From more than 20 years of railML development

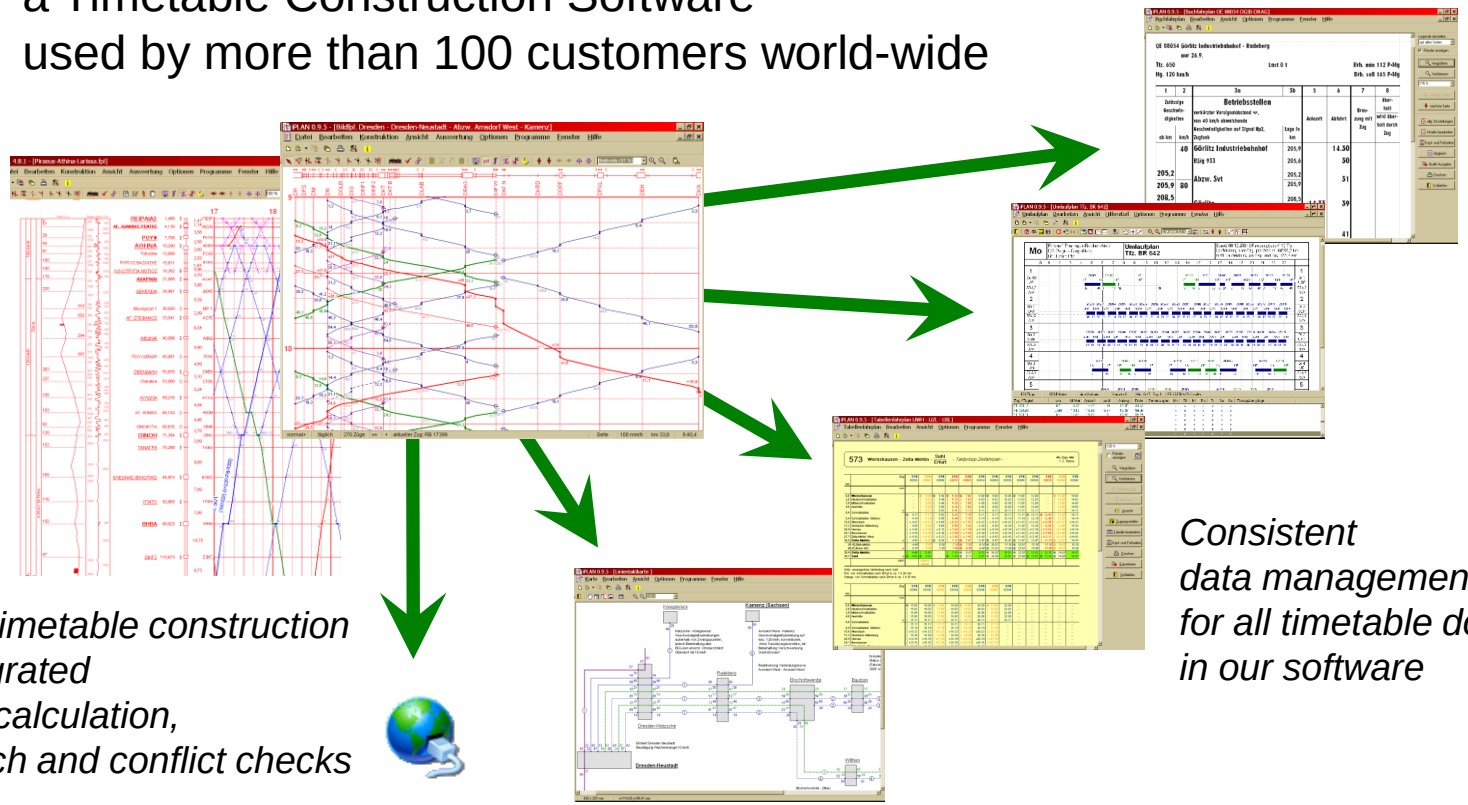
– *Lessons learnd by iRFP* –

railML.org

Dirk Bräuer
iRFP Dresden
dirk.braeuer@irfp.de
Tel. 0351 4706819

Who is iRFP, what do we represent?

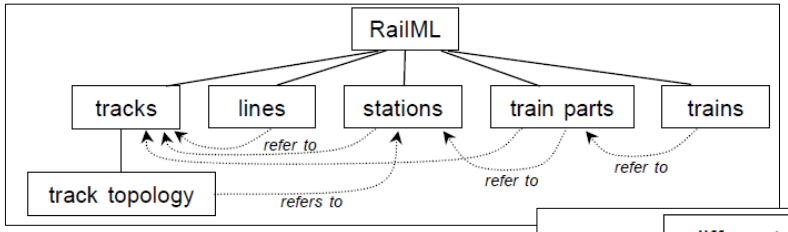
- iRFP is an Institute emerged from Dresden University of Technology, Faculty of Traffic Sciences *Friedrich List*
- iRFP is a foundation member of railML.org from the first minute
- We develop and maintain *Fahrplanbearbeitungssystem FBS*, a Timetable Construction Software used by more than 100 customers world-wide

