- railOscope: railML-validation und –visualization, other railML tools
- TreMOla: Optimization of maintenance schedules in the GBT
- LayoutLib: Automatic generation of schematic graphs
- Custom-tailored tools for railway companies

We analyze data
- Timetables/operation: Quality, Stability, Punctuality
- Service concepts: key figures, accounting
- Railway data in several formats (railML, HAFAS, VDV, Info+, ...)

We do research & development
- RTSE (closed-loop railway disposition, founded by Swiss government)
- Prototype for automatic generation of periodic timetables
railML Motivation

- Railway models are large, programs need to exchange data
- First: point to point integration. Some tools define import or export interface. Other tools produce those formats.
- \( \rightarrow \) Interface explosion (quadratic)
- Need one model for all: birth of railML
railML Overview

- Exchange format for railway data based on xml
- Open-Source, current version is 2.2
- 10 years of development
- Supported by UIC, ERA/RINF

- Supports timetable, infrastructure, rolling stock, interlocking
- Active community with bi-annual meetings
- Only registration at railml.org needed
Subschemes

- **railML organized into subschemes:**

  ![Diagram showing railML and subschemes]

- **Additional subschemes:**
  - *station facilities*: On hold, currently no requirements from users.
  - *crew rostering*: Data is being gathered; railML.org-working group was established.
  - *interlocking*: railML.org-working group active with regular meetings, Compilation of elements, allowing connection to existing subschemes achieved. First use case: Interlocking data for ETCS.
  - ...

railML Content

- **Timetable scheme:**
  - trains, train groups, rosterings
  - operating periods, timetable periods

- **Infrastructure scheme:**
  - lines, tracks, operation points
  - track elements (signals, switches, tunnels, bridges, kilometrization, balises, maxSpeed, electrification, radius, ...)
  - coordinates (geo and sheet coordinates)

- **Rolling Stock scheme:**
  - vehicles (engines, wagons), formations

- **Interlocking scheme:**
  - routes, blocks, switches, signal aspects
Example File

```xml
<?xml version="1.0" encoding="UTF-8"?>
<railml xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance" xsi:noNamespaceSchemaLocation="timetable.xsd">
  <timetable version="1.1">
    <train trainID="RX 100.2" type="planned" source="opentrack">
      <timetableentries>
        <entry posID="ZU" departure="06:08:00" type="begin"/>
        <entry posID="ZWI" departure="06:10:30" type="pass"/>
        <entry posID="ZOER" arrival="06:16:00" departure="06:17:00" minStopTime="9" type="stop"/>
        <entry posID="WS" departure="06:21:00" type="pass"/>
        <entry posID="DUE" departure="06:23:00" type="pass"/>
        <entry posID="SCW" departure="06:27:00" type="pass"/>
        <entry posID="NAE" departure="06:29:00" type="pass"/>
        <entry posID="UST" arrival="06:34:30" type="stop"/>
      </timetableentries>
    </train>
  </timetable>
</railml>
```
## Roadmap

<table>
<thead>
<tr>
<th>Version and subscheme</th>
<th>railML V 0.x</th>
<th>railML V 1.0</th>
<th>railML V 1.1</th>
<th>railML V 2.0</th>
<th>railML V2.2</th>
<th>railML V 3.x</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timetable</td>
<td>First test &amp; use cases</td>
<td>Ready for daily use</td>
<td>Elements added</td>
<td>Total reorganisation</td>
<td>Elements added</td>
<td>No changes</td>
</tr>
<tr>
<td>Rolling stock</td>
<td>First test &amp; use cases</td>
<td>Ready for daily use</td>
<td>Elements added</td>
<td>No changes</td>
<td>No changes</td>
<td>No changes</td>
</tr>
<tr>
<td>Infrastructure macroscopic</td>
<td>Not implemented</td>
<td>First test &amp; use cases</td>
<td>Ready for daily use</td>
<td>No changes</td>
<td>Elements added</td>
<td>Total reorganisation</td>
</tr>
<tr>
<td>Infrastructure microscopic</td>
<td>Not implemented</td>
<td>First test &amp; use cases</td>
<td>No changes</td>
<td>No changes</td>
<td>Total reorganisation</td>
<td></td>
</tr>
<tr>
<td>Infrastructure interlocking</td>
<td>Not implemented</td>
<td>Not implemented</td>
<td>Not implemented</td>
<td>Not implemented</td>
<td>Ready for daily use</td>
<td></td>
</tr>
</tbody>
</table>

