News of railML-Common parts

24th railML.org® – meeting

Susanne Wunsch

railML.org

Paris, September 18th, 2013

General introduction

Implementation process

Development cycle

Documentation

Coordinators meetings

General concepts

Identities

References

Code lists

Selected topics from sub-schemas

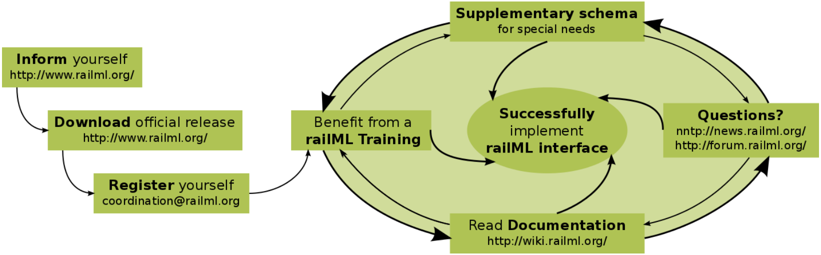
Stop posts, platform edges and service sections

Some `infrastructure` objects

Geographical localizations

MathML integration for `rollingstock` formulas

How to use and implement railML?



Inform yourself
http://www.railml.org/

Download official release
http://www.railml.org/

Register yourself
coordination@railml.org

Benefit from a **railML Training**

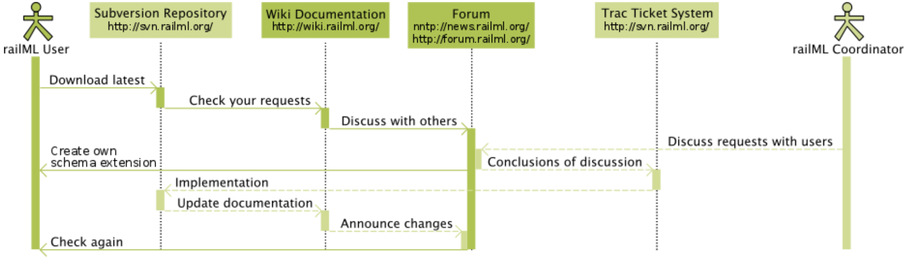
Supplementary schema for special needs

Successfully implement **railML interface**

Questions?
nntp://news.railml.org/
http://forum.railml.org/

Read **Documentation**
http://wiki.railml.org/

How to develop new features in railML?



Documentation in the wiki and on the web site

`http://wiki.railml.org/`

- General wiki pages with FAQ character
- Each XML element is documented on a single wiki page with its XML attributes and some sample code
- Links to the appropriate wiki page are located inside the XML schemas

`http://www.railml.org/index.php/dokumentation.html`

- The latest official railML-schema version is illustrated on HTML web sites, created by automatic XML schema documentation tools

Small coordinators meetings in Dresden

- Feature enhancements for railML release 2.2 clarified
- Transparent results: directly passed to Trac tickets
- Boosted mainly the `infrastructure`-development
- Further discussion and results: see railML forum posts

Official Release of railML 2.2 was published on June 11th, 2013

Identities with xs:ID

- Base type `tGenericID` used for attribute `id` of type `xs:ID`
- XML Validators check for unique attributes of `xs:ID` inside an XML File
- XML Validators check for not more than one attribute of `xs:ID` inside an XML Element
- XML Validators check for lexical constraints (`NCName`): starting with a `Letter` or '_', no whitespaces

<ocp id="ocp80BL"...
<formation id="fCNL"...
<train id="t1242"...