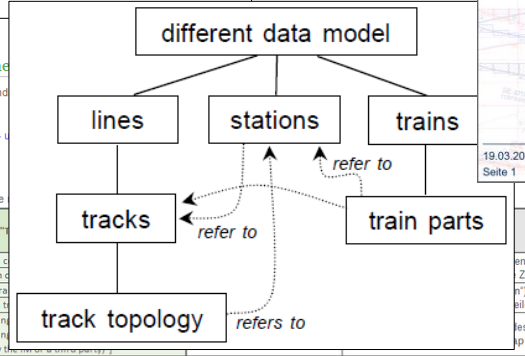


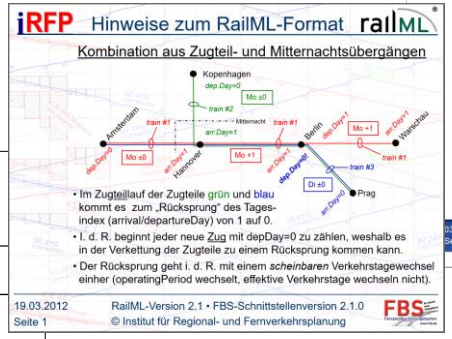
Graphic timetable construction with integrated run-time calculation, slot search and conflict checks

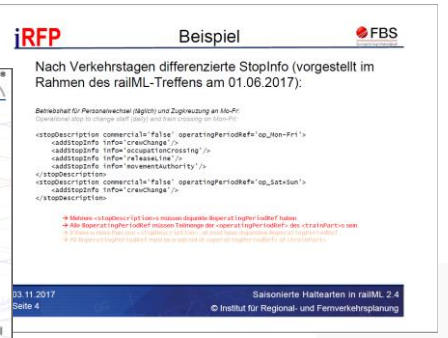
Consistent data management for all timetable documents in our software

(Extract from) Contribution of iRFP at railML.org









Best practice & Examples / Empfohlene Anwendung & Beispiele / Bonne

The following list shows, how the values of @type correspond with different standards. The considered stand

- **TAF/TAP TSI:** TrainActivityType (always Englisch)
- **ÖBB:** Auftragsmanagement der ÖBB Infrastruktur AG (Halteinfos sind teilweise durchgereicht für Kunden-
- **SBB:** Netzweites Trassen-System (NeTS) der SBB (always German)
- **DB:** Auftragsmanagement der DB Netz AG (V2, V3) (always German)

railML® maps values common in two or more of these standards. Other, more special/national values should be

<stopActivity>.type	English explanation		TAF/TAP TSI "T			Auftragsmanagement der ÖBB Infrastruktur AG (Halteinfos sind teilweise durchgereicht für Kunden- und Folgesysteme)	SBB (NeTS)	Auftragsmanager (V2, V3)
collect	Couple or uncouple vehicles / train parts		0013: attach c			Beigabe von Wagen oder Wagengruppen, Zugbildung	Nur Zufuhr, Nur Zufuhr bei Bedarf*S	Wagen zusetzen
drop	- intended for non-self-propelling train parts	Please consider relation to formations (as far as used)	0014: detach c			Abstellen von Wagen oder Wagengruppen, Zugauflösen	Nur Abfuhr, Nur Abfuhr bei Bedarf*S	Wagen absetzen
split	Couple or uncouple vehicles / train parts		0017: split tra			Zugteilung (besetzte Ausfahrt)	Schwächen/Trennen	
join	- intended for self-propelling train parts		0016: attach t			Zugvereinigung (besetzte Einfahrt)	Stärken/Vereinigen	
shunting	Shunting movements are planned during stop (may also be used additionally to collect/drop/split/join/staple on demand)		0020: shunting 0021: shunting (if offered by the IM or a third party)					
crewChange	Stop to change engine or train crew		0023: Loco driver change 0025: Crew change	Personalwechsel	Halt zum Wechsel von Tfz.- und/oder Zugpers			
crewBreak	Stop for engine or train crew to pause		0024: Loco driver break	Personalpause	Halt zum Einhalten der Pausenzeiten von Tfz.			
reverse	Stop to change driving direction of train							
runAround	Shunting of the engine to the other end of train (normally during change of driving direction)							
engineAttach	Receiving of pilot/pushing/double-heading eng							
engineDetach	Dropping of pilot/pushing/double-heading eng							
engineChange	Change of engine (no train coupling and sharin							
staple	Notes the necessity of staple track(s) during th							
occupation	Stop caused by occupation of lines, tracks, blo specification							
occupationBlock	Stop caused by occupation of route by another							
occupationCrossing	Stop caused by occupation of route by an oncc							
occupationStation	Stop caused by occupation of station track(s)							
powerSystemChange	Stop caused by a change of power supply (e.g.							
vehicleInspection	Stop for technical / technological vehicle inspe							
gaugeChange	Stop caused by a change of gauge (e.g. to chan							
authorityCheck	Stop for purposes of passport or customs chec							
photo								
catering	Supply or disposal of passengers or payload,							
supplyOrDisposal	Supply of vehicles with operating supplies or r including taking water for steam traction or st							
movementAuthority	Stop to receive the permission to proceed into *intented to be used with simplified train operati*							
releaseLine	Stop to release the permission to proceed fron *intented to be used with simplified train operati*							
shuntingPermission	Receive and/or release the permission to shun *intented to be used with simplified train operati*							
misc	Specification possible by using @purpose							
other:...	individual extension of the enumeration				individuelle Erweiterung der Aufzählung			

