

Technische Universität Dresden  
Fakultät Verkehrswissenschaften „Friedrich List“  
Institut für Verkehrstelematik  
Professur für Verkehrsleitsysteme und –prozessautomatisierung

Studienarbeit

**Vergleichende Analyse von NeTEx und railML zur  
Bereitstellung von intermodalen  
Fahrgastinformationen**

eingereicht von Jessica Pöpping

geb. am 25.10.1990 in Bautzen

verantwortlicher Hochschullehrer: Prof. Dr.-Ing. Jürgen Krimmling  
Betreuer: Dipl.-Ing. Astrid Schindelhauer  
Dipl.-Ing. Susanne Wunsch

Dresden, 03.03.2015



**T h e m e n b l a t t  
zur Studien-Arbeit\*)**

von Frau/Herrn cand. ing. **Jessica Pöpping**

Thema (Aufgabenstellung siehe Anlage):

**Vergleichende Analyse von railML und NeTEx zur  
Bereitstellung von intermodalen Fahrgastinformationen**

Institut für Verkehrstelematik

1. Prüfer: Prof. Dr.-Ing. J. Krimmling
2. Prüfer / Beisitzer: Dr.-Ing. M. Lehnert

Zur Anfertigung der Studien-Arbeit wurde eine dreiseitige Vereinbarung (TUD, Studierender, Dritter) abgeschlossen: Ja / Nein (zutreffendes unterstreichen, nicht zutreffendes streichen)

Unterschrift des Prüfers

Dresden, den 05.11.2014

Ausgabetag: 03.12.2014

Abgabetermin: 03.03.2015

Bestätigung durch die Fakultät: Wolfske

Abgabetag: .....

Bestätigung durch die Fakultät: .....

Bestätigung durch die Fakultät für eine genehmigte Verlängerung der Bearbeitungszeit: .....

Hiermit bestätige ich den Empfang der Aufgabenstellung für meine Studien-Arbeit und erkenne die Festlegungen der Richtlinie für die Anfertigung der Studien-Arbeit – insbesondere den Punkt 11 - an:

Unterschrift des Studierenden

Dresden, den 03.12.14

\*) siehe Diplomprüfungsordnung § 7 und Modulbeschreibung Modul 203 sowie Studiendokumente 4., Punkt 2 der Regelung für die Ausgabe und Registratur der Studien-Arbeiten und Diplom-Arbeiten

## Aufgabenstellung für Studienarbeit

für Frau Jessica Pöpping

### **Vergleichende Analyse von railML und NeTEx zur Bereitstellung von intermodalen Fahrgastinformationen**

NeTEx „Network and Timetable Exchange“ (CEN/TS 16614) ist der europäische Standard für die Übertragung von Soll-Daten im Öffentlichen Verkehr bezüglich Netzwerk und Fahrplan. NeTEx wurde basierend auf TransModel „European Reference Data Model for Public Transport“ (EN 12896) entwickelt. Für die Netzbeschreibungen wurden Konzepte aus IFOPT „Identification of Fixed Objects in Public Transport“ (CEN/TS 28701) integriert. In die aktuelle Entwicklung von NeTEx sind vorrangig Erfahrungen aus Nahverkehrssystemen eingeflossen.

RailML ist ein internationaler Industriestandard für den Austausch von Soll-Daten im Schienenverkehr bezüglich Infrastruktur, Rollmaterial und Fahrplan. Die aktuelle Version railML 2.2 enthält weitreichende Konzepte zur eindeutigen Beschreibung von Schienenverkehrsleistungen in unterschiedlichen Granularitäten.

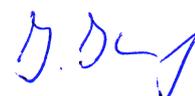
Als Basis für intermodale Fahrgastinformationen werden Soll-Daten in einem möglichst einheitlichen Datenformat benötigt, um bei unvorhergesehenen Ereignissen alternative Empfehlungen anbieten zu können. Beide Standards, railML und NeTEx, sind anhand von intermodalen Beispieldatensätzen in der Region München-Nürnberg zu vergleichen. Der Fokus dieser Arbeit liegt hierbei auf den folgenden Themen:

- Fernverkehr (ICE, Fernbus) inkl. betrieblich gekuppelter Fernverkehr bzw. unterschiedlicher Haltestellen bei Fernbussen
- Nahverkehr (S-/U-Bahn, RB/RE)
- verschiedene bzw. teilweise gemeinsame Fahrwege der schienengebundenen Transportmittel
- Umsteigewege in großen Bahnhöfen
- Fahrpreise
- Flug inkl. Transferzeiten (Check-In)

Die dabei festzustellenden Gemeinsamkeiten und Unterschiede sind geeignet zu dokumentieren. Für etwaige Defizite der Standards sind geeignete Erweiterungen vorzuschlagen. Für NeTEx sind insbesondere die Erkenntnisse des Fernverkehrs von Interesse, für die Konzeption von railML 3.0 die intermodalen Zugänglichkeiten.

Die Ergebnisse der Arbeit sind in einem schriftlichen Bericht zu dokumentieren und in einem Vortrag zu präsentieren. Alle erarbeiteten Materialien (Bericht, Anlagen, Präsentation usw.) sind in elektronischer Form auf einem der Arbeit beigefügten Datenträger abzugeben. Dabei sind jeweils die editierbaren Dateiformate zu verwenden, die der TU-Betreuer vorgibt.

**Betreuer:** Dipl.-Ing. Susanne Wunsch (TU Dresden)  
Dipl.-Ing Astrid Schindelhauer (TU Dresden)



**Prof. Dr.-Ing. J. Krimmling**  
Verantwortlicher Hochschullehrer

# **Bibliografischer Nachweis**

Pöpping, Jessica

## **Vergleichende Analyse von NeTEx und railML zur Bereitstellung von intermodalen Fahrgastinformationen**

Studienarbeit

*Technische Universität Dresden, Fakultät Verkehrswissenschaften „Friedrich List“,  
Institut für Verkehrstelematik*

Studiengang Verkehrsingenieurwesen

50 Seiten, 26 Abbildungen, 1 Tabelle, 19 Quellenangaben

## **Autorenreferat**

Diese Arbeit beschreibt die Analyse und den Vergleich der zwei Datenaustauschformate für den öffentlichen Verkehr *railML*® und *NeTEx*. Es soll dabei vor allem ihre Eignung untersucht werden, intermodale Fahrgastinformationen bereitzustellen.

Datenaustauschformate werden im öffentlichen Verkehr benötigt, um Daten zwischen verschiedenen Akteuren auszutauschen und somit letztendlich auch eine verbesserte Kommunikation dem Fahrgast gegenüber zu erreichen.

Des Weiteren soll eine mögliche Erweiterung der beiden Austauschformate miteinander untersucht werden.

## Thesen

1. Das Datenaustauschformat *railML*® eignet sich nicht zur Beschreibung von multimodalen und intermodalen Vorgängen.
2. Das Datenaustauschformat *railML*® kann besser betriebliche, *NeTEx* besser fahrgastrelevante Informationen beschreiben.
3. *NeTEx* kann einige der vorgegeben Besonderheit im Eisenbahnverkehr, wie gekuppelte Zugfahrten, nicht abbilden.
4. Das Datenaustauschformat *railML*® beherrscht nicht die Beschreibung von Fußwegen sowie die Fußgängernavigation.
5. Die beiden Datenaustauschformate *railML*® und *NeTEx* lassen sich leicht miteinander erweitern, da sie beide auf XML-Basis verfasst sind.

# Inhaltsverzeichnis

Abbildungsverzeichnis .....	VI
Tabellenverzeichnis .....	VII
Abkürzungsverzeichnis .....	VII
1 Motivation und Ziele .....	1
2 Ausgangssituation .....	4
3 Grundlagen .....	6
3.1 NeTEx .....	6
3.1.1 Basisobjekte .....	7
3.1.2 CEN/TS 16614-1: Netzwerktopologie .....	8
3.1.3 CEN/TS 16614-2: Fahrpläne, Fahrten und Fahrzeiten .....	9
3.2 railML® .....	9
4 Vorgehensweise .....	11
5 Analyse und Vergleich von NeTEx und railML® .....	14
5.1 Multimodalität .....	15
5.2 Intermodalität und Fahrplan .....	17
5.2.1 Intermodalität .....	17
5.2.2 Fahrplan .....	20
5.3 Bahnhof und Fußwege .....	26
5.4 Fahrpreise .....	30
5.5 Weitere Herausforderungen beim Erstellen der Beispiele .....	31
5.5.1 Infrastruktur .....	31
5.5.2 Rollmaterial .....	34
5.5.3 Kupplung von Zügen .....	35
5.6 Vergleich .....	38
6 Erweiterungsvorschläge .....	42
6.1 Erweiterungsvorschläge für railML® .....	43
6.2 Erweiterungsvorschläge für NeTEx .....	44
7 Zusammenfassung .....	45
8 Ausblick .....	48
9 Literaturverzeichnis .....	49
10 Anhang .....	VIII

## Abbildungsverzeichnis

Abbildung 1-1: Verbindung zw. einem Datenaustauschformat und dem Fahrgast.....	2
Abbildung 1-2: Screenshot einer Fahrplanauskunft auf einem Smartphone .....	2
Abbildung 3-1: Gliederung <i>NeTEx</i> -Modell.....	7
Abbildung 3-2: <i>NeTEx</i> Teil 1 - Modell.....	8
Abbildung 3-3: <i>NeTEx</i> Teil 2 - Modell.....	9
Abbildung 4-1: Zugfahrten auf der Karte .....	11
Abbildung 4-2: Busfahrt auf der Karte .....	11
Abbildung 4-3: kurze Übersicht der gewählten Beispielfahrten.....	12
Abbildung 5-1: Auszug aus <i>railML</i> -Beispiel - connections.....	18
Abbildung 5-2: Auszug aus <i>NeTEx</i> -Beispiel - journeyInterchanges.....	20
Abbildung 5-3: Auszug aus <i>railML</i> -Beispiel: timetablePeriods .....	21
Abbildung 5-4: Auszug aus <i>railML</i> -Beispiel - operatingPeriods.....	22
Abbildung 5-5: Auszug aus <i>NeTEx</i> -Beispiel - dayTypes .....	23
Abbildung 5-6: Auszug aus <i>NeTEx</i> -Beispiel - dayTypes - Feiertag .....	24
Abbildung 5-7: Auszug aus <i>NeTEx</i> -Beispiel - AvailabilityCondition .....	25
Abbildung 5-8: Auszug aus <i>NeTEx</i> -Beispiel - ServiceJourney.....	26
Abbildung 5-9: Lageplan Bahnhof Treuchtlingen.....	28
Abbildung 5-10: Auszug aus <i>NeTEx</i> -Beispiel - navigationPaths.....	29
Abbildung 5-11: Auszug aus <i>NeTEx</i> -Beispiel - scheduledStopPoints.....	30
Abbildung 5-12: Auszug aus <i>NeTEx</i> -Beispiel - stopAssignments.....	30
Abbildung 5-13: Auszug aus <i>railML</i> -Beispiel - tracks .....	32
Abbildung 5-14: Auszug aus <i>railML</i> -Beispiel – kommerzielle Kupplung von Zügen....	36
Abbildung 5-15: Auszug aus <i>railML</i> -Beispiel - betriebliche Kupplung von Zügen .....	36
Abbildung 5-16: Auszug aus <i>NeTEx</i> -Beispiel - journeyMeetings .....	37
Abbildung 5-17: Auszug aus <i>NeTEx</i> -Beispiel - JourneyPart .....	37
Abbildung 5-18: Auszug aus <i>NeTEx</i> -Beispiel - journeyMeetings unter Call.....	38

## Tabellenverzeichnis

Tabelle 6-1: Zusammenfassung möglicher Erweiterungen.....	42
---	----

## Abkürzungsverzeichnis

<b>ATOC</b>	Association of Train Operating Companies
<b>BMBVBW</b>	Bundesministerium für Verkehr-, Bau- und Wohnungswesen (1998-2005)
<b>BMVBS</b>	Bundesministerium für Verkehr, Bau und Stadtentwicklung (2005-2013)
<b>BMVI</b>	Bundesministerium für Verkehr und digitale Infrastruktur (seit 2013)
<b>EPSG</b>	European Petroleum Survey Group Geodesy
<b>FIS</b>	Fahrgastinformationssystem
<b>Hbf</b>	Hauptbahnhof
<b>ID</b>	Identifikator
<b>IFOPT</b>	Identification of Fixed Objects in Public Transport
<b>NeTEx</b>	Network and Timetable Exchange
<b>ÖPNV</b>	öffentlicher Personennahverkehr
<b>ÖV</b>	öffentlicher Verkehr
<b>RB</b>	Regionalbahn
<b>RE</b>	Regionalexpress
<b>VDV</b>	Verband Deutscher Verkehrsunternehmen

# 1 Motivation und Ziele

Der Fahrgast spielt im öffentlichen Personenverkehr eine entscheidende Rolle, da er die zahlende Funktion in diesem Prozess übernimmt und die Dienstleistung dadurch realisiert wird. Aus diesem Grund ist es wichtig, für eine hohe Kundenzufriedenheit zu sorgen und das Reisen so einfach wie möglich zu gestalten. In einer Befragung von Fahrgästen der *National Rail*<sup>1</sup> in Großbritannien hat sich herausgestellt, dass Fahrgäste Verspätungen von Zügen durchaus tolerieren, solange sie darüber informiert sind. Es hat sich ebenfalls gezeigt, dass Großbritanniens unpünktlichster Betreiber eine höhere Kundenzufriedenheit zu verzeichnen hat, als das pünktlichste Unternehmen. Dies ist darauf zurückzuführen, dass eben dieser unpünktlichste Betreiber seine Kunden jedoch ausführlich über die Verspätung informiert. Die gleiche Befragung ergab außerdem, dass der richtige Umgang mit Verspätungen der beste Weg ist, um die Kundenunzufriedenheit zu reduzieren. [1]

Um einem Fahrgast diese und weitere Informationen zur Verfügung stellen zu können, ist es notwendig, dass die einzelnen Akteure im Verkehrswesen miteinander kommunizieren und gegenseitig Daten austauschen.