References with xs:IDREF

- Base type `tGenericRef` used for attribute `ref` of type `xs:IDREF`
- XML Validators check for presence of equivalent `xs:ID` inside the XML File
- XML Validators check for lexikal constraints (`NCName`): starting with a `Letter` or '_', no whitespaces

<ocpRef ref=''ocp80BL''/>
<formationRef ref=''fCNL''/>
<trainRef ref=''t1242''/>

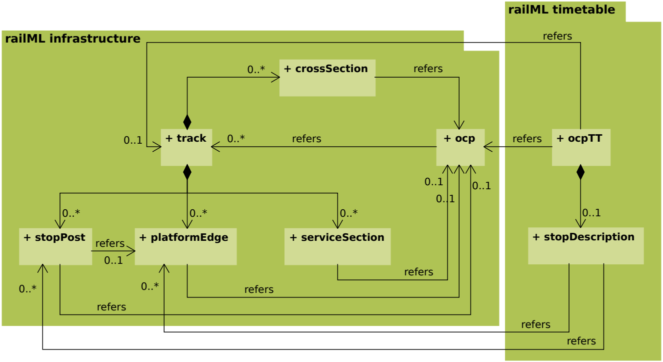
XML code lists instead of XML Schema enumerations

- `TrainProtectionSystems.xml`
  separated for systems “at track” and systems “on vehicle”
- `Registers.xml`
  organization-specific registers for operation or control points (ocps), e.g. `RL100`
- `InfrastructureManagerCodes.xml`
  abbreviation of mostly European infrastructure managers

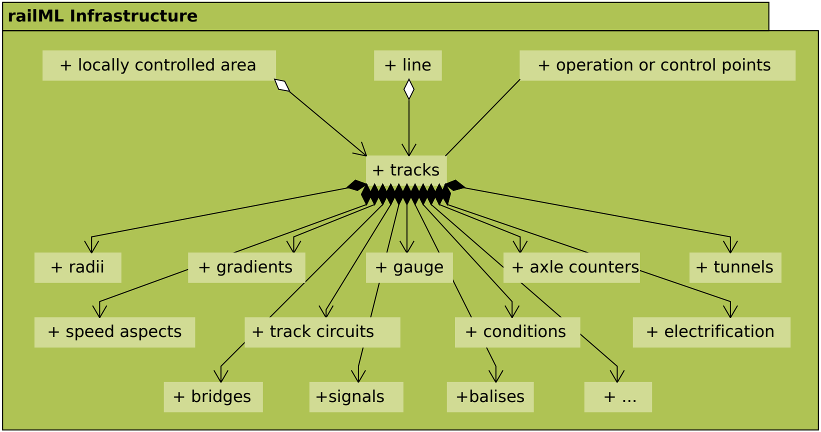
Current strategy for XML code list files

- The lists are provided as a minimum starting point. They should be reviewed and enhanced by the railML-communities experiences.
- Each topic is defined in a separate file.
- Code list files are located in the same folder as the railML-schema files.
- XML schemas for these lists are provided at a separate folder (`codelist-schemas`).

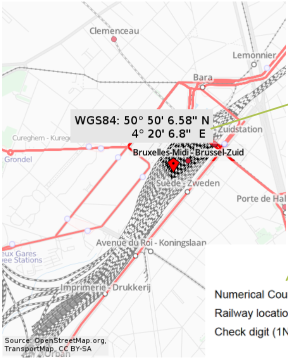
Overview over stop posts, platform edges and service sections in infrastructure and timetable



Some infrastructure objects



Localization of operation or control points



WGS84: 50° 50' 6.58" N
4° 20' 6.8" E

Source: OpenStreetMap.org, TransportMap, CC BY-SA

Linear mileage, incl. overlapping and missing stations

Coordinates with reference system, e.g. WGS84 for GPS

Standardized code, RU- or IM-specific, country-specific or global, e.g. TSI-TAP

NN NNNNN N

Numerical Country code (2N)
Railway location number (5N)
Check digit (1N)

Source: TAP TSI, Annex B.9

Sample: Train resistance

$$17,456N + 5,677N\frac{s}{m} * v + 1,234N(\frac{s}{m})^2 * v^2$$

- MathML provides both content and presentation mode.
- MathML presentation mode is widely available in web browsers and document generation (publishing domain).
- MathML content mode seems to be less widespread than the presentation mode.
- For integration of MathML formulas into railML elements, the content mode should be preferred.

Any experiences are warmly welcomed.

Thank you for your attention.