Zusammenhang zur konkreten Formation beachten (sofern verwendet)

Different stop types

General traffic stop

```
<ocpTT ocpRef='ocp_DRAG' ocpType='stop'>
  <times scope='scheduled' arrival='12:30:32' departure='12:31:02'/>
  <stopDescription commercial='true' stopOnRequest='false'>
    <stopTimes minimalTime='PT30S'/>
  </stopDescription>
</ocpTT>
```

Operational stop infrastructure

```
<ocpTT ocpRef='ocp_DKT' ocpType='stop'>
  <times scope='scheduled' arrival='12:51:25' departure='12:51:55'/>
  <stopDescription commercial='false' operationalStopOrdered='false'>
    <stopTimes minimalTime='PT30S'/>
  </stopDescription>
</ocpTT>
```

Operational stop by TOC

```
<ocpTT ocpRef='ocp_DKT' trackInfo='3' ocpType='stop'>
  <times scope='scheduled' arrival='12:49:52' departure='12:50:18'/>
  <stopDescription commercial='false' operationalStopOrdered='true'>
    <stopTimes minimalTime='PT30S'/>
  </stopDescription>
</ocpTT>
```

Stop on request / on demand

```
<ocpTT ocpRef='ocp_DIG' ocpType='stop'>
  <times scope='scheduled' arrival='13:33:26' departure='13:33:56'/>
  <stopDescription commercial='true' stopOnRequest='true'>
    <stopTimes minimalTime='PT12S'/>
  </stopDescription>
</ocpTT>
```

Stop to alight only

```
<ocpTT ocpRef='ocp_DN' ocpType='stop'>
  <times scope='scheduled' arrival='12:58:23' departure='13:00:23'/>
  <stopDescription commercial='true' stopOnRequest='false' onOff='off'>
    <stopTimes minimalTime='PT2M0S'/>
  </stopDescription>
</ocpTT>
```

Semantic Constraints / Semantische Beschränkungen / Contraintes semantiques

Semantic Constraint "TT006":

The following table summarises the semantical constraints between the attributes <ocpTT>.ocpType, <stopDescription>.guaranteedPass, .commercial, .onOff, .stopOnRequest and .operationalStopOrdered

Nr.	<ocpTT>.ocpType	<stopDescription>.guaranteedPass	.commercial	.onOff	.stopOnRequest	.operationalStopordered	Description
1.1	pass	true	attribute not to be used	attribute not to be used	attribute not to be used	attribute not to be used	guaranteed pass
1.2		false	attribute not to be used	attribute not to be used	attribute not to be used	attribute not to be used	non-guaranteed pass
2.1	stop, begin, end	attribute not to be used	true	both	true	attribute not to be used	commercial stop on request for on and off
2.2		attribute not to be used	true	both	false	attribute not to be used	commercial stop for on and off
2.3		attribute not to be used	true	on	true	attribute not to be used	commercial stop on request for on only
2.4		attribute not to be used	true	on	false	attribute not to be used	commercial stop for on only
2.5		attribute not to be used	true	off	true	attribute not to be used	commercial stop on request for off only
2.6		attribute not to be used	true	off	false	attribute not to be used	commercial stop for off only
2.7		attribute not to be used	false	currently not supported	currently not supported	true	operational stop ordered by the TOC
2.8		attribute not to be used	false	currently not supported	currently not supported	false	operational stop introduced by the IM

- Green cells are default values.
- ocpType='begin','end' are deprecated from railML 2.2.
- If no <stopDescription> is given, it is either a non-guaranteed pass (1.2) or a stop with undefined properties, depending on the attribute ocpType.
- The term "commercial" of the attribute in railML traditionally refers to the contractual relationship between RU and end-customer, not to the contractual relationship between IM and RU.
- The term "ordered" in the attribute operationalStopOrdered refers to the contractual relationship between IM and RU.

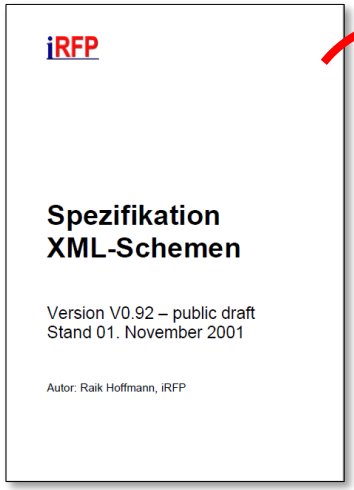
Proposed on September 3rd 2018
Approved on June 20th 2019
FIXME add Link to discussion!
Please, recognize our guidelines on semantic constraints