Die einzelnen Betreiber nutzen eine Vielzahl unterschiedlicher Softwaremodule, wie zum Beispiel eine Software für die Infrastruktur- oder auch Fahrplandatenbank. Es ist daher notwendig, eine einheitliche Schnittstelle für die einzelnen Module zu schaffen, um Informationsflüsse herzustellen [2]. Datenaustauschformate stellen eine solche Schnittstelle dar. (vgl. Abbildung 1-1)

---

<sup>1</sup> *National Rail* ist ein unternehmensübergreifender Markenname von den in der *Association of Train Operating Companies (ATOC)* vertretenen britischen Eisenbahngesellschaften

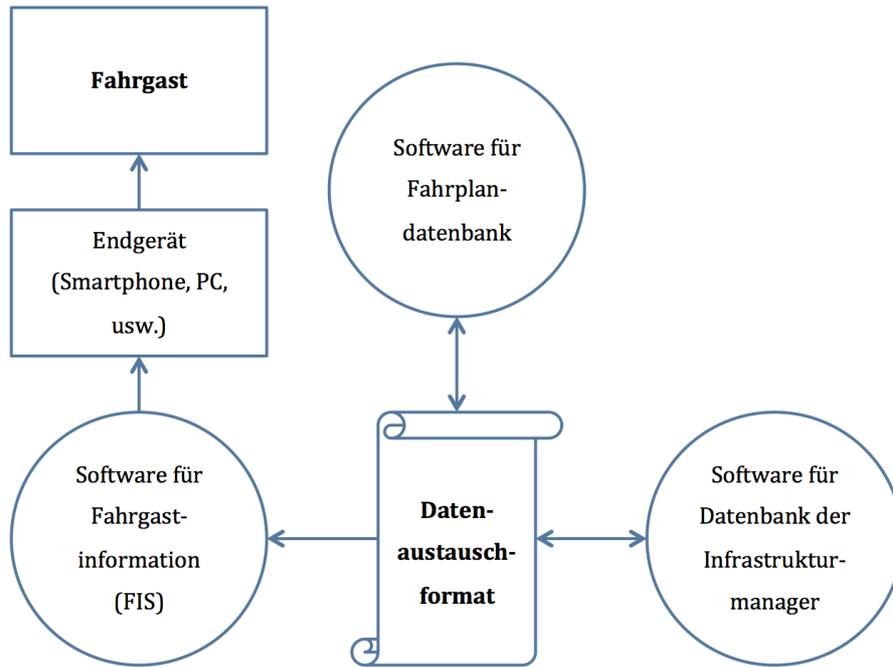


Abbildung 1-1: Verbindung zw. einem Datenaustauschformat und dem Fahrgast

Auch eine Software, die dem Fahrgast Informationen bereitstellt, kann somit auf alle notwendigen Daten zugreifen, die relevanten Informationen extrahieren und diese dem Nutzer anschaulich darstellen. Dieser kann anschließend über ein Endgerät seiner Wahl, z.B. über eine App auf seinem Smartphone, Auskünfte einholen (vgl. Abbildung 1-2).

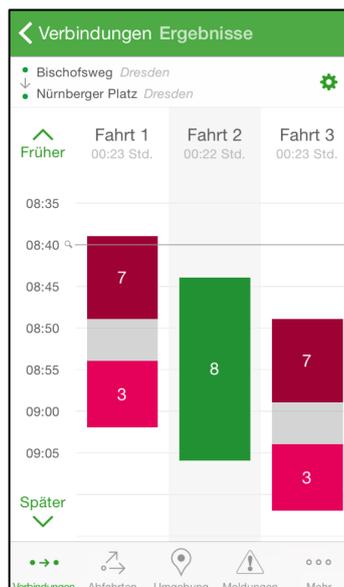


Abbildung 1-2: Screenshot einer Fahrplanauskunft auf einem Smartphone

Der Fahrgast stellt sich vor Antritt einer Fahrt verschiedene Fragen, die ihm dabei helfen, sich für eine bestimmte Verbindung zu entscheiden. Diese Fragestellungen sind folgende:

- Welche Verkehrsmittel stehen zur Auswahl?
- Wie genau komme ich von A nach B?
- Wo muss ich hin?
- Wie hoch ist der Preis?

Im Rahmen dieser Arbeit sollen die beiden Datenaustauschformate *NeTEx* und *railML*® auf ihre Fähigkeit untersucht werden, die Datengrundlage zur Beantwortung dieser Informationen bereitzustellen. Sie sollen anschließend analysiert und miteinander verglichen werden. Dabei soll vor allem auf die Eignung dieser beiden Formate, intermodale Fahrgastinformationen bereitzustellen, eingegangen werden.

Besonders auf diese Fragestellungen hin sollen die beiden Datenaustauschformate *railML*® und *NeTEx* untersucht werden. Das heißt sie sollen anhand verschiedener Reisemöglichkeiten auf einer vorgegebenen Beispielstrecke analysiert und miteinander verglichen werden. Anschließend sind Vorschläge über eine mögliche Erweiterung der Austauschformate bei eventuellen Defiziten zu erbringen.

## 2 Ausgangssituation

Für den Datenaustausch zwischen den einzelnen Akteuren existieren verschiedene speziell auf die Anforderungen des öffentlichen Verkehrs zugeschnittene Schnittstellen und Datenformate. Diese wurden im deutschsprachigen Raum hauptsächlich durch das *Bundesministerium für Verkehr-, Bau- und Wohnungswesen (BMBVBW)* bzw. für *Verkehr, Bau und Stadtentwicklung (BMVBS)* – heute bekannt als *Bundesministerium für Verkehr und digitale Infrastruktur (BMVI)* – und dem *Verband Deutscher Verkehrsunternehmen (VDV)* veranlasst und anschließend als *VDV-Schrift* veröffentlicht. Von Interesse sind hier insbesondere die *VDV-Schriften* 452 (Standardschnittstelle Liniennetz/Fahrplan), 453 (Ist-Daten-Schnittstelle Anschlussicherung – Dynamische Fahrgastinformation – Visualisierung – Allgemeiner Nachrichtendienst) und 454 (Ist-Daten-Schnittstelle Fahrplanauskunft). [3]

Im Rahmen dieser Arbeit ist vor allem die *VDV-Schrift* 452 von Bedeutung. Sie wurde auf Basis des *VDV-ÖPNV-Datenmodells (V.4.1)* entwickelt und dient der Übertragung von Liniennetz- und Fahrplandaten. [2]

Die verschiedenen *VDV-Spezifikationen* sind besonders im deutschsprachigen Raum sehr verbreitet und werden seit vielen Jahren verwendet. Sie genügen jedoch nur noch bedingt den gestiegenen Anforderungen der Verkehrsunternehmen bzw. der Fahrgäste. Des Weiteren ist mit ihnen ein Austausch von Daten mit anderen europäischen Ländern nur schwer möglich, da viele Länder andere Anforderungen haben als im deutschsprachigen Bereich üblich. [3]

Zum besseren Datenaustausch innerhalb des öffentlichen Verkehrs in Europa wurde der neue europäische Standard *NeTEx* ausgearbeitet. Bei der Entwicklung wirkten Experten verschiedener europäischer Länder mit und es sind bereits vorhandene Modelle und Austauschformate für den öffentlichen Verkehr unterschiedlicher Länder berücksichtigt worden. Auch die Übertragung aller Informationen, die bereits in der *VDV-Schnittstelle* 452 integriert sind, wird ermöglicht. Der Umfang von *NeTEx* ist jedoch erheblich größer. [4] *NeTEx* enthält zum Beispiel die Möglichkeit der Fußgänger-

gernavigation innerhalb einer Haltestelle, was die *VDV*-Schnittstelle jedoch nicht betrachtet.

Für die Zuordnung der *VDV*-Elemente zu denen von *NeTEx* hat der *VDV* eine Tabelle entworfen, welche online verfügbar ist<sup>2</sup>.

Ein bisher ausschließlich im Bereich des Schienenverkehrs verwendetes Datenaustauschformat ist *railML*®. Eine Besonderheit hierbei ist, dass es sich um ein Open Source Projekt handelt. Es wird unter anderem von der Deutschen Bahn verwendet, welche zur Planung von Verkehrswegen und deren Auslastung verschiedene Simulationssoftware benutzt. *RailML*® wird dabei verwendet, um Verkehrsdaten, beruhend auf Prognosen des Bundesverkehrsministeriums, auszutauschen und somit Simulationen auf der Infrastruktur durchzuführen. [5]

Weitere Ausführungen zu *NeTEx* und *railML*® erfolgen in den Kapiteln 3.1 und 3.2 dieser Arbeit.

---

<sup>2</sup> Website mit der Tabelle: <https://www.vdv.de/netex.aspx>

## 3 Grundlagen

### 3.1 NeTEx

*NeTEx* steht für „Network and Timetable Exchange“ und ist der europäische Standard für den Austausch von Soll-Daten im Öffentlichen Verkehr, welcher auf XML-Basis verfasst wird. Er wurde im März 2014 als europäische Norm CEN/TS 16614 veröffentlicht, im Dezember 2014 folgte die Veröffentlichung als deutsche Norm DIN CEN/TS 16614. Es handelt sich hierbei momentan um einen Standard, der zwar zertifiziert ist, aber noch nicht im Betrieb verwendet wird.

Eine Besonderheit von *NeTEx* ist, dass es die Möglichkeit zur Beschreibung verschiedener Verkehrsträger bietet, zum Beispiel für:

- Eisenbahnen
- U-Bahnen
- Straßenbahnen
- Busse
- Fähren
- Flüge

Als Grundlage für diesen Standard dienten zwei Referenzdatenmodelle. Diese Modelle beschreiben verschiedene Zusammenhänge, jedoch erst das Datenaustauschformat ermöglicht das Einfügen von Daten.

*Transmodel*<sup>3</sup> ist eines dieser beiden Referenzdatenmodelle. Es beschäftigt sich mit Zusammenhängen im öffentlichen Personennahverkehr (ÖPNV). Es ermöglicht z.B. die Beschreibung eines Verkehrsnetzes sowie eines Fahrplanes. Auch Informationen zu Tarifzonen sind möglich. [6]

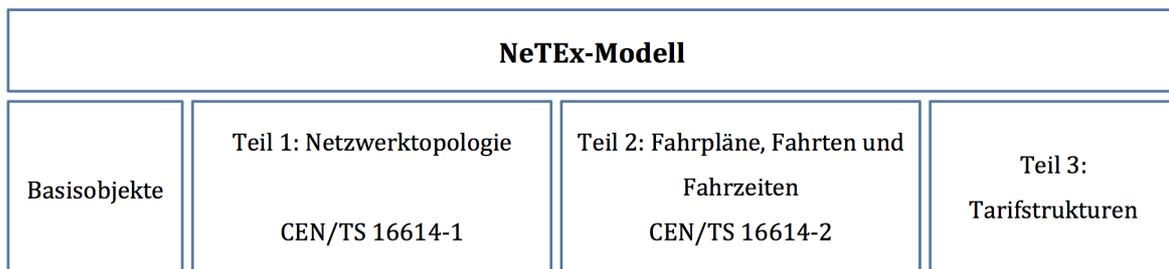
---

<sup>3</sup> EN12896, Referenzdatenmodell für den öffentlichen Verkehr, 2006

Als zweites Modell kam *IFOPT*<sup>4</sup> (Identification of Fixed Objects in Public Transport) zum Einsatz. Es beschreibt alle unbeweglichen Objekte, die mit dem Zugang zum öffentlichen Verkehr zusammenhängen. Das beinhaltet z.B. Eingänge und Wege zum ÖPNV sowie auch Haltestellen an sich. [7]

Beide Modelle können als Grundlage oder zum besseren Verständnis von *NeTEx* herangezogen werden.

Das *NeTEx*-Modell gliedert sich wie folgt [3]:



**Abbildung 3-1: Gliederung *NeTEx*-Modell**

Die Teile 1 und 2 sind jeweils in einer eigenen Norm festgehalten. Teil 1 ist weitgehend autark, wohingegen die Teile 2 und 3 auf den ersten aufbauen. [8]

Teil 3 befindet sich noch im Entwicklungszustand und ist noch nicht zertifiziert. Es wurde daher auf eine genaue Analyse dieses Themas verzichtet, jedoch sind in Kapitel 5.4 „Fahrpreise“ grundlegende Informationen zu diesem Teil zu finden.

