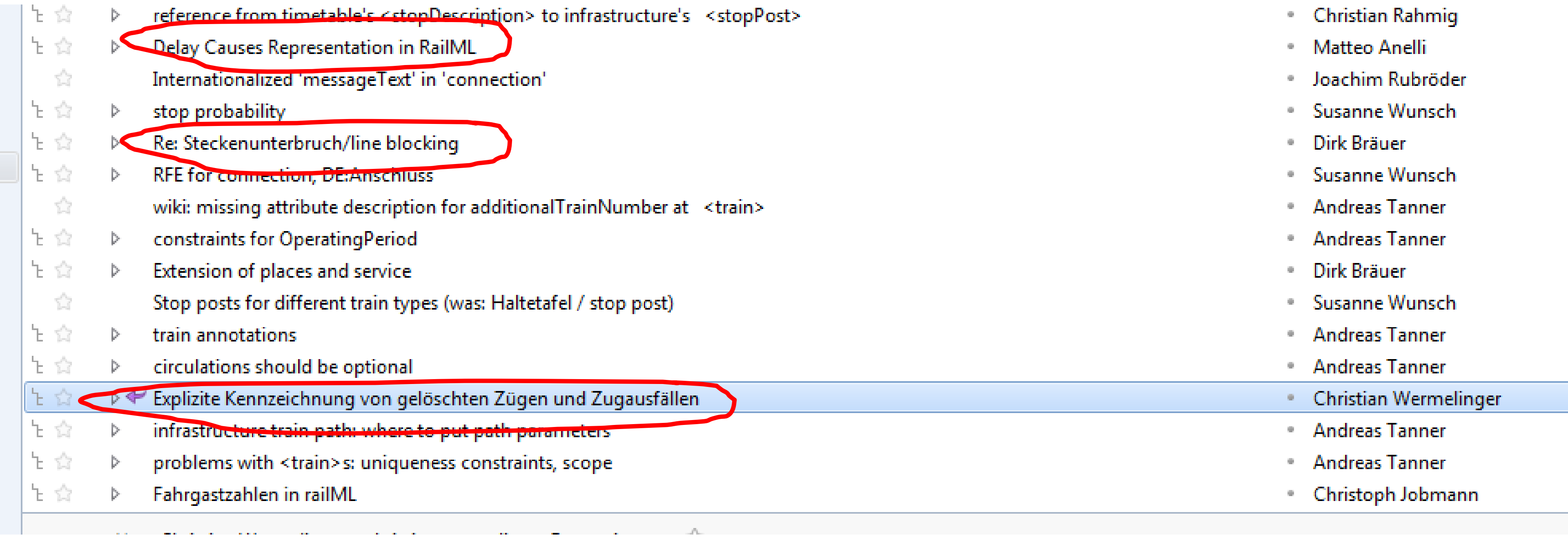
Functional interface for railML differential data exchange

Dr. Andreas Tanner
IVU AG
2013-09-17

Motivation

- Synchronisation of timetables and vehicle working schedules modeled in railML
- Use cases:
  - Import: TrenItalia, MAV – Differential import for timetables (planning)
  - Export: VR (Finland), Export for dispositional schedule changes
  - All use cases have strong non-functional (performance) requirements
- railML 2.2 -Standard has no support for „data telegrams“
- Aim: Consolidated solution, symmetry for import and export

Motivation: From the railML-Forum



reference from timetable's <stopDescription> to infrastructure's <stopPost>	Christian Rahmig
Delay Causes Representation in RailML	Matteo Anelli
Internationalized 'messageText' in 'connection'	Joachim Rubröder
stop probability	Susanne Wunsch
Re: Steckenunterbruch/line blocking	Dirk Bräuer
RFE for connection, DE:Anschluss	Susanne Wunsch
wiki: missing attribute description for additionalTrainNumber at <train>	Andreas Tanner
constraints for OperatingPeriod	Andreas Tanner
Extension of places and service	Dirk Bräuer
Stop posts for different train types (was: Haltetafel / stop post)	Susanne Wunsch
train annotations	Andreas Tanner
circulations should be optional	Andreas Tanner
Explizite Kennzeichnung von gelöschten Zügen und Zugausfällen	Christian Wermelinger
infrastructure train path: where to put path parameters	Andreas Tanner
problems with <train>s: uniqueness constraints, scope	Andreas Tanner
Fahrgastzahlen in railML	Christoph Jobmann