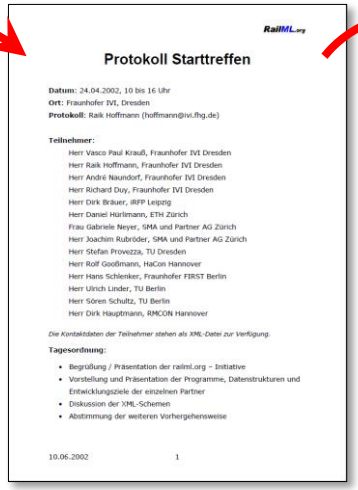
...cher Wagenuntersuchungen (vgl. jedoch authorityCheck)	Technische Wagenuntersuchung Wagentechnische Behandlung	---	
...els der Spurweite	---	---	---
...asskontrolle o. a. Behandlung durch Behörden	Grenzaufenthalt, Grenzkontrolle durch Grenzpolizei, Verzollen	---	---
	Sonderhalt (z.B.Fotohalt)	---	Fotohalt
...ng von Reisenden oder Ladung, z. B. Zuladung Catering	Verpflegungshalt (z.B. Beigabe von frischem Gebäck)	---	Halt für Bewirtsc
...ng von Fahrzeugen mit Betriebsstoffen oder Müll (auch: Wasser-Dampfheizung)	---	---	---
...len der Erlaubnis, den nächsten Streckenabschnitt zu befahren *...ne von Zugleitbetrieb u. ä. vorgesehen*	Fahranfrage	---	RI1436: (Anfrage: RV-NE: Fahranfra
...ben der Erlaubnis, den zurückliegenden Streckenabschnitt zu *...zu Vorherigem; i. S. von Zugleitbetrieb u. ä. vorgesehen*	Ankunftsmeldung	---	RI1436: Ankunftsm RV-NE: Ankunftsm
...eben der Erlaubnis zum Rangieren während des Haltes innerhalb der *...im Sinne von Zugleitbetrieb u. ä. vorgesehen*	Verschubanfrage	---	RI1436: Rangieren RV-NE: ---
...der @purpose möglich	Sonstiges	---	sonstiger Haltegr werden)

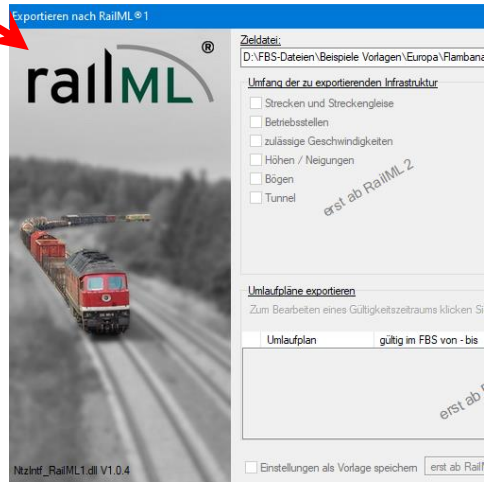
The beginnings of *rail*ML…

The beginnings of *rail*ML more than 20 years ago:

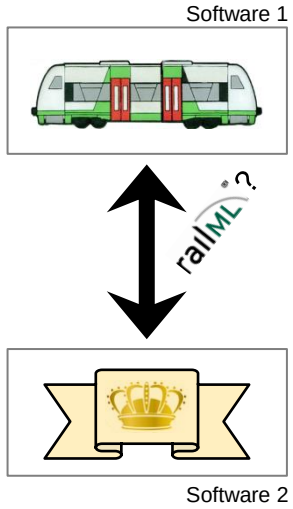
- Starting point was a raising number of „proprietary“ interfaces with the same claim:
  - VDV452 & Co. („derivatives“ derived from it)
  - FBS-XML (own development)
  - Exports based on Microsoft Excel and Text files
- This led to the idea of „one standard interface for all program combinations“
- The creation of a non-committed, neutral, open-source format leads to more acceptance than the selection of one proprietary format







*rail*ML® theses of the first hour



- *rail*ML is a data format purely designed for **data exchange** between two software programs – not for data storage

  → No persistent primary keys etc.

- There need to be at least **two independent partners** to place a demand for requirements

  → Not one side alone can demand something

  → Data existing purely in one software are placed back

- **Each requirement** of two independent partners shall be represented somehow

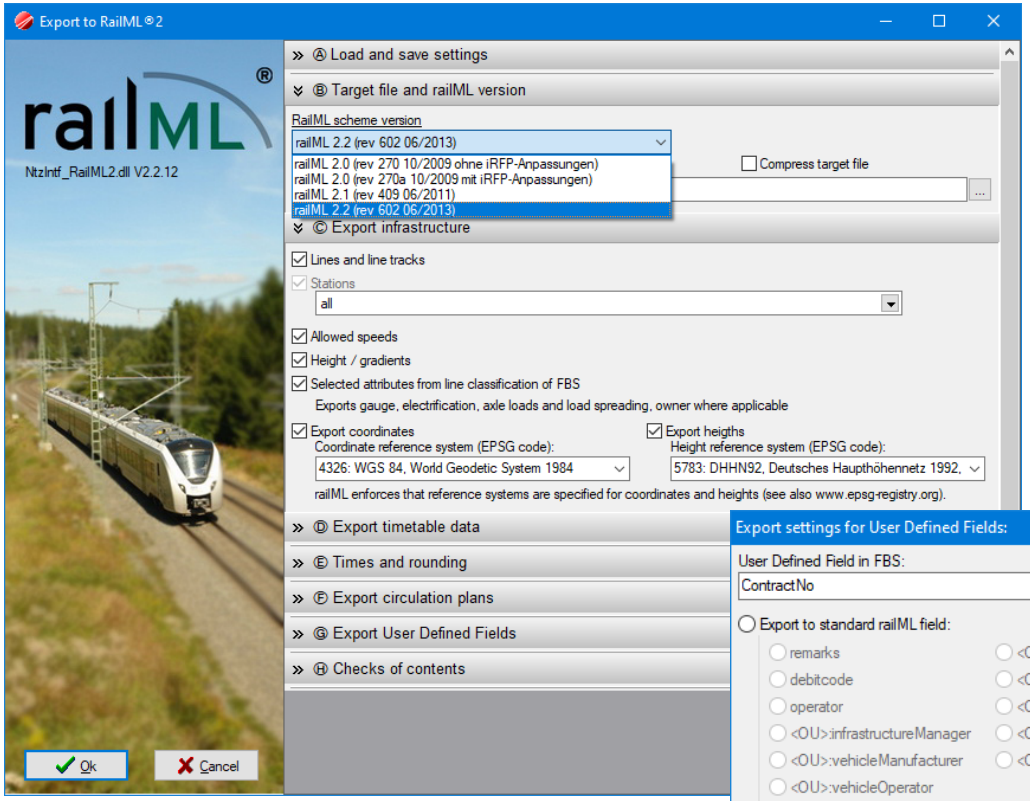
  (a practical existing requirement cannot be refused)