### 3.1.1 Basisobjekte

*„Die Basisobjekte enthalten Definitionen, die von allen NeTEx-Teilen verwendet werden und die Grundlage des Datenmodells bilden. Als generische Objekte halten sie in vielen Fällen grundlegende Attribute und Eigenschaftsdefinitionen bereit, die von stärker spezialisierten Objektklassen geerbt werden können. Auf diese Weise sind alle Objektklassen auf gleiche Basisklassen zurückführbar, was das Modell klarer macht und das Verständnis erleichtert.“ [3]*

Die Basisobjekte enthalten zum Beispiel ein Generisches Modell für Punkte (1D) und Strecken (2D) sowie ein Modell für wiederverwendbare Komponenten, welches bei-

<sup>4</sup> EN 28701, Identifizierung fester Objekte im Öffentlichen Verkehr, 2012

spielsweise wiederverwendbare Verfügbarkeiten enthält, das heißt die Standarddefinitionen von Zeiträumen.

### 3.1.2 CEN/TS 16614-1: Netzwerktopologie

Der Teil 1 beschäftigt sich mit der Topologie eines Verkehrsnetzes [9]. Er unterteilt sich in die Beschreibung des Netzwerkes, ein Modell aller festen Objekte, wie Haltestellen, Gebäude und Wege und in die Komponenten der taktischen Planung (vgl. Abbildung 3-2). [3]

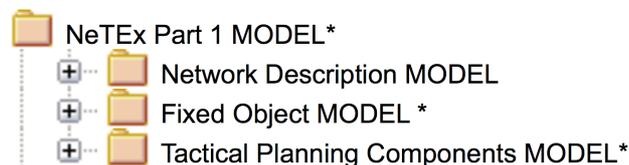


Abbildung 3-2: *NeTEx* Teil 1 - Modell

In der Netzwerkbeschreibung sind unter anderem

- die Netzwerkinfrastruktur,
- das Liniennetz sowie
- das Streckennetz

enthalten.

Dem Modell der fixierten Objekte sind z.B.

- Modelle zur Beschreibung von Haltestellen und
- Wegebeschreibungen

untergeordnet.

Der Teil der taktischen Planung umfasst die Planung aller Arten von Fahrzeugaktivitäten inklusive nachgelagerter Prozesse. Hier ist zum Beispiel eine Verknüpfung zwischen physikalischen Haltestellenobjekten und Fahrplanhaltepunkten hinterlegt. [3]

### 3.1.3 CEN/TS 16614-2: Fahrpläne, Fahrten und Fahrzeiten

„Teil 2 von *NeTEx* beschäftigt sich mit den zeitbehafteten Objekten: Fahrpläne, Fahrten und Fahrzeiten. Dieser Teil ist in vier Bereiche gegliedert (vgl. Abbildung 3-3):

- Fahrten und Fahrzeiten
- Modell der Fahrten eines Betriebstags (Tagesfahrplan)
- Modell der Ankunfts- und Abfahrtszeiten
- Modell der Fahrpläne und Fahrzeugumläufe“ [3]

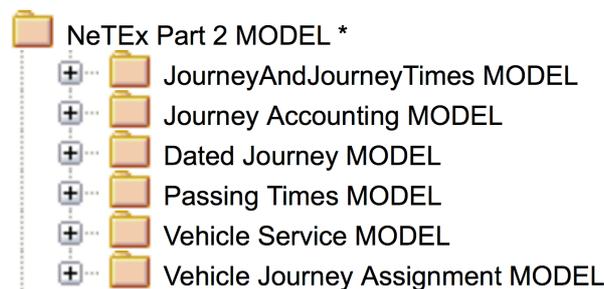


Abbildung 3-3: *NeTEx* Teil 2 - Modell

## 3.2 railML®

*railML*® (im Folgenden nur *railML* genannt) ist ebenfalls ein Datenaustauschformat auf XML-Basis. Allerdings befasst es sich bisher ausschließlich mit dem Verkehrsträger Eisenbahn. Es unterteilt sich in vier große Schemen:

- infrastructure (Eisenbahninfrastruktur)
- rollingstock (Rollmaterial/Fahrzeuge)
- timetable (Fahrplan)
- interlocking (Sicherungstechnik)

Die Infrastruktur beschäftigt sich mit allen Daten bezüglich der Streckenplanung, d.h. mit der Strecke selbst, den Gleisen oder auch Bahnsteigen.

Unter „rollingstock“ werden verschiedene Fahrzeuge mit ihren einzelnen Elementen beschrieben und zum Beispiel Informationen zum Antrieb oder der Sitzplatzanzahl hinterlegt.

Das dritte große Schema „timetable“ beinhaltet alle Informationen zum Fahrplan, das heißt über die Gültigkeit eines Fahrplans oder auch die Fahrzeiten der Züge.

Der vierte Abschnitt beschäftigt sich mit der Sicherungstechnik im Bahnverkehr. Dieses Teilschema hat jedoch keine Relevanz für diese Arbeit.

Bei *railML* handelt es sich um ein Open Source Projekt. Es darf somit von jedem kostenfrei benutzt und auch weiterentwickelt werden.

*„railML® ist ein gemeinsames Projekt mehrerer europäischer Eisenbahnen, entwickelnder Software- und Consultingfirmen sowie internationalen wissenschaftlichen Einrichtungen. (...) Der railML®-Standard wird nicht durch zentrale Instanzen festgelegt. Er entwickelt sich im Arbeits- und Diskussionsprozess der beteiligten Partner.“ [10]*

*RailML* findet bereits Anwendung, Unternehmen die dieses Format zum Teil bereits nutzen sind unter anderem [11]:

- DB Fernverkehr AG
- DB Regio AG
- S-Bahn Berlin
- S-Bahn Hamburg
- Schweizer Bundesbahnen
- Trenitalia

In dieser Arbeit wurde *railML* in der Version 2.2 verwendet. Es ist angedacht, einige Erkenntnisse dieser Arbeit in die Entwicklung von *railML* 3.0 mit einfließen zu lassen, wobei vor allem die Informationen zu intermodalen Zugänglichkeiten von Interesse sind.

Ein Großteil der folgenden Informationen zu *railML* stammt aus mehreren Expertengesprächen mit Susanne Wunsch, die dieses Datenaustauschformat mitentwickelt hat [12].

## 4 Vorgehensweise

Um Defizite und Vorzüge von *railML* und *NeTeX* herauszuarbeiten, wurde eine Beispielstrecke gewählt, die in beiden Datenaustauschformaten modelliert werden sollte. Es wurde die Strecke von München Hauptbahnhof (Hbf) nach Nürnberg Hauptbahnhof vorgegeben, da hier viele unterschiedliche Transportmöglichkeiten vorhanden sind. Zu den verschiedenen Verbindungen zählen (vgl. Abbildung 4-1 und Abbildung 4-2):

- ein langsamer, direkter Regionalverkehr über Treuchtlingen
- eine Nahverkehrsverbindung mit Umstieg über Augsburg und Treuchtlingen
- ein schneller, direkter Regionalverkehr über Ingolstadt
- eine direkte Fernverkehrsverbindung mit gekuppelten Zügen
- Flugverbindung inklusive Zubringer und Wartezeiten
- Fernbus

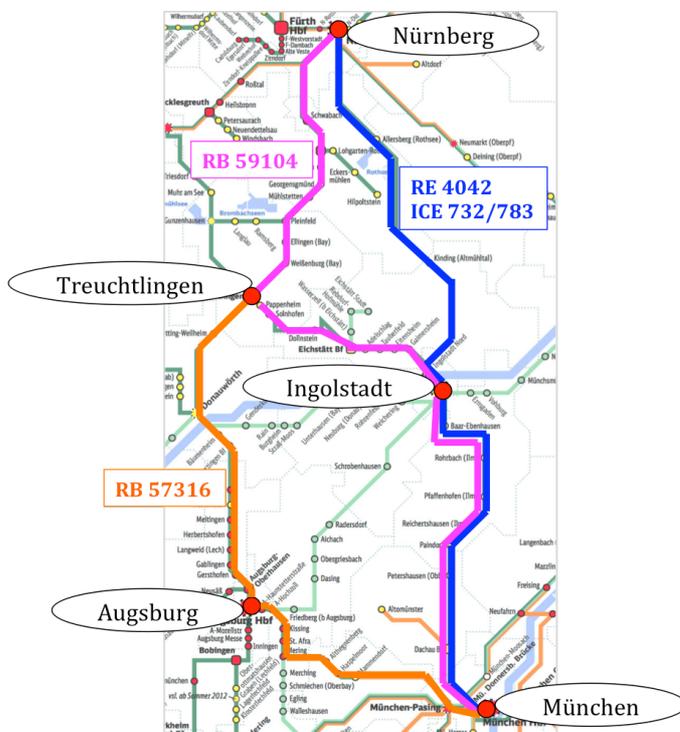


Abbildung 4-1: Zugfahrten auf der Karte [13]



Abbildung 4-2: Busfahrt auf der Karte [14]

Alle Fahrten wurden an einem beliebig gewählten Werktag mit Abfahrt in der Zeit zwischen 17:00 und 20:00 Uhr ausgewählt. Die einzige Ausnahme hiervon bildet der Fernbus, da in der genannten Zeit keine Verbindung zwischen München und Nürnberg vorhanden war. Die Abbildung 4-3 zeigt eine kurze Übersicht aller gewählten Beispielfahrten. Eine ausführliche Tabelle mit Informationen zu den Beispielfahrten ist im Anhang A.1 zu finden.

Stichtag: 16.12.14

abgerufen am: 09.12.14

Verkehrsmittel	Fahrtnummer	Abfahrt		Ankunft		Fahrzeit	Fahrpreis (regulär/spontan)	Bemerkung	
		Ort	Zeit	Ort	Zeit				
Zug	RB 59104	München Hbf	17:29	Nürnberg Hbf	20:17	2:48	35,40 €	langsamer direkter Regionalverkehr	
	RE 57316	München Hbf	17:35	Treuchtlingen	19:20	1:45	35,40 €	Nahverkehr mit Umstieg	
	RB 59104	Treuchtlingen	19:25	Nürnberg Hbf	20:17	0:52			
	gesamt:						2:37		
	ICE 732/782	München Hbf	18:20	Nürnberg Hbf	19:31	1:11	55,00 €	gekuppelter ICE über Schnellfahrstrecke	
RE 4042	München Hbf	19:08	Nürnberg Hbf	20:48	1:40	35,40 €	schneller RE über Ingolstadt		
Flug (inkl. Zubringer)	S8	München Hbf (Tunnel)	17:35	München Flughafen Terminal	18:15	0:40	10,80 €	Direktflug (Inklusive Zubringer und Wartezeiten)	
	LH 2162	MUC	18:55	NUE	19:30	0:35	82,49 €		
	U2	Nürnberg Flughafen	19:52	Nürnberg Hbf (U-Bahn)	20:05	0:13	2,50 €		
	gesamt:						2:05		95,79 €
(Fern-)Bus	MeinFernbus N05	München ZOB	21:30	Nürnberg ZOB	23:40	2:10	19,50 €	Fernbus	

Abbildung 4-3: kurze Übersicht der gewählten Beispielfahrten

Es wurde bei der Wahl der Beispielfahrten darauf geachtet, dass verschiedene interessante und relevante Besonderheiten vorhanden waren.

Bei den Zugfahrten war eine Auswahl von Regionalzügen mit unterschiedlichen Fahrwegen wichtig, um die Nachbildung dieser in den beiden Formaten zu untersuchen.

- Die Regionalbahn **RB 59104** fährt direkt von München (Mü) nach Nürnberg (Nü) über die Bahnhöfe Ingolstadt Hbf (Ing Hbf) und Treuchtlingen (Tr)
  - o Mü - Ing Hbf - Tr - Nü
- Wohingegen der Regionalexpress **RE 4042** ebenfalls über Ingolstadt Hbf, anschließend jedoch direkt über die Schnellfahrstrecke nach Nürnberg weiterfährt.
  - o Mü - Ing Hbf - Nü

- Eine weitere Fahrt ist mit einem Umstieg verbunden. Hier fährt die Regionalbahn **RE 57316** von München über Augsburg (Aug) nach Treuchtlingen und dort findet anschließend ein Umstieg in die Regionalbahn **RB 59104** statt.
  - **Mü - Aug - Tr → Tr - Nü**

Neben den Regionalzügen sollte auch der Fernverkehr Beachtung finden. Beim **ICE 732/782** ist noch eine Besonderheit gegeben, denn hierbei handelt es sich um eine gekuppelte Zugfahrt. Dieser Zug verkehrt kombiniert bis Hannover (Han), wird dort entkoppelt und fährt anschließend als **ICE 732** weiter nach Oldenburg (Oldb) beziehungsweise als **ICE 782** weiter nach Hamburg Altona (HH).

- **Mü - Nü - Han**      **→ Oldb**  
                                  **→ HH**

Außer dem Fernverkehr auf der Schiene sollte auch der Fernbus modelliert werden. Da die exakte Streckenführung des Fernbusses **N05** von *MeinFernbus* nicht bekannt war, wurde eine Strecke über die Autobahn A9 gewählt, wobei der Bus in München-Schwabing auf die A9 auffährt und sie am Dreieck Nürnberg/Feucht wieder verlässt.

Eine weitere Reisemöglichkeit besteht in einem Flug von München nach Nürnberg, welcher ebenfalls in dieser Arbeit betrachtet werden sollte. Hier ist zu beachten, dass sich die Flughäfen in einiger Entfernung zu den Bahnhöfen befinden und somit noch eine Verbindung mit dem Nahverkehr zum bzw. vom Flughafen erforderlich ist.