Solution concept

- Functional interface, SOAP based
- Requirement: Functional or technical keys for railML entities
  - <train>
  - <ocpTT> within <train>
  - <formation>
  - …
- TODO: define key for train
- Key for <ocptt> index

Solution concept

1. Full Resynchronisation of a train
   - <synchronizeTrain>
     - Complete transmission of state of a train identified by functional key

Solution concept

2. Modification telegrams
   - <modifyTrain>
     - trainKey (trainNumber, additionalTrainNumber, operatingPeriod)
     - Attribut „state“ with values originalData/changed/new/deleted
     - Simple attributes (if changed)
     - List of <modifyOcptt> for changes of itinerary and track occupancy

Solution concept

2. Example for telegram

```
<modifyTrain>
<ocps>
<ocp id=„ocp1“ code=„H“/>
<ocp id=„… />
</ocps>
<formations>
<formation id=… …/>
</formations>
<operatingPeriods>
<operatingPeriod id=.. startDate= … endDate= … />
</operatingPeriods>

<trainKey trainNumber=„1“ additionalTrainNumber=„11“ operatingPeriodRef=…>
<formationTT state=„changed“ formationRef=„…“ />
<modifyOcpTT index = 13>
<times state=„changed“ >…
```

Key

Example for changed stop

```
…
<trainKey trainNumber=„1“ additionalTrainNumber=„11“ operatingPeriodRef=…>
<formationTT state=„new“ formationRef=„…“ />
<modifyOcpTT index=13>
<ocpTT state=„changed“ ocpType=„pass“ …><times>…</ocpTT>
<ocpTT state=„originalData“>…
<modifyOcpTT/>
```

Example for deleted stop

```
…
<trainKey trainNumber=„1“ additionalTrainNumber=„11“ operatingPeriodRef=…>
<formationTT state=„new“ formationRef=„…“ />
<modifyOcpTT index=14>
<ocpTT state=„deleted“/>…
```

Example for new stop

```
…
<trainKey trainNumber=„1“ additionalTrainNumber=„11“ operatingPeriodRef=…>
<formationTT state=„new“ formationRef=„…“ />
<modifyOcpTT index=10.5>
<ocpTT state=„new“>…
…
<modifyOcpTT index=-1>
<ocpTT state=„new“>…
```

New stop between 10th and 11th stop

New Stop at beginning

Use cases to cover

- Partial cancellation of a train
- Data transfer
- Activating contingency plan
- Track assignment
- Track change
- Track un-assignment
- Stop pattern change
- Increase vehicle number(s)
- Decrease vehicle number(s)
- Trip class change
- Departure time change
- Arrival time change
- Change of planned vehicle type
- Commercial line change
- Assigning vehicle
- Vehicle un-assignment
- Change of vehicle assignment
- Partial extension of a train path
- Change of vehicle formation
- Position change of vehicle inside one track section

Use cases to cover

modifyTrain

- Partial cancellation of a train
- Data transfer
- Activating contingency plan
- Track assignment
- Track change
- Track un-assignment
- Stop pattern change
- Increase vehicle number(s)
- Decrease vehicle number(s)
- Trip class change
- Departure time change
- Arrival time change
- Change of planned vehicle type
- Commercial line change
- Assigning vehicle
- Vehicle un-assignment
- Change of vehicle assignment
- Partial extension of a train path
- Change of vehicle formation
- Position change of vehicle inside one track section

Yes, if formation change. No, if changes on coupling level

Use cases to cover

synchronizeTrain

- Partial cancellation of a train
- Data transfer
- Activating contingency plan
- Track assignment
- Track change
- Track un-assignment
- Stop pattern change
- Increase vehicle number(s)
- Decrease vehicle number(s)
- Trip class change
- Departure time change
- Arrival time change
- Change of planned vehicle type
- Commercial line change
- Assigning vehicle
- Vehicle un-assignment
- Change of vehicle assignment
- Partial extension of a train path
- Change of vehicle formation
- Position change of vehicle inside one track section

Summary

- Extending standard to support data changes would be helpful.
- Data telegrams could cover popular use cases.
- IVU will implement (proprietary, for now) solution.
- For standardisation, collaboration is required.

Thanks for your attention.

Andreas Tanner
ata@ivu.de