- **Avoidance of redundancies**: What already can be expressed in one kind shall not be expressible in any other kind.

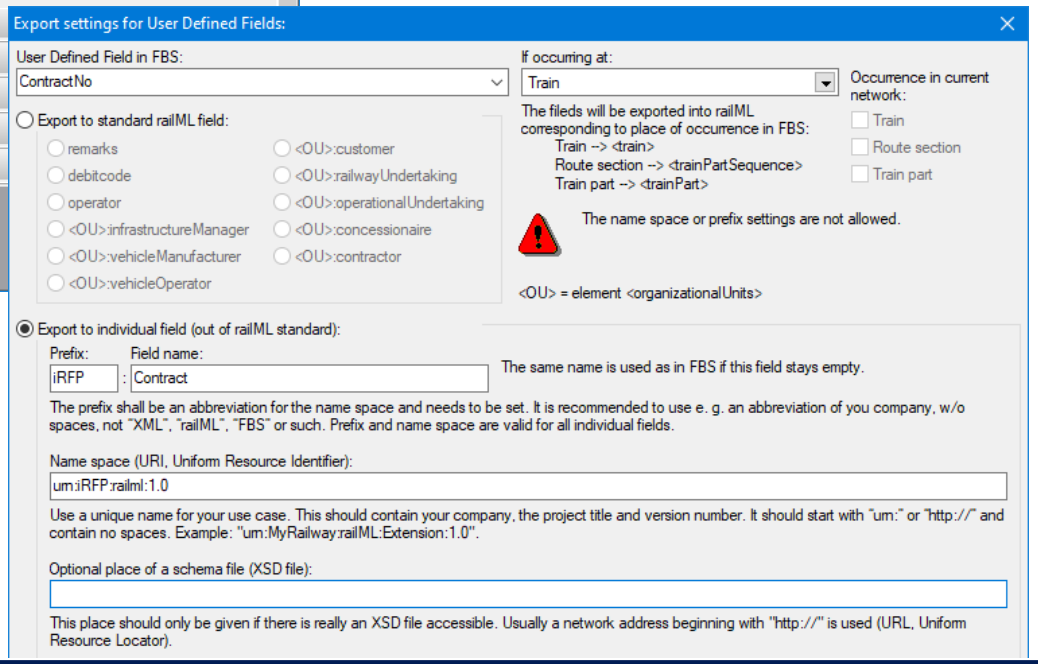
  Where there are unavoidable redundancies, there shall be a preference of one solution at least by a semantical constraint.