Bei der Erarbeitung der Beispiele ließen sich sehr gut die Besonderheiten beider Formate aufdecken. Die Ergebnisse sind im folgenden Kapitel 5 verzeichnet.

An dieser Stelle sei noch einmal erwähnt, dass es nicht vorrangig um das Finden der besten Verbindung für die vorgegebene Strecke geht, sondern vielmehr darum, diese speziellen Anwendungsfälle in beiden Datenaustauschformaten umzusetzen und herauszufinden, inwieweit sie sich zur Abbildung dieser Beispiele eignen.

## 5 Analyse und Vergleich von NeTEx und railML®

In dieser Arbeit sollen die beiden Datenaustauschformate *railML* und *NeTEx* vor allem in Hinblick auf die für den Fahrgast relevanten Informationen verglichen werden. Besonders die Intermodalität soll dabei eine große Rolle spielen.

Bevor ein Fahrgast eine Reise antritt, muss er sich verschiedene Fragen stellen, um sich für eine Möglichkeit zu entscheiden. Diese relevanten Fragen sind folgende:

- Welche Verkehrsmittel stehen zur Auswahl? —> **Multimodalität** (Kapitel 5.1)
- Wie genau komme ich von A nach B? —> **Intermodalität /Fahrplan** (Kapitel 5.2)
- Wo muss ich hin? —> **Bahnhof / Fußwege** (Kapitel 5.3)
- Wie hoch ist der Preis? —> **Fahrpreise** (Kapitel 5.4)

Da es sich bei *railML* um ein Datenaustauschformat ausschließlich für den Schienenverkehr handelt, beziehen sich die meisten folgenden Kapitel und Auszüge aus den Beispielen hauptsächlich auf die Beispielzüge. In *NeTEx* werden die Busfahrten jedoch analog zu den Zugfahrten abgebildet, weswegen die meisten Beschreibungen für *NeTEx* in den folgenden Ausführungen ebenso auf Busfahrten anzuwenden sind. Die Flugreise wird im folgenden Verlauf der Arbeit nicht näher erläutert, da bisher keine Beispiele oder ähnliches vorliegen, welche die Beschreibung dieser erklären. Es müsste daher von Grund auf ein Beispiel für einen Flug entwickelt werden, was jedoch den Rahmen dieser Arbeit übersteigen würde.

Des Weiteren wurde bei der Erstellung der Beispieldateien versucht, eine möglichst vollständige Darstellung der gewählten Fahrten abzubilden. Es wurden daher neben Informationen, die für den Fahrgast Relevanz besitzen, auch solche abgebildet, die keine vorrangige Bedeutung für den Fahrgast haben. Die Analyse dieser Themen ist in Kapitel 5.5 „Weitere Herausforderungen beim Erstellen der Beispiele“ zu finden.

Abschließend werden am Ende dieses Kapitels die Erkenntnisse der Analyse vergleichend gegenübergestellt, um eventuelle Defizite besser zu erkennen.

## 5.1 Multimodalität

Multimodaler Verkehr bezeichnet den Transport von Gütern oder Personen mit zwei oder mehreren verschiedenen Verkehrsträgern. [15]

In den beiden hier betrachteten Austauschformaten ergeben sich Unterschiede bei der Modellierung verschiedener Verkehrsmittel.

### *railML*

Wie bereits in den Grundlagen erwähnt, konzentriert sich *railML* bisher ausschließlich auf den Datenaustausch von Eisenbahnsystemen. Eine Multimodalität ist aufgrund dieser Ausrichtung nicht gegeben. Auf dem Gebiet des Eisenbahnverkehrs ist mit dieser Schnittstelle jedoch eine äußerst detaillierte Beschreibung, vor allem der betrieblich notwendigen Informationen, möglich. Es ist zum Beispiel möglich, viele technische Details über das verwendete Rollmaterial zu hinterlegen, z.B. Informationen über die maximal zulässige Geschwindigkeit, die Funkausstattung oder auch die Bremsen und Achsen eines Fahrzeugs. Dies kann aus betrieblicher Sicht entscheidend sein für die Wahl der Bahnstrecke, denn beispielsweise dürfen nicht alle Strecken mit hohen Geschwindigkeiten befahren werden.

### *NeTEx*

*NeTEx* hat es sich zur Aufgabe gemacht, ein Standard-Datenaustauschformat für den gesamten öffentlichen Verkehr darzustellen. Dies beinhaltet viele verschiedene Verkehrsträger. Innerhalb dieser Arbeit von besonderer Bedeutung sind:

- Eisenbahn
- Bus
- Flug.
  
- Eisenbahn

Die Modellierung des Eisenbahnverkehrs ist in der Schnittstelle *NeTEx* integriert. Vor allem die Beschreibung der für den Fahrgast relevanten Informationen ist sehr gut möglich. Allerdings ist für die Angabe der betrieblichen Daten nicht immer die notwendige Detailtiefe vorhanden, es fehlt zum Beispiel die Möglichkeit, eine in Deutschland verbreitete bahninterne Streckenkilometrierung und -bezeichnung zu

hinterlegen. Des Weiteren werden die Dateien bei größeren Netzstrukturen, wie z.B. in Fernverkehrsnetzen, schnell unübersichtlich und sehr komplex, da sehr viele zusätzliche Informationen eingetragen werden können. Viele dieser Informationen sind jedoch nicht obligatorisch.

### - Bus

Da *NeTEx* vorrangig für die Beschreibung des städtischen Verkehrs entwickelt wurde, lassen sich Busfahrten sehr gut modellieren. Fernbusreisen unterscheiden sich in ihren Anforderungen nur geringfügig von denen des innerstädtischen Busverkehrs und sind daher einfach abbildbar.

### - Flug

Da in der Norm CEN/TS 16614, vor allem in der grundlegenden Beschreibung des Datenaustauschformates *NeTEx*, oft auch Flugreisen Erwähnung finden, soll diese Arbeit auch die Möglichkeit der Darstellung von Flügen untersuchen. Dabei soll es nicht um betriebliche Abläufe sondern um die für den Fahrgast relevanten Informationen gehen.

Bei der Untersuchung des Standards hat sich herausgestellt, dass keine Beispiele oder Ansätze für die Modellierung einer Flugreise vorhanden sind. Für Flüge sollen vor allem die bereits vorhandenen Elemente der anderen Verkehrsträger genutzt werden. Eine detaillierte Beschreibung von Flughäfen, um darauf aufbauend Fluggäste zu bestimmten Einrichtungen zu navigieren, ist sicher umsetzbar. Dieses Beispiel müsste jedoch von Grund auf entwickelt werden, was den Zeitrahmen dieser Arbeit überschreiten würde.

### - Sonstige Verkehrsmittel

Auch die Beschreibung weiterer Verkehrsmittel ist mit Hilfe von *NeTEx* möglich. So könnten ebenfalls zum Beispiel U-Bahnen und Straßenbahnen modelliert werden.

## 5.2 Intermodalität und Fahrplan

Unter Intermodalität versteht man

*„ein Transportsystem, mit dessen Hilfe zwei oder mehrere Verkehrsträger verwendet werden, um dieselbe Ladeinheit oder dasselbe Straßenfahrzeug in einer integrierten Art und Weise, ohne Be- oder Entladung, zu einer Haus-zu-Haus Transportkette zu vervollständigen.“ [15]*

In diesem Kapitel werden vor allem die Umstiegsbeziehungen – sowohl zwischen Zügen als auch zwischen unterschiedlichen Verkehrsmitteln betrachtet. Des Weiteren ist auch die Modellierung des Fahrplans inklusive verschiedener Gültigkeiten von Interesse.

Im Beispiel wurde in beiden Austauschformaten der Umstieg in Treuchtlingen vom *RE 57316* zum *RB 59104* modelliert.

Bei den Gültigkeiten gab es bei den gewählten Beispielen einige Besonderheiten:

- Der *RE 4042* fährt die Strecke München-Ingolstadt nur von Montag bis Freitag und nicht an Feiertagen, wohingegen er von Ingolstadt nach Nürnberg täglich fährt.
- Der gekuppelte *ICE 732/782* fährt in bestimmten Zeiträumen innerhalb des Fahrplanjahres von Montag bis Freitag (6. Januar bis 27. März 2015; 24. August bis 11. Dezember 2015) und sonst verkehrt er täglich. Ebenso verhält es sich mit dem *ICE 732* von Hannover nach Oldenburg (Oldb), wobei sich die Zeiträume unterscheiden

In den folgenden Kapiteln soll untersucht werden, ob es möglich ist, diese speziellen Anforderungen in beiden Datenaustauschformaten abzubilden und es soll ebenfalls erklärt werden, wie diese Beispiele umgesetzt wurden.

### 5.2.1 Intermodalität

#### *railML*

Aufgrund der Auslegung von *railML* für den Schienenverkehr ist hier nur die Beschreibung eines Umstieges zwischen Zügen umsetzbar. Es ist beispielsweise nicht möglich, einen Bahnhof von einer Bushaltestelle zu unterscheiden, da es keine Attribute gibt, die dies näher bestimmen können.

Im Code von railML wird die Information über den Umstieg unter „timetable“ und dort „ocpTT“ hinterlegt. „ocpTT“ beschreibt einen Bahnhof innerhalb eines Fahrplans und für den Umstieg wird unter dem Bahnhof, an dem der Umstieg stattfinden soll, eine Verbindung („connection“) angelegt (vgl. Abbildung 5-1). Hier wird nun hinterlegt, in welchen Zug an dieser Station umgestiegen werden kann. Es können noch zusätzliche Informationen angegeben werden, z.B. eine minimale oder maximale Wartezeit und auch, ob es sich um eine betrieblich oder für den Fahrgast relevante Verbindung handelt (connType='commercial'). In dem Beispiel aus Abbildung 5-1 ist ebenfalls noch connOperation='IsExpectedBy' enthalten. Hierbei wird die Art der Verbindung beschrieben – in dem Beispiel bedeutet es, dass der *RE 57316* von dem anderen Zug *RB 59104* mit der Zugreferenz trainRef='trc2' erwartet wird und die maximale Umstiegszeit (laut Fahrplan) fünf Minuten beträgt (maxConnTime='PT5M'). Im Umkehrschluss wird unter „trainPart“ für den *RE 59104* im Bahnhof Treuchtlingen auch eine Verbindung („connection“) eingetragen, bei der folgendes gilt: connOperation='IsWaitingFor'.

```

<trainPart id="tp04" trainNumber="57316" processStatus="planned" description="Muenchen Hbf - Treuchtlingen ueber Ingol
  <formationTT formationRef="form01" />
  <operatingPeriodRef ref="op02" />
  <ocpsTT>
    <ocpTT ocpRef="ocp01" ocpType="begin">
      <times scope="scheduled" departure="17:35:00" />
      <sectionTT section="sect01" distance="17800">
        <trackRef ref="tr06" />
      </sectionTT>
    </ocpTT>
    ...
    <ocpTT ocpRef="ocp19" ocpType="end">
      <times scope="scheduled" arrival="19:20:00" />
      <connections>
        <connection trainRef="trc2" connOperation="IsExpectedBy" connType="commercial"
          trainPartRef="tp03" maxConnTime="PT5M" />
      </connections>
    </ocpTT>
  </ocpsTT>
</trainPart>

```

Abbildung 5-1: Auszug aus railML-Beispiel - connections

**NeTEx**

In diesem Standarddatenaustauschformat werden die Umstiege im „ServiceFrame“ hinterlegt. Hier können Verbindungen zwischen Verkehrsmitteln unterschiedlich detailliert und vor allem intermodal beschrieben werden. *NeTEx* unterscheidet dabei folgende Möglichkeiten:

- Umstieg zwischen zwei Fahrten
- Umstieg zwischen Fahrtmustern
- Hinterlegen der Information, dass ein Umstieg besteht (auch in andere Verkehrsmittel)
  
- Umstieg zwischen zwei Fahrten

Hierbei wird der Umstieg zwischen zwei ganz genau definierten Fahrten angegeben. Das könnte z.B. der Umstieg von Zug A zu einer bestimmten Uhrzeit in den Bus X zu einer genauen Zeit ein paar Minuten später sein. Im Beispiel wurde diese Variante gewählt, um den Umstieg vom *RE 57316* zum *RB 59104* zu realisieren. *NeTEx* bietet ebenfalls die Möglichkeit, die Wege zwischen den Umstiegen zu modellieren. Dies wird im folgenden Kapitel 5.3 „Bahnhof und Fußwege“ näher erläutert.

Diese Möglichkeit wurde in den Beispieldaten umgesetzt. Für die Verbindung von München nach Nürnberg im *RE 57316* und in der *RB 59104* ist ein Umstieg in Treuchtlingen notwendig. Zunächst wird im „ServiceFrame“ eine Verbindung („connections“) angelegt, welche nur aussagt, dass ein Umstieg von einer Haltestelle zu einer anderen möglich ist. Auch der Umstieg innerhalb des gleichen Bahnhofs kann hiermit abgebildet werden. In diesem Fall wird ein Umstieg von Treuchtlingen nach Treuchtlingen definiert.