---

*railML* is a standard for representing railway data. Each version (V 0.x, V 1.0, V 1.1, V 2.0, V2.2, V 3.x) includes improvements and features compared to its predecessor.
Organization

• Semi-annual conferences to exchange experience and discuss basics
• Project coordinators for the individual subschema moderate and establish releases
• Documentation with a railML-Wiki and HTML explanation files
• Discussions in German and English; Documentation is entirely in English
• Coordination in Dresden & Zurich
Information Sources

• Webpage
  – http://www.railml.org/

• Download xsd schemas and examples

• Online documentation with browsable element hierarchy

• Wiki
  – http://wiki.railml.org

• Forum
  – http://forum.railml.org/
Data Sources

• Examples for all schemas are part of the railML distributions

• Model of the entire Norwegian network, provided by Jernbaneverket
Current Developments

- Preparation of railML 3.0
- Support for UIC Topo-Model
  http://www.railml.org/index.php/railml3-development.html
- New subscheme for Interlocking
- Reorganize organization and structure
- Certification process
railML in OpenTrack

• Daniel Hürlimann is founding member of railML consortium
• OpenTrack was one of the first tools to support railML (since v0.95)
• OpenTrack uses all schemas for import and export
• Automatic imports gets more and more important
railML in OpenTrack

• **New in OpenTrack 1.8: support for railML 2.2**
  
  • Functions -> Exchange Rolling Stock Data -> Export Rolling Stock (railML-Format) – Version 2.2
  • Functions -> Exchange Rolling Stock Data -> Import Rolling Stock (railML-Format) – Version 2.2
  
  • Functions -> Exchange Timetable Data -> Export Timetable (railML-Format) – Version 2.2
  • Functions -> Exchange Timetable Data -> Import Timetable (railML-Format) – Version 2.2
  
  • Functions -> Exchange Infrastr. Data -> Export Infrastructure (railML-Format) – Version 2.2
  • Functions -> Exchange Infrastr. Data -> Import Infrastructure (railML-Format) – Version 2.2
  
  • Functions -> Exchange Station Data -> Export Stations (railML-Format) – Version 2.2
  • Functions -> Exchange Station Data -> Import Stations (railML-Format) – Version 2.2

• **Mapping for timetable and rolling stock straight forward**
• **Infrastructure modeling different in OpenTrack (double vertex graph) and railML (line / track oriented)**
Other railML tools

Some tools supporting railML:

- FBS, iRFP
- IVU.suite, IVU
- LUKS, VIA-Con
- OnTime, trafIT solutions / VIA-Con
- OpenPowerNet, IfB
- OpenTimeTable, distributed by VIA-Con
- Treno, Lif lab
- TPS, HaCon
- Viritio, SMA and Partner
- VISUM, PTV
- ...

- new: Open-Source tool railVIVID funded by UIC

http://www.railml.org/index.php/programme.html
railOScope by trafIT solutions

validate, visualize and analyze rail-data (railML®)

- Master rail data
  - Validate railML files
  - Edit, correct, complete (rule based)
  - Revise, convert, publish

- Access files and understand data
  - Fault-tolerant handling of non-standard files
  - Visualize: standard views

- A reliable product
  - Support, Updates, Upgrades
  - Customization
Summary

- railML is a mature exchange format
- railML is now supported by UIC
- OpenTrack supports v1.x and 2.2 (new)
- Many other tools for railML available

- Join the railML initiative!

- Next meeting (28th conference): 05 Nov 2015, UIC, Paris
- http://www.railml.org/