  → Ease of import from *rail*ML files

FBS-RailML Interface in particular: Export



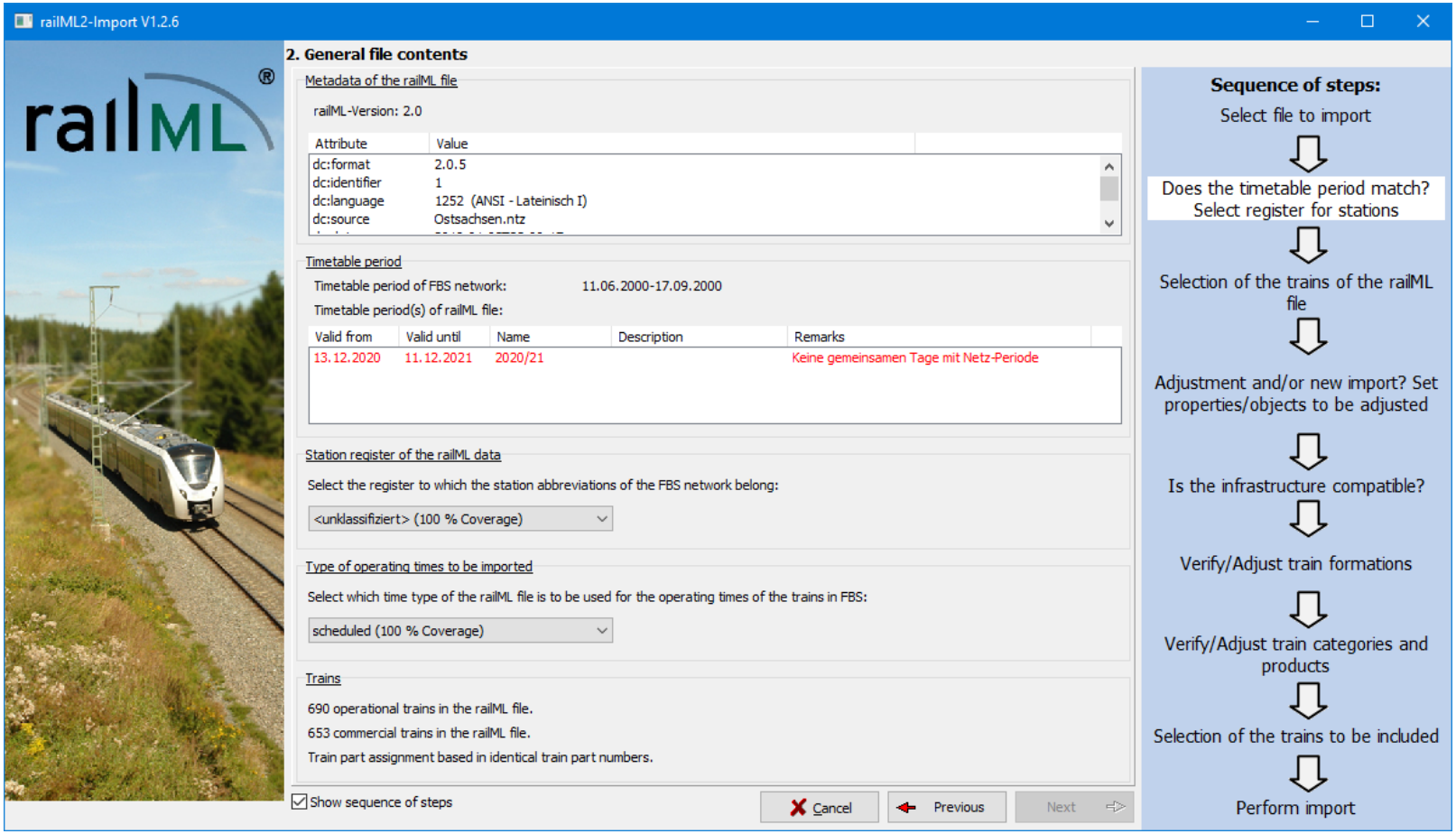
- *rail*ML-Versions 2.x
- Infrastructure
- Vehicles and formations
- Timetable:
  - Calendar data
  - Trains, train parts and their categories, operational & commercial



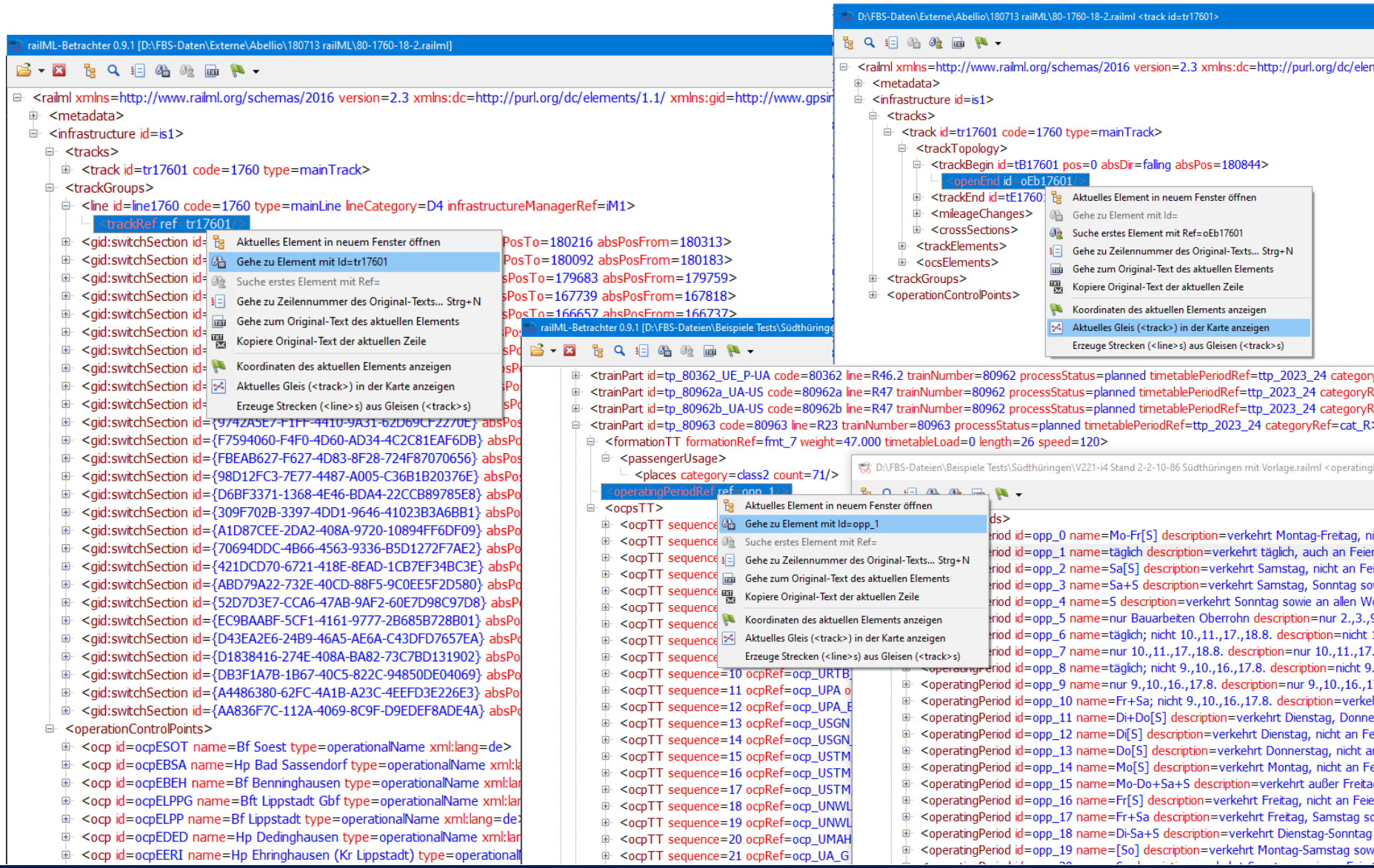
- Optional vehicle circulation, extendable for “folded” circulations
- Optional user defined fields