Der nächste Teil zur Beschreibung des Umstiegs findet im „TimetableFrame“ statt. Dort gibt es das Element „journeyInterchanges“. Hier wird die Verbindung („connections“) referenziert, ebenso die Haltestellen, an denen der Umstieg stattfindet (hier in beiden Fällen Treuchtlingen). Außerdem muss noch die Zugfahrt („serviceJourney“) von der bzw. in die umgestiegen wird, angegeben werden. (vgl. Abbildung 5-2)

```

<journeyInterchanges>
  <ServiceJourneyInterchange version="any" id="SJI01">
    <Description>Umstieg in Treuchtlingen nach Nürnberg</Description>
    <ConnectionRef ref="C01"/>
    <StaySeated>false</StaySeated>
    <FromPointRef ref="SSP19"/>
    <ToPointRef ref="SSP19"/>
    <FromJourneyRef ref="SJ04"/>
    <ToJourneyRef ref="SJ03"/>
  </ServiceJourneyInterchange>
</journeyInterchanges>

```

Abbildung 5-2: Auszug aus NeTeX-Beispiel - journeyInterchanges

- Umstieg zwischen zwei Fahrtmustern

NeTeX bietet die Möglichkeit, anstatt genauer Fahrten auch Fahrtmuster anzulegen. Dies ist dann sinnvoll, wenn zum Beispiel eine Straßenbahnlinie aller zehn Minuten immer zur gleichen Minute abfährt. Dann wird nur ein Muster dieser Fahrt angelegt und nicht für jede einzelne Fahrt ein Abschnitt. Wenn sich nun die regelmäßig verkehrende Straßenbahn X und der immer gleich abfahrende Bus Y an einer Haltestelle treffen und ein Umstieg möglich ist, wird die Verbindung zwischen diesen beiden Fahrtmustern angegeben.

- Hinterlegen der Information, dass ein Umstieg möglich ist

Die einfachste Variante besteht darin, dass an einer Haltestelle hinterlegt werden kann, welche anderen Verkehrsmittel dort noch verkehren. Dies wäre zum Beispiel der Fall, wenn an einem Bahnhof angezeigt wird, dass hier ein Umstieg in eine U-Bahn, Straßenbahn oder auch einen Bus besteht, ohne jedoch eine konkrete Linie oder genaue Zeiten anzugeben.

## 5.2.2 Fahrplan

Da sich diese Arbeit vor allem auf den Fahrgast konzentrieren soll, wird in diesem Kapitel die Beschreibung des Aushangfahrplans betrachtet. Es gibt im Bahnverkehr noch andere Fahrpläne, z.B. Umlaufpläne, welche im Folgenden jedoch nicht betrachtet werden, da sie für den Fahrgast nicht von Interesse sind.

In diesem Kapitel wird zunächst beschrieben, wie verschiedene Fahrplangültigkeiten beschrieben werden können, anschließend wird auf das Zusammensetzen eines Fahrplans eingegangen. Alle Angaben sind analog auch für den Busfernverkehr verwendbar.

**railML**

In *railML* wird das Fahrplanjahr, also der Zeitraum, in dem die Zugfahrt festgelegt ist, als „timetablePeriod“ direkt unter „timetable“ festgelegt. Dort werden ebenfalls die allgemeinen Feiertage hinterlegt (vgl. Abbildung 5-3).

```
<timetablePeriods>
  <timetablePeriod id="ttp01" description="Fahrplan 2015" startDate="2014-12-14" endDate="2015-03-28">
    <holidays>
      <holiday holidayDate="2014-12-25" description="1. Weihnachtstag" />
      <holiday holidayDate="2014-12-26" description="2. Weihnachtsfeiertag" />
      <holiday holidayDate="2015-01-06" description="Heilige Drei Koenige" />
      <holiday holidayDate="2015-04-03" description="Karfreitag" />
      <holiday holidayDate="2015-04-06" description="Ostermontag" />
      <holiday holidayDate="2015-05-01" description="Tag der Arbeit" />
      <holiday holidayDate="2015-05-14" description="Christi Himmelfahrt" />
      <holiday holidayDate="2015-05-25" description="Pfingstmontag" />
      <holiday holidayDate="2015-06-04" description="Fronleichnam" />
      <holiday holidayDate="2015-08-15" description="Mariae Himmelfahrt" />
      <holiday holidayDate="2015-10-31" description="Tag der deutschen Einheit" />
      <holiday holidayDate="2015-11-01" description="Allerheiligen" />
    </holidays>
  </timetablePeriod>
</timetablePeriods>
```

Abbildung 5-3: Auszug aus *railML*-Beispiel: timetablePeriods

Die Zuweisung, dass eine Zugfahrt nicht an Feiertagen verkehrt, etwa in der Art „Mo-Fr, nicht an allgemeinen Feiertagen, nicht 24.12. und 31.12.“, erfolgt unter „operatingPeriod“.

*„Die Angabe operatingCode='1111100' bedeutet, dass die Regelung an den ersten fünf Wochentagen (= Montag bis Freitag) verkehrt. Die Angabe operatingCode='0000000' holidayOffset='0' bedeutet, dass die Regelung abweichend davon nicht verkehrt (=0000000), wenn der jeweilige Tag auf einen Feiertag fällt (holidayOffset=0 - „null Tage Versatz zu einem Feiertag“ hat).“ [16]*

Das Element „holidayOffset“ bezeichnet einen Versatz. Das heißt, es gibt an, viele Tage vor oder nach einem Feiertag eine Fahrt (nicht) stattfindet. Dies ist zum Beispiel notwendig, wenn ein Güterzug länger als einen Tag benötigt, um an sein Ziel zu gelangen und seine Waren nicht an einem Feiertag zustellen kann. Dafür definiert der Versatz, dass dieser Güterzug auch einen Tag vor Feiertagen nicht verkehren soll, da er sonst am Feiertag ankommen würde. Dies würde mit operatingCode='0000000' und holidayOffset='-1' angegeben werden.

Weitere einzelne Tage, an denen ein Zug nicht verkehrt, zum Beispiel Heilig Abend (24.12.), können mit Hilfe von „specialService“ hinzugefügt werden. Dass ein Zug an einem bestimmten Datum nicht fährt, wird über `type='exclude'` definiert, soll ein Datum eingefügt werden, an dem trotz der übrigen Bedingungen ein Zug fährt, wird dies über `type='include'` erreicht. (vgl. Abbildung 5-4)

```
<operatingPeriods>
  <operatingPeriod id="op05" description="Mo-Fr, nicht an allgemeinen Feiertagen, nicht 24.12. und 31.12.">
    <operatingDay operatingCode="1111100">
      <operatingDayDeviance operatingCode="0000000" holidayOffset="0"/>
    </operatingDay>
    <specialService type="exclude" singleDate="2014-12-24"/>
    <specialService type="exclude" singleDate="2014-12-31"/>
  </operatingPeriod>
</operatingPeriods>
```

Abbildung 5-4: Auszug aus *railML*-Beispiel - `operatingPeriods`

Nachdem die Gültigkeiten definiert wurden, legt man in *railML* sogenannte „trainParts“ an. Dieses Element enthält die Ankunfts- beziehungsweise Abfahrtszeiten, sowie die Referenz zu einer bestimmten Zugformation und einer „operatingPeriod“. Der Sinn im Anlegen von Zugteilen („trainParts“), welche später unter „trains“ zu einem Zug zusammengesetzt werden, zeigt sich bei gekuppelten Zugfahrten oder bei Zügen mit unterschiedlichen Gültigkeiten. Das Kuppeln von Zügen wird in Kapitel 5.5.3 näher beschrieben.

Die Beispielfahrten enthalten einen Zug mit den eben erwähnten geteilten Gültigkeiten. Der *RE 4042* verkehrt von München bis Ingolstadt Hbf mit folgender Gültigkeit: Mo-Fr, nicht an allgemeinen Feiertagen, nicht 24.12. und 31.12. Von Ingolstadt Hbf bis Nürnberg fährt dieser Zug jedoch täglich. Um diese Bedingungen abbilden zu können, wurden für den *RE 4042* zwei Zugteile („trainParts“) angelegt – jeweils einer von München bis Ingolstadt Hbf und ein zweiter von Ingolstadt Hbf bis Nürnberg. Jeder Zugteil bekam anschließend eine eigene „operatingPeriod“ zugewiesen.

Zusammengesetzt werden die Züge dann, wie bereits erwähnt, unter „trains“. Hier wird aufgezählt, ob es sich um eine betriebliche oder fahrgastrelevante Zugfahrt handelt (wichtig beim Kuppeln von Zügen).

Für das Zusammensetzen von Zugfahrten gibt es, je nach Anforderung, zwei Varianten. Zum einen gibt es den eben beschriebenen Beispielfall, bei dem topologisch und zeitlich nacheinander zwei Zugfahrten stattfinden. In diesem Fall

werden unter „train“ zwei „trainPartSequence“ angelegt. Im Kapitel 5.5.3 „Kupplung von Zügen“ wird dann die zweite Variante beschrieben, bei der zwei „trainParts“ in einer „trainPartSequence“ zusammengefasst werden.

### **NeTEx**

Bei diesem Datenaustauschformat werden die Fahrplangültigkeiten in zwei verschiedenen Frames definiert:

- im „ServiceCalendarFrame“ und
- im „TimetableFrame“.

#### - Service Calendar Frame

In diesem Abschnitt wird festgelegt, an welchen Tagen ein Zug fährt, zum Beispiel Montag-Freitag, täglich oder nicht an Feiertagen. Dabei werden hier alle Möglichkeiten angelegt, die später für die zu beschreibenden Fahrpläne notwendig sind. Zuerst einmal wird allgemein das Fahrplanjahr angegeben, anschließend folgt die Eingabe der verschiedenen Tages-Typen, „dayTypes“ in *NeTEx*. Unter „Name“ wird die Beschreibung, wie sie auch im Fahrplan zu finden ist, übernommen. Bei „PropertyOfDay“ und dort „DaysOfWeek“ werden die betreffenden Wochentage hinterlegt, indem sie mit Hilfe der vorgegebenen Attributen aufgezählt werden. Es ist hier zum einen die Eingabe jedes einzelnen Tages möglich, zum anderen kann noch aus „Everyday“ für täglich, „Weekdays“ für Wochentage oder auch „Weekends“ für das Wochenende gewählt werden. Um zu definieren, dass an Feiertagen keine Züge verkehren wird bei „HolidayTypes“ „NotHoliday“ ausgewählt. (vgl. Abbildung 5-5)

```
<DayType version="any" id="DT03">
  <Name>täglich, nicht an allgemeinen Feiertagen</Name>
  <properties>
    <PropertyOfDay>
      <DaysOfWeek>Everyday</DaysOfWeek>
      <HolidayTypes>NotHoliday</HolidayTypes>
    </PropertyOfDay>
  </properties>
</DayType>
```

**Abbildung 5-5: Auszug aus NeTEx-Beispiel - dayTypes**

Anschließend werden in diesem Abschnitt noch „dayTypes“ für die Feiertage angelegt. *NeTEx* erkennt durch die Zuordnung bestimmter Attribute wie „NationalHoliday“, „LocalHoliday“ oder „EveOfHoliday“, dass es sich um einen Feiertag handelt, welcher bei „NotHoliday“ (vgl. Abbildung 5-5) benötigt wird. Dazu wird

anstatt von Tagen innerhalb einer Woche ein bestimmter Tag im Jahr angegeben („DayOfYear“), wobei das Jahr durch einen Bindestrich ersetzt wird, da es sich um einen jährlich stattfindenden Feiertag handelt. (vgl. Abbildung 5-6)

```
<DayType version="any" id="DT14">
  <Name>Mariae Himmelfahrt</Name>
  <properties>
    <PropertyOfDay>
      <DayOfYear>--08-15</DayOfYear>
      <HolidayTypes>LocalHoliday</HolidayTypes>
    </PropertyOfDay>
  </properties>
</DayType>
```

Abbildung 5-6: Auszug aus NeTEx-Beispiel - dayTypes - Feiertag

Für die Tage beziehungsweise Abende vor Feiertagen, wie zum Beispiel Heilig Abend (24.12.) oder Silvester (31.12.), gibt es das eigene Attribut „EveOfHoliday“.

#### - Timetable Frame

Zu Beginn dieses Abschnitts kann noch einmal die Gültigkeit des Fahrplanjahres mit Hilfe von „validityConditions“ angegeben werden. Dies ist auch in anderen Frames auf diese Weise möglich [17].

Anschließend können unter „contentValidityConditions“ Gültigkeiten eingetragen werden, die häufig von den Schema-Elementen verwendet werden. Dies hat den Vorteil, dass sie nur jeweils einmal zentral definiert werden und dann von den einzelnen Elementen lediglich referenziert werden müssen. [17]

Gültigkeiten werden in NeTEx mit Hilfe von „AvailabilityCondition“ definiert. Dieses Element zählt zu den sogenannten Basisobjekten (vgl. Kapitel 3.1.1) und kann somit von allen Teilen und in verschiedenen Bereichen verwendet werden. Sowohl unter „validityConditions“ als auch bei „contentValidityConditions“ werden nun also entsprechend ihrer Aufgaben „AvailabilityCondition“ angegeben.