FBS-RailML Interface in particular: Import

- *rail*ML-Versions 2.x
- Timetable data only
- Integrated “route search” through the network
- Optional substitution of vehicles, formations, categories etc.



iRFP-own “support tools”, e. g. *rail*ML viewer



Practical use cases at iRFP 1/3: Export

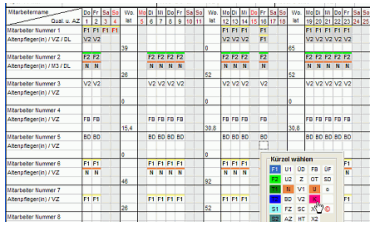
a) Passenger information: Transfer of (planned) timetable data incl. circulation
   - into the trains
   - to the platforms
   - into (online) media

b) Disposition / Control centre / RBL (“computer-aided operations management”): Transfer of (planned) timetable data incl. circulation
Includes also, for instance, “SAT-ZB” of FH Wels

c) Staff planning/rostering systems, with or without vehicle circulation data

d) Booking / reservation systems: Transfer of (planned) time-table data incl. train formation and carriage booking no’s.

e) Prognosis systems for passenger demand / seat capacity utilisation: Transfer of (planned) timetable data
Includes e. g. *PTV-VISION* software family



f) Driver assisting systems: Transfer of infrastructure data incl. geographic coordinates and planned timetable data

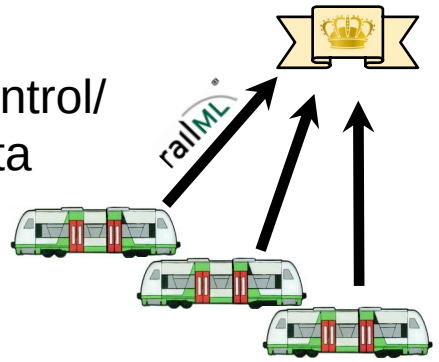
g) Special, rather individual solutions
such as an automatic export of ad-hoc planned trains to an invoicing system

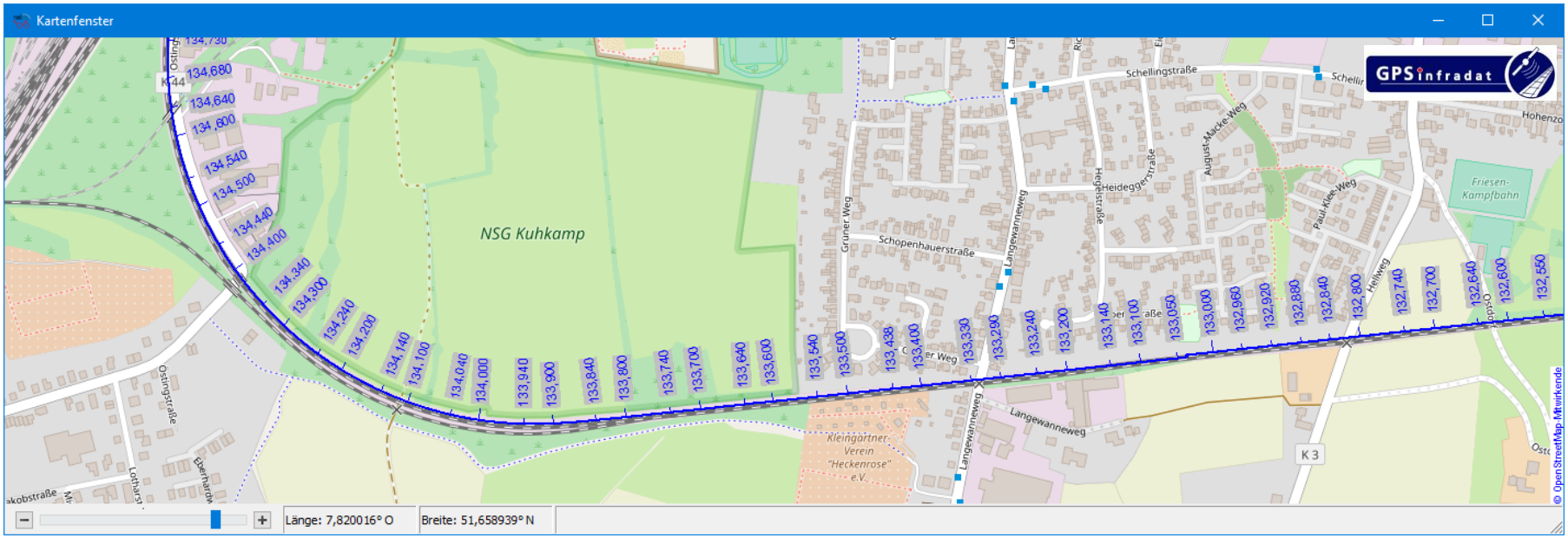
Practical use cases at iRFP 2/3: Import

a) Collecting of next-years' timetable offers at Public Transport Authorities
here also: FBS-to-FBS data exchange via *rail*ML for "neutralisation" only
(a controverse use case, admittedly)