In „contentValidityConditions“ werden anschließend alle vorher bereits im „ServiceCalendarFrame“ angelegten Gültigkeiten so zusammengefasst, dass die Gültigkeiten der Fahrpläne abgebildet werden können. Bei dem Beispiel besitzt der RE 4042 von München bis Ingolstadt folgende Gültigkeit: Mo-Fr, nicht an allgemeinen Feiertagen, nicht 24.12. und 31.12. Unter „AvailabilityCondition“ wird zuerst einmal die Gültigkeit des Fahrplans eingetragen, welche durchaus von dem allgemeinen Fahrplanjahr abweichen kann. Im Beispiel gilt der Fahrplan z.B. nur vom 14.12.2014 bis 28.03.2015. Unter „dayTypes“ werden anschließend alle für diesen Fahrplan

zutreffenden Tages-Typen referenziert. Beim *RE 4042* sind das die IDs für „täglich, nicht an Feiertagen“ und die aller Feiertage. Es ergibt sich eine sehr lange „AvailabilityCondition“ (vgl. Abbildung 5-7).

```
<contentValidityConditions>
  <AvailabilityCondition version="any" id="AC02">
    <Description>nicht an allgemeinen Feiertagen, nicht 24.12. und 31.12.</Description>
    <FromDate>2014-12-14T00:00:00Z</FromDate>
    <ToDate>2015-03-28T24:00:00Z</ToDate>
    <IsAvaiLable>true</IsAvaiLable>
    <dayTypes>
      <DayTypeRef ref="DT03"/>
      <DayTypeRef ref="DT04"/>
      <DayTypeRef ref="DT05"/>
      <DayTypeRef ref="DT06"/>
      <DayTypeRef ref="DT07"/>
      <DayTypeRef ref="DT08"/>
      <DayTypeRef ref="DT09"/>
      <DayTypeRef ref="DT10"/>
      <DayTypeRef ref="DT11"/>
      <DayTypeRef ref="DT12"/>
      <DayTypeRef ref="DT13"/>
      <DayTypeRef ref="DT14"/>
      <DayTypeRef ref="DT15"/>
      <DayTypeRef ref="DT16"/>
      <DayTypeRef ref="DT17"/>
      <DayTypeRef ref="DT18"/>
    </dayTypes>
  </AvailabilityCondition>
```

**Abbildung 5-7: Auszug aus NeTEx-Beispiel - AvailabilityCondition**

Im „TimetableFrame“ werden des Weiteren auch die Ankunfts- beziehungsweise Abfahrtszeiten angegeben. Dies wird mit Hilfe des Elements „vehicleJourneys“ umgesetzt. Es gibt hierbei zwei unterschiedliche Arten von Fahrten. „ServiceJourney“ enthält alle Fahrplanangaben von Fahrten, die für Fahrgäste bestimmt sind. Sie referenziert unter anderem die benötigte Verfügbarkeit („AvailabilityCondition“) und das verwendete Fahrzeug („trainRef“) (vgl. Abbildung 5-8). Diese Information ist vor allem bei Zugfahrten von großem Interesse, da die Fahrzeiten unter anderem abhängig vom verwendeten Rollmaterial sind. Außerdem ist es möglich einen „DeadRun“ zu definieren, welcher alle Leerfahrten ohne Fahrgäste beinhaltet.

Um dem Fahrplan nun konkrete Fahrzeiten hinzuzufügen, benötigt man das Element „calls“. Ein „call“ referenziert einen Haltepunkt und fügt ihm die Ankunfts- und Abfahrtszeit des Verkehrsmittels an dieser Haltestelle hinzu.

```

<vehicleJourneys>
  <ServiceJourney version="any" id="SJ01">
    <validityConditions>
      <AvailabilityConditionRef ref="AC02"/>
    </validityConditions>
    <Description>RE 4042</Description>
    <DepartureTime>19:08:00</DepartureTime>
    <TrainRef ref="T01"/>
    <PublicCode>RE 4042</PublicCode>
    <calls>
      <Call id="Ca01" order="1">
        <ScheduledStopPointRef ref="SSP01"/>
        <Arrival>
          <ForAlighting>false</ForAlighting>
        </Arrival>
        <Departure>
          <Time>19:08:00</Time>
        </Departure>
      </Call>
    </calls>
  </ServiceJourney>
</vehicleJourneys>

```

Abbildung 5-8: Auszug aus NeTEx-Beispiel - ServiceJourney

## 5.3 Bahnhof und Fußwege

### *railML*

Bahnhöfe werden in *railML* als „operationControlPoints“ (ocp) bezeichnet. Diese werden einzeln unter „infrastructure“ angelegt und können zum Beispiel mit ihren Koordinaten versehen werden. Die Einbindung der Bahnhöfe in die Netzstruktur erfolgt mit Hilfe des Attributs „trackTopology“ unter „track“.

Für die Angabe der Koordinaten wird ein bestimmtes Format benötigt – abhängig vom verwendeten Koordinatensystem. Um herauszufinden, wie die Koordinaten hinterlegt werden müssen, muss in der EPSG Datenbank nachgelesen werden. Auf der Website der Datenbank<sup>5</sup> kann mit Hilfe von bestimmten Schlagwörtern nach dem Code für das benötigte Koordinatensystem gesucht werden. Im Beispieldatensatz wurde das System WGS 84 verwendet. Der EPSG-Code kann ebenfalls innerhalb von „geoCoord“ angegeben werden. In der Datenbank ist auch die Angabe der Achsen angegeben, bei dem hier verwendeten Koordinatensystem werden zuerst die Breiten-, dann die Längengrade genannt.

Es ist in *railML* nur schwer möglich, bestimmte, für den Fahrgast notwendige Informationen zu hinterlegen. Dieses Datenaustauschformat betrachtet den Bahnhof eher aus betrieblicher Sicht. Die Nummern der Gleise („track“) beziehen sich auf die bahninterne Streckenbezeichnung, nicht auf die Bahnsteignummer in einem Bahnhof. Es ist nicht möglich, bestimmte Abschnitte auf einem Bahnsteig zu markieren, um zum

<sup>5</sup> <http://www.epsg-registry.org>

Beispiel über die Wagenposition von Fernverkehrszügen zu informieren. Allerdings können zum Beispiel die Positionen von Haltetafeln, an denen ein Zug bestimmter Länge halten soll, angegeben werden.

### **NeTEx**

Dieses Standard-Datenaustauschformat besitzt zwei unterschiedliche Elemente für Bahnhöfe. Es gibt einmal

- die Beschreibung des Bahnhofs aus Fahrgastsicht, welche sich „StopPlace“ nennt und
- die Beschreibung von Bahnhöfen innerhalb der Infrastruktur, „ScheduledStopPoint“ genannt.

Diese beiden Teile werden mit Hilfe des Elements „PassengerStopAssignment“ miteinander verknüpft. [18]

#### - StopPlace

Dieses Element beschreibt die Haltestellen aus Sicht des Fahrgastes. Das heißt, es gibt einen gemeinsamen Namen, jedoch sind unterschiedliche Bereiche innerhalb der Haltestelle möglich – zum Beispiel für Bahn, Bus und andere Verkehrsmittel. [18]

Physisch handelt es sich um verschiedene Haltestellen, für den Fahrgast haben sie jedoch alle den gleichen Namen. Es ist auch möglich, die Wege zwischen den unterschiedlichen Haltestellen von Bus und Bahn zu beschreiben und so eine Fußgängernavigation bereit zu stellen.

Das StopPlace-Modell ist sehr umfangreich - die meisten Planungssysteme beinhalten jedoch nicht dieses Maß an Informationen. Es ist möglich, alle Bestandteile einer Haltestelle zu beschreiben – zum Beispiel die Ticketautomaten, Eingänge mit Öffnungszeiten oder auch Lifte und Treppen. Des Weiteren wird unter „StopPlace“ hinterlegt, wie viele Gleise ein Bahnhof hat, wie die Plattformen zusammenhängen und welche Abschnitte es auf einem Bahnsteig gibt. In den Beispieldateien wurden diese Informationen zur Veranschaulichung in München Hbf eingepflegt. Außerdem wurde auch Treuchtlingen genauer beschrieben als andere Haltestellen, da dies eine Voraussetzung für die Fußgängernavigation ist.

### Fußwege innerhalb einer Haltestelle

Innerhalb einer Haltestelle müssen Fahrgäste verschiedene Wege zurücklegen – zum Beispiel von einem Gleis zum anderen oder auch von einem Eingang zu einer Halteposition. Um diese Wege zu beschreiben, ist es notwendig, alle Elemente, die passiert werden müssen, anzulegen. Im Beispieldatensatz wurden im Bahnhof Treuchtlingen die Wege beschrieben, um vom *RE 57316* auf Gleis 4 in den *RB 59104* auf Gleis 3 umzusteigen. In Treuchtlingen muss eine Unterführung genutzt werden, um von Gleis 4 zu Gleis 3 zu gelangen (vgl. Abbildung 5-9).

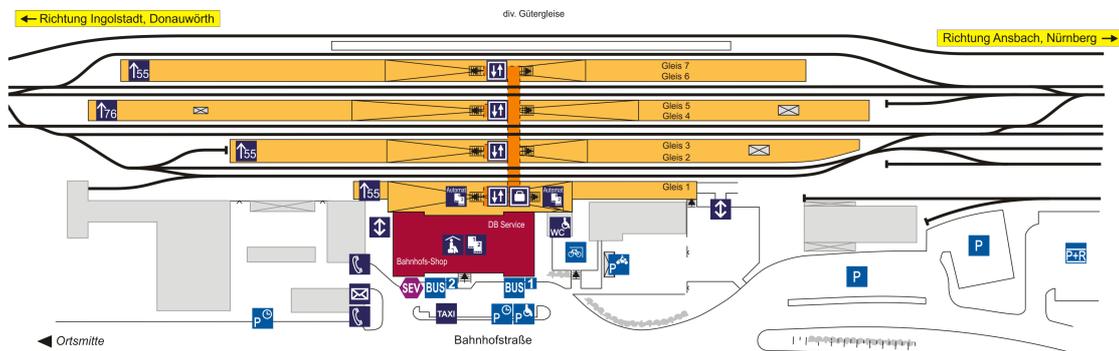


Abbildung 5-9: Lageplan Bahnhof Treuchtlingen [19]

Für die Fußgängernavigation müssen zuerst „pathLinks“ angelegt werden. Diese bezeichnen die Verbindung zwischen zwei Punkten, wobei es sich dabei um viele verschiedene Elemente innerhalb einer Haltestelle handeln kann (z.B. Lifte, Treppen, Ebenen, ...). Auch Zeitinformationen können hinterlegt werden – d.h. die Information, wie lange man für das Ablaufen dieser Verbindung benötigt. Ein „pathLink“ ist eine gerichtete Verbindung, die jedoch, wenn nicht anders definiert, in beide Richtungen benutzt werden kann. Bei Vorhandensein von „parentQuay“ und „childQuay“, das heißt, wenn sich zwei Gleise auf einem Bahnsteig befinden, wird empfohlen, nur auf den Bahnsteig zu verlinken, da andernfalls zu viele Verbindungen entstehen würden. Das passende Gleis kann durch *NeTEx* abgeleitet werden. [9]

Für den Beispieldatensatz mussten zwei Etagen angelegt werden – eine für die Unterführung, eine zweite für die Bahnsteigebene. Des Weiteren wurden die Ein-/Ausgänge der beiden Treppen definiert. „pathLinks“ werden in der Regel zwischen Eingängen definiert – wobei es sich hierbei auch um einen Treppeneingang handeln kann. Es wurden dann folgende Verbindungen angelegt:

- Plattform von Gleis 2 und 3 → oberer Treppeneingang von Gleis 2 und 3
- oberer Treppeneingang von Gleis 2 und 3 → unterer Treppeneingang von Gleis 2 und 3
- unterer Treppeneingang von Gleis 2 und 3 → unterer Treppeneingang von Gleis 4 und 5
- usw.

Um aus den einzelnen Verbindungen zwischen zwei Punkten einen Gesamtweg zu modellieren, wird das Element „navigationPaths“ benötigt. Hier werden die „pathLinks“ mit Hilfe von „pathLinkInSequence“ in eine Reihenfolge gebracht. Dabei kann auch ein Ebenen-Wechsel angegeben werden, zur Auswahl stehen nach oben („up“), nach unten („down“) oder gleichbleibend („level“). (vgl. Abbildung 5-10)

```
<navigationPaths>
  <NavigationPath version="any" id="NP01">
    <Name>Gleis 2/3 zu Gleis 4/5</Name>
    <NavigationType>quayToQuay</NavigationType>
    <pathLinksInSequence>
      <PathLinkInSequence version="any" id="PLIS01" order="1">
        <PathLinkRef ref="SPL01"/>
        <Description>Gleis 4/5 zur oberen Treppe</Description>
        <Reverse>true</Reverse>
        <Transition>level</Transition>
      </PathLinkInSequence>
      <PathLinkInSequence version="any" id="PLIS02" order="2">
        <PathLinkRef ref="SPL02"/>
        <Description>oberes Treppeneinde zu unterem Treppeneinde Gleis 4/5</Description>
        <Reverse>true</Reverse>
        <Transition>down</Transition>
      </PathLinkInSequence>
    </pathLinksInSequence>
  </NavigationPath>
</navigationPaths>
```

Abbildung 5-10: Auszug aus NeTEx-Beispiel - navigationPaths

#### - ScheduledStopPoint

Unter „ScheduledStopPoint“ versteht *NeTEx* die Beschreibung von Haltestellen aus Infrastruktursicht. Dieses Element wird also benötigt, um die Abfolge von Haltestellen innerhalb des Streckennetzes zu definieren. „ScheduledStopPoint“ entspricht auch den „VDV-Haltestellen“, das heißt ein Omnibusbahnhof hat viele Haltestellen (= Bussteige) und eine Fahrt der Linie X hält an einem dieser Punkte.“ [18]

Diesem Element werden nicht viele Attribute mitgegeben, da die meisten relevanten Informationen bereits bei „StopPlace“ hinterlegt sind (vgl. Abbildung 5-11). Es genügt, noch einmal den Namen und das Bahnhofskürzel anzugeben.

```

<scheduledStopPoints>
  <ScheduledStopPoint version="any" id="SSP01">
    <Name>Muenchen Hauptbahnhof</Name>
    <ShortName>MH</ShortName>
  </ScheduledStopPoint>

```

Abbildung 5-11: Auszug aus *NeTEx*-Beispiel - *scheduledStopPoints*

Die Verbindung zwischen „StopPlace“ und „ScheduledStopPoint“ wird mit Hilfe von „PassengerStopAssignment“ hergestellt. Dafür wird noch einmal der Name des Bahnhofs angegeben und die beiden Elemente des gleichen Bahnhofs werden referenziert (vgl. Abbildung 5-12).

```

<stopAssignments>
  <PassengerStopAssignment version="any" id="PSA01" order="1">
    <Description>Muenchen Hauptbahnhof</Description>
    <ScheduledStopPointRef ref="SSP01"/>
    <StopPlaceRef ref="SP01"/>
  </PassengerStopAssignment>

```

Abbildung 5-12: Auszug aus *NeTEx*-Beispiel - *stopAssignments*

## 5.4 Fahrpreise

### *railML*

Das Datenaustauschformat *railML* bietet nicht die Möglichkeit, Fahrpreise oder Tarifstrukturen zu beschreiben.

### *NeTEx*

Der dritte Teil von *NeTEx* sieht die detaillierte Abbildung von Tarifen und Fahrpreisen vor. Er soll später eine eigene Norm bilden, welche sich jedoch noch in der Ausarbeitung befindet und noch nicht zertifiziert ist.

Es geht dabei nicht um das einfache Hinterlegen eines Fahrpreises, sondern um die detaillierte Abbildung der gesamten Tarifstruktur und Tarifdaten. Dazu werden die Netzwerk- und Fahrplandaten der ersten beiden *NeTEx*-Teile sowie das Fahrpreisberechnungsmodell von *Transmodel V51* verwendet. [8]

Die Ziele des dritten Teils sind folgende [8]:

- Die vielen möglichen Tarifstrukturen im öffentlichen Verkehr zu beschreiben (z.B. zonenabhängige, distanzabhängige oder auch zeitbasierte Fahrpreise, Saisonpässe, etc.).

- Die Beschreibung der Tarifprodukte, die auf Grund dieser Tarifstrukturen erworben wurden, und die Definition der Konditionen, die auf bestimmte Tarife zutreffen könnten (z.B. wenn ein Ticket nur für eine bestimmte Personengruppe zugelassen ist oder an zeitliche Einschränkungen gebunden ist).
- Der Austausch der jeweils gültigen Preisinformationen soll ermöglicht werden. NeTEx selbst ist nicht dafür zuständig, Preisalgorithmen oder die Art und Weise der Preisberechnung festzulegen – dazu ist das Preismanagement-System zuständig. *NeTEx* kann dazu benutzt werden, die zahlreichen Parameter einer Preisberechnung, die zur Erklärung oder Rechtfertigung eines Fahrpreises benötigt werden, auszutauschen.
- Es sollen auch die Attribute und Textbeschreibungen, die notwendig sind, um die Fahrpreise und ihre Verkaufsbedingungen der Öffentlichkeit präsentieren, mitinbegriffen sein.

## 5.5 Weitere Herausforderungen beim Erstellen der Beispiele

### 5.5.1 Infrastruktur

#### *railML*

Die Infrastruktur unter *railML* wird mit Hilfe von „tracks“ beschrieben. Dieses Element steht für die Gleise in einem Bahnnetz und ermöglicht die detaillierte Beschreibung jedes einzelnen Gleises. Innerhalb eines „track“ wird die Gleistopologie näher beschrieben („trackTopology“), wobei der Beginn und das Ende eines Bahnabschnitts eingetragen werden. Ausgangs- und Endpunkte von Abschnitten sind dabei jeweils an Verzweigungen oder Kreuzungen anzusetzen, da dies zur vollständigen Beschreibung des Netzes vollkommen ausreicht. Alle Bahnhöfe oder andere wichtige Objekte innerhalb dieses „track“ können mit Hilfe von „crossSection“ hinzugefügt werden. All diesen Attributen kann eine relative („pos“) und eine absolute Position („absPos“) in Metern mitgegeben werden. Die relative Position bezieht sich auf den Beginn des tracks, bei der absoluten Position können die bahninternen Kilometerangaben eingetragen werden. (vgl. Abbildung 5-13)

```

<infrastructure id="is01">
  <tracks>
    <track id="tr01">
      <trackTopology>
        <trackBegin pos="0" id="trb01" absPos="0">
          <macroscopicNode ocpRef="ocp01" />
        </trackBegin>
        <trackEnd pos="80992" id="tre01" absPos="80992">
          <macroscopicNode ocpRef="ocp10" />
        </trackEnd>
        <crossSections>
          <crossSection ocpRef="ocp02" absPos="17800" pos="17800" id="crs01"/>
          <crossSection ocpRef="ocp03" absPos="36438" pos="36438" id="crs02"/>
          <crossSection ocpRef="ocp04" absPos="40182" pos="40182" id="crs03"/>
          <crossSection ocpRef="ocp05" absPos="43847" pos="43847" id="crs04"/>
          <crossSection ocpRef="ocp06" absPos="49662" pos="49662" id="crs05"/>
          <crossSection ocpRef="ocp07" absPos="60169" pos="60169" id="crs06"/>
          <crossSection ocpRef="ocp08" absPos="72913" pos="72913" id="crs07"/>
          <crossSection ocpRef="ocp09" absPos="80992" pos="80992" id="crs08"/>
        </crossSections>
      </trackTopology>
    </track>
  </tracks>
</infrastructure>

```

Abbildung 5-13: Auszug aus *railML*-Beispiel - tracks

Die Infrastruktur der Eisenbahn besitzt neben einer internen Streckenkilometrierung auch eine interne Streckenbezeichnung, welche unter „trackGroups“ aus den einzelnen Gleisen („track“) zusammengesetzt werden. Um in einem Bahnhof die verschiedenen Gleise zu modellieren, muss die Infrastruktur des Bahnhofs exakt nachgebildet werden, indem für jedes Gleis ein eigener „track“ angelegt wird.

### NeTeX

In *NeTeX* gibt es verschiedene Möglichkeiten, eine Infrastruktur anzulegen. Zum einen begründet sich dies durch die vielen verschiedenen Verkehrsmittel und Länder, die sich an der Entwicklung beteiligt haben. Dadurch ergeben sich unterschiedliche Anforderungen, die alle in einem Standard vereint werden sollen. Zum anderen orientiert sich *NeTeX* stark an Transmodel, welches sich durch einen sehr hohen Abstraktionsgrad auszeichnet [17]. Es ist sehr schwer, allein anhand des Standards die Unterschiede zwischen diesen Möglichkeiten und die Anwendungsfälle, die der Entwicklung dieser verschiedenen Möglichkeiten zu Grunde liegen, zu verstehen. In diesem Kapitel soll auch der Fernbus gesondert betrachtet werden.

*NeTeX* unterscheidet grundlegend zwischen

- dem geografischen Verlauf einer Linie und
- der Topologie im Sinne von bedienten Haltestellen /Orten. [17]

- Geografischer Verlauf einer Linie

Die Objekte „Route“ und „RouteLink“ beschreiben die Geographie, d.h. es könnte damit eine Linie auf einer Karte eingezeichnet werden (...). „RouteLink“ wird nur benötigt, wenn der Verlauf der Gleise exakt dargestellt werden soll. Die reine Abfolge von Haltepunkten würde man über „ServicePattern“ modellieren.[17]

Ein „RouteLink“ ist die Verbindung zwischen zwei Punkten. Dieser Punkt kann sowohl ein „ScheduledStopPoint“ als auch ein „RoutePoint“ sein. Ein „RoutePoint“ ist ein Punkt, der notwendig zur Beschreibung der Strecke, jedoch keine Haltestelle ist. Unter „RouteLink“ können auch Entfernungen angegeben werden. *NeTEx* ermöglicht jedoch nur die Angabe von relativen Positionen. Eine „Route“ definiert alle Haltestellen und zusätzlich angelegten Punkte „RoutePoint“ auf einer bestimmten Strecke in der richtigen Reihenfolge.

- Topologie (bediente Haltestellen)

*NeTEx* enthält weiterhin die Möglichkeit, „servicePatterns“ anzulegen. Diese beschreiben die Folge der bedienten Haltestellen („ScheduledStopPoints“). [17]

Weiterhin ist es möglich, „journeyPatterns“ anzulegen.

„servicePattern“ liefert zunächst einmal nur die Haltestellenfolge und mögliche Fahrzeitprofile (= time demand types), das heißt auf einem Fahrweg kann es mehrere unterschiedliche Fahrzeiten geben (z.B. Haupt- oder Schwachverkehrszeit). Über „journeyPatterns“ erhält man anschließend den Fahrzeitverlauf, der tatsächlich für die Fahrt verwendet wird. Dies ist für Züge eher ungeeignet, da Fahrzeiten meist vom Typ der Lok, Behängung oder ähnlichem abhängen, als von Haupt- oder Nebenverkehrszeit. [17]

- Besonderheit: Fernbus

In der bisherigen Arbeit wurden Busreisen nicht gesondert betrachtet, da sie sich in ihren Angaben nicht unterscheiden. Dies ist bei der Infrastruktur ähnlich, jedoch gibt es für den Bus kein systemeigenes Streckennetz, sondern es werden die öffentlichen Straßen genutzt. Es gibt daher eine Vielzahl an möglichen Strecken, die verwendet werden können. *NeTEx* bietet hier die Möglichkeit, bestimmte Routen vorzugeben, was mit dem geografischen Verlauf der Strecke gleichzusetzen ist. Dafür werden zusätzlich zu den angefahrenen Haltestellen, die ebenfalls als „StopPlace“ und „ScheduledStopPoint“ definiert werden, noch weitere sogenannte „routePoints“

eingefügt. Diese Punkte geben an, welche Stellen auf der Fahrt von einem Bahnhof zum nächsten passiert werden müssen. In dem Beispiel wurden zur Veranschaulichung dieser Besonderheit die Autobahnauffahrt München-Schwabing sowie die Ausfahrt Dreieck Nürnberg/Feucht als „RoutePoint“ definiert. Anschließend wurden sie mit den Haltestellen zusammen unter „Route“ zu einer definierten Strecke zusammengesetzt, wobei sie unter „PointOnRoute“ referenziert wurden.

Unter „servicePattern“ werden die zusätzlich angelegten Punkte, beispielsweise für die Auf-/Abfahrt nicht mehr referenziert. Dieser Abschnitt und auch alle anderen Informationen werden analog zum Zugverkehr angelegt.

### 5.5.2 Rollmaterial

#### *railML*

*railML* definiert unter „rollingstock“ das verwendete Rollmaterial beziehungsweise die Fahrzeuge, die zum Einsatz kommen. Hier werden zunächst einmal mit Hilfe von „vehicles“ alle benötigten Arten von Fahrzeugen angelegt, jede Lok, jeder Wagen, etc. bekommt eine eigene ID. Außerdem können dem Fahrzeug noch weitere Informationen über die Ausstattung für den Fahrgast (Anzahl Sitze, Bordbistro, Toiletten, ...) sowie auch betriebliche beziehungsweise technische Informationen zum Fahrzeug übergeben werden.

Nach Definition jedes einzelnen Fahrzeuges werden diese einzelnen Zugteile zu Formationen („formations“) zusammengesetzt, welche wiederum für den Fahrplan notwendig sind.

#### *NeTEx*

*NeTEx* geht etwas anders an die Beschreibung der Fahrzeuge heran. Hier werden nicht alle verwendeten Fahrzeugeinheiten erst definiert und dann zusammengesetzt, sondern es wird alles direkt innerhalb der Formation beschrieben.

Das Element „vehicleTypes“ ist unter „TimetableFrame“ angeordnet und erhält nicht wie in *railML* einen eigenen übergeordneten Abschnitt. Anschließend wird ein Zug („train“) angelegt, welcher verschiedene „trainComponent“ enthält. Hier wird nun nacheinander mit Angabe der Reihenfolge jeder Wagen beziehungsweise jedes Fahrzeug ausführlich beschrieben, wobei es nicht möglich ist, detaillierte technische

Details und Ausrüstungsinformationen des Zuges zu hinterlegen. *NeTEx* betrachtet die Fahrzeuge hier eher aus Fahrgastsicht.

Die ID von „train“ kann im Fahrplan referenziert werden, was bei Zugfahrten durchaus sinnvoll ist, da die Umlaufzeiten auch vom Rollmaterial abhängen.