b) Import of real-time train data from CTC (Centralised Train Control/ Signalling) centres for comparison with planned timetable data

c) Import of geographic coordinates
(so far, a special solution for internal purposes only)





Practical use cases 3/3: Discarded and deferred

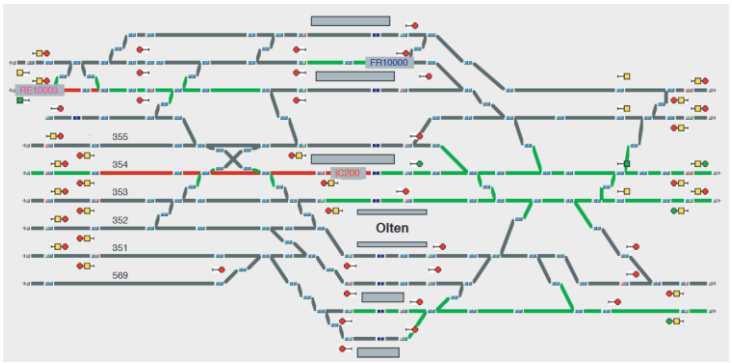
a) Data exchange of rolling stock data from Rolling Stock (data input)
Internet form – too less demand, too “enclosed” data…

b) Export of planned timetable data to synchronous simulations
(such as *OpenTrack*) – very limited applications, only a few replies

c) Export of (electronic) Driver’s timetables, “EBuLa”
– PDF is easier…

d) Import of infrastructure data
– “stand-alone” / “one-hit” solutions only, many different “realisations” in *rail*ML, much effort for a limited demand esp. in competition to *OpenStreetMap* data

Beispiel für einen elektronischen Buchfahrplan aus FBS; Stern & Hafferl, Gmunden



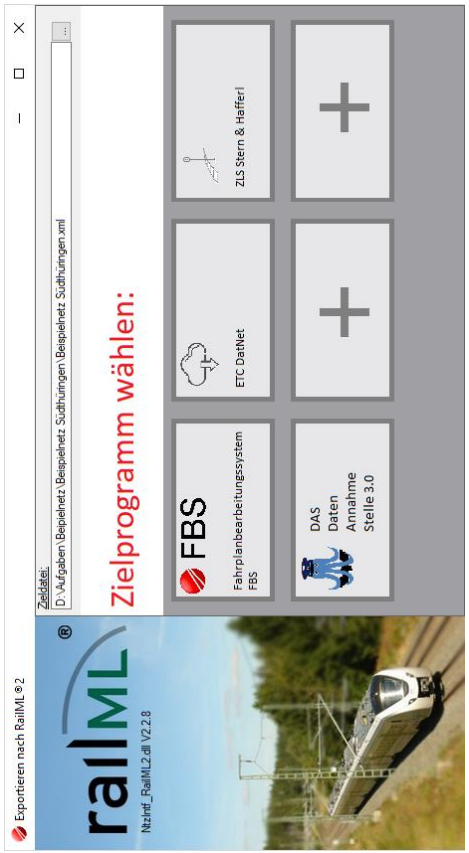
OpenTrack als Beispiel für Synchronsimulation, Quelle: Dr. Hürlimann, Zürich

A selection of users of FBS-RailML Interface

Where are we now?

A special situation for iRFP is:

We do not "oversee" what happens with "our" data!



- The FBS-RailML-Interface is developed (and merchandised) as a *general* solution – according to the original idea
- Each customer can export data to whichever purpose – with no need to consult iRFP
- iRFP only gets note of a *rail*ML "use case" if something does *not* work – which (presumably, fortunately) does not happen very often…

→ The task of "collecting" and "afterwards-systematising" of our use cases is an ongoing challenge.

Did we finally reduce the number of interfaces?

→ Just like Radio Yerevan: In principle, yes. But…

- "Standards in the Standard" are a raising problem
- Programming effort vs. Configuring effort

And finally – wishes to railML.org?

Can *rail*ML be less “unwieldy”?

- Practical demand with nowadays typical time limits only would allow short-term extensions of *rail*ML.
- If once there is a solution with an extension, there will very often never be (time & money for) a “proper” solution…

We see a raising respect for *rail*ML as an accepted industrial standard, but also a raising demand back to proprietary solutions due to raising acceptance problems and/or access barriers (or fear for…)

- because of difficult inner data structures,
- because of easier development (less bureaucracy, no certification)

***rail*ML is – from our view – a good general starting basis for a data exchange interface, but there is a huge amount of additional effort necessary to bring it into a certain, working, well-documented use case.**