### 5.5.3 Kupplung von Zügen

In den Beispielfahrten ist auch eine gekuppelte Zugreise enthalten. Der *ICE 732/782* fährt in München los und wird in Hannover entkoppelt. Anschließend fährt der *ICE 732* weiter nach Oldenburg (Oldb) und der *ICE 782* nach Hamburg Altona.

#### *railML*

Kupplungsvorgänge werden in *railML* mit Hilfe von „trainParts“ definiert. Zunächst einmal wird für jeden Abschnitt, den ein Zug mit der gleichen Zugnummer ohne einen solchen Vorgang zurücklegt, ein eigener „trainPart“ angelegt. In dem Beispiel müssen somit insgesamt vier Zugteile angelegt werden, zwei je Zugnummer:

- *ICE 732*: München – Hannover bzw. Hannover – Oldenburg (Oldb)
- *ICE 782*: München – Hannover bzw. Hannover – Hamburg Altona

Unter „trains“ werden die einzelnen Zugteile anschließend zu Zügen zusammengesetzt. Dabei unterscheidet *railML* zwischen

- einer kommerziellen (Fahrgast-)Sicht und
- einer betrieblichen Betrachtungsweise.

#### - Fahrgastsicht

Hier werden die Züge so definiert, wie sie der Fahrgast wahrnimmt. Dieser sieht lediglich, dass es einen *ICE 732* von München nach Oldenburg und einen *ICE 782* von München nach Hamburg gibt. Dass die beiden Züge auf einem Teilabschnitt zusammen verkehren, interessiert den Fahrgast in der Regel nicht.

Daher definiert *railML* erst einmal einen Zug vom Typ `type='commercial'` mit der Zugnummer 782, bei dem die Strecke München – Hannover und Hannover – Hamburg Altona nacheinander abgefahren werden. Analog dazu das gleiche für den *ICE 732*. (vgl. Abbildung 5-14)

```

<train type="commercial" id="trc3" trainNumber="782">
  <trainPartSequence sequence="1">
    <trainPartRef ref="tp06"/>
  </trainPartSequence>
  <trainPartSequence sequence="2">
    <trainPartRef ref="tp08"/>
  </trainPartSequence>
</train>
<train type="commercial" id="trc4" trainNumber="732">
  <trainPartSequence sequence="1">
    <trainPartRef ref="tp05"/>
  </trainPartSequence>
  <trainPartSequence sequence="2">
    <trainPartRef ref="tp07"/>
  </trainPartSequence>
</train>

```

Abbildung 5-14: Auszug aus railML-Beispiel – kommerzielle Kupplung von Zügen

#### - Betriebliche Sicht

Allein durch Beschreibung aus Fahrgastsicht ist noch nicht klar, dass die Züge, zumindest teilweise, zusammen verkehren. Dies wird in der Beschreibung aus betrieblicher Sicht definiert. Hierbei ist es notwendig zu wissen, unter welcher Zugnummer der Zug verkehrt, bis er sich trennt. Im Beispielfall ist dies die Zugnummer 782. Das heißt der Zug mit dieser Nummer besteht im ersten Abschnitt von München bis Hannover aus zwei Zugteilen, die gemeinsam auf dieser Strecke verkehren. Deshalb werden sie innerhalb der gleichen „trainPartSequence“ erwähnt, wobei hier der namensgebende Teil an der Spitze fährt und die position='1' bekommt. Der Zug mit der Nummer 732 verkehrt aus betrieblicher Sicht nur von Hannover bis Oldenburg und bildet damit den vierten und letzten Zug zur Beschreibung des Kupplungsvorgangs.

```

<train type="operational" id="tro1" trainNumber="782">
  <trainPartSequence sequence="1">
    <trainPartRef ref="tp05" position="2"/>
    <trainPartRef ref="tp06" position="1"/>
  </trainPartSequence>
  <trainPartSequence sequence="2" >
    <trainPartRef ref="tp08"/>
  </trainPartSequence>
</train>
<train type="operational" id="tro2" trainNumber="732">
  <trainPartSequence sequence="1">
    <trainPartRef ref="tp07"/>
  </trainPartSequence>
</train>

```

Abbildung 5-15: Auszug aus railML-Beispiel - betriebliche Kupplung von Zügen

### NeTEx

Die Kupplung von Zügen muss in NeTEx an mehreren Stellen definiert werden. Zunächst einmal wird unter „TimetableFrame“ das Element „journeyMeetings“ angelegt. Hier wird definiert, an welchem Bahnhof verschiedene Zugfahrten

zusammengelegt oder getrennt werden, wobei sowohl die Nummer des Bahnhofs („ScheduledStopPoint“) als die des Fahrplans („ServiceJourney“) referenziert wird. (vgl. Abbildung 5-16)

```
<journeyMeetings>
  <JourneyMeeting version="any" id="JM01">
    <AtStopPointRef ref="SSP01"/>
    <FromJourneyRef ref="SJ05"/>
    <ToJourneyRef ref="SJ06"/>
    <Reason>joining</Reason>
  </JourneyMeeting>
  <JourneyMeeting version="any" id="JM02">
    <AtStopPointRef ref="SSP48"/>
    <FromJourneyRef ref="SJ05"/>
    <ToJourneyRef ref="SJ06"/>
    <Reason>splitting</Reason>
  </JourneyMeeting>
</journeyMeetings>
```

Abbildung 5-16: Auszug aus NeTEx-Beispiel - journeyMeetings

Weiterhin müssen Kupplungsvorgänge in den betreffenden Fahrplänen eingetragen werden. Zunächst wird in der entsprechenden „ServiceJourney“ unter „parts“ die Strecke des Fahrplans in Teilabschnitte unterteilt. Bei dem Beispiel-ICE 732 von München nach Hamburg bedeutet dies, dass der erste Teil („JourneyPart“) von München nach Hannover und der zweite von Hannover nach Hamburg verläuft. (vgl. Abbildung 5-17).

```
<parts>
  <JourneyPart version="any" id="JP782_01">
    <Description>München - Hannover</Description>
    <ParentJourneyRef ref="SJ06"/>
    <MainPartRef ref="JP732_01"/>
    <FromStopPointRef ref="SSP01"/>
    <ToStopPointRef ref="SSP48"/>
    <StartTime>18:20:00</StartTime>
    <EndTime>22:40:00</EndTime>
  </JourneyPart>
  <JourneyPart version="any" id="JP782_02">
    <Description>Hannover - Hamburg</Description>
    <ParentJourneyRef ref="SJ06"/>
    <MainPartRef ref="JP782_02"/>
    <FromStopPointRef ref="SSP48"/>
    <ToStopPointRef ref="SSP54"/>
    <StartTime>22:44:00</StartTime>
    <EndTime>00:29:00</EndTime>
  </JourneyPart>
</parts>
```

Abbildung 5-17: Auszug aus NeTEx-Beispiel - JourneyPart

Ebenfalls in der entsprechenden „ServiceJourney“ wird noch in dem Bahnhof („Call“), in dem der Kupplungsvorgang stattfindet, das eingangs erwähnte „JourneyMeeting“ referenziert sowie der Grund für das Treffen eingetragen. Bei dem Beispiel-ICE werden die Züge in Hannover getrennt, weswegen bei dem „Call“ von Hannover das

„JourneyMeeting“ für das Entkuppeln sowie der Grund für das Treffen – hier Zugteilung („splitting“) – eingetragen werden. (vgl. Abbildung 5-18).

```

<Call id="Ca782_08" order="8">
  <ScheduledStopPointRef ref="SSP48"/>
  <Arrival>
    <Time>22:40:00</Time>
    <journeyMeetings>
      <JourneyMeetingView>
        <JourneyMeetingRef ref="JM02"/>
        <Reason>splitting</Reason>
      </JourneyMeetingView>
    </journeyMeetings>
  </Arrival>
  <Departure>
    <Time>22:44:00</Time>
  </Departure>
</Call>

```

Abbildung 5-18: Auszug aus NeTEx-Beispiel - journeyMeetings unter Call

## 5.6 Vergleich

Der größte Unterschied zwischen den beiden Datenaustauschformaten *railML* und *NeTEx* liegt sicherlich in ihrer Auslegung. Während es sich *NeTEx* zur Aufgabe gemacht hat, den gesamten öffentlichen Verkehr abzubilden, beschränkt sich *railML* bisher auf den Schienenverkehr.

Somit ist es natürlich nicht möglich, **Multimodalität** in *railML* abzubilden. *NeTEx* ermöglicht hingegen die Beschreibung jedes Verkehrsträgers, wobei sich in dieser Arbeit gezeigt hat, dass es für Flugreisen noch eines Beispiels bedarf, um diese umsetzen zu können.

Betrachtet man die **Umstiege** zwischen unterschiedlichen Verkehrsträgern, ist *railML* wieder im Nachteil, jedoch lässt sich sehr gut der Wechsel zwischen Zügen realisieren. Hierfür werden die Umstiege direkt im Bahnhof des jeweiligen Fahrplans definiert.

*NeTEx* unterscheidet bei Umstiegen zwischen drei Möglichkeiten. Es können einerseits Umstiege zwischen zwei bestimmten Fahrten beschrieben werden, ebenso Umstiege zwischen zwei Fahrtmustern. Es ist jedoch auch möglich, nur darüber zu informieren, welche weiteren Verkehrsmittel an einer Haltestelle verfügbar sind.

Der **Fahrplan** ist in beiden Datenaustauschformaten ähnlich aufgebaut, denn es gibt einen Abschnitt, der alle Haltestellen mit einer Ankunfts- und einer Abfahrtszeit versieht.

Die Definition der Gültigkeiten unterscheidet sich jedoch, denn in *NeTEx* gibt es zwei Möglichkeiten diese zu hinterlegen. Die erste besteht darin, dass alle Gültigkeiten zentral angelegt und später im Fahrplan nur referenziert werden. Die zweite Variante ist es, die verschiedenen Tagestypen erst direkt im jeweiligen Fahrplan aufzurufen und somit die Gültigkeiten erst in diesem Abschnitt zusammenzusetzen.

*RailML* ist da eindeutiger, hier wird zuerst festgelegt, welche Tage Feiertage sind und anschließend werden die Betriebstage festgelegt. Dabei werden wie in der ersten Variante bei *NeTEx* die Gültigkeiten zentral angelegt und später im Fahrplan aufgerufen.

Bei **Bahnhöfen** gibt es große Unterschiede in der Detailtiefe bezüglich der fahrgastrelevanten Informationen.

*NeTEx* besitzt zwei verschiedene Möglichkeiten, Bahnhöfe zu beschreiben. Zum einen ist die Definition einer Haltestelle, wie sie der Fahrgast wahrnimmt möglich, zum anderen gibt es ein Element, das zur Beschreibung der Infrastruktur benötigt wird. Beide Arten können miteinander verknüpft werden. Besonders hervorzuheben ist bei *NeTEx* die Möglichkeit, alle Objekte und Elemente innerhalb einer Haltestelle zu beschreiben.

Bei *railML* gibt es einen Bahnhof, der auch einige Informationen enthalten kann, jedoch nicht die fahrgastrelevanten Informationen von *NeTEx*, wie zum Beispiel Informationen zu Bahnsteigen, Liften oder Eingängen.

Die Beschreibung von **Fußwegen** ist nur in *NeTEx* möglich. Die Grundlage hierfür bildet die detaillierte Beschreibung der festen Objekte einer Haltestelle.

Ebenfalls in *railML* nicht enthalten ist die Möglichkeit, **Fahrpreise** anzugeben. Bei *NeTEx* befindet sich dieser Teil zur Zeit noch in der Entwicklung. Es wird jedoch damit möglich sein, Fahrpreise und komplexe Tarifstrukturen zu beschreiben.

In der **Infrastruktur** besitzen die beiden Datenaustauschformate eine unterschiedliche Herangehensweise. *RailML* definiert zwei Punkte auf der Strecke und gibt die Entfernung des Endpunktes zum Anfang an. Es ist möglich, hier eine zusätzliche Ki-

lometrierung hinzuzufügen. Zwischen dem Anfangs- und Endpunkt können anschließend noch weitere Punkte auf dieser Strecke eingetragen werden. Dies können zum Beispiel Bahnhöfe oder Haltetafeln sein, wobei ebenfalls immer die Distanz zum Anfangspunkt angegeben wird.

*NeTeX* unterscheidet bei der Infrastruktur zwei Varianten: die Beschreibung des Verlaufs der Strecke und die Topologie der bedienten Haltestellen. Das Hinterlegen von Distanzen ist hierbei in der Verlaufsbeschreibung möglich, wobei hier immer eine Verbindung zwischen zwei Punkten angelegt und ihre Länge definiert wird.

Beim **Rollmaterial** haben die beiden Formate eine komplementäre Herangehensweisen. In *railML* werden zuerst einzelne Fahrzeuge angelegt und diese anschließend zu einem Zug zusammengesetzt.

In *NeTeX* wird ein Zug definiert und innerhalb dieses Zuges werden die einzelnen Fahrzeuge beschrieben.

Bei der Beschreibung der Fahrzeuge besitzt *railML* eine wesentlich größere Detailtiefe. Hier können technische Details, beispielsweise über die Achsen oder die Funkausstattung hinterlegt werden. *NeTeX* beschränkt sich hier auf weniger Details, zum Beispiel die Energieversorgung.

Der letzte in dieser Arbeit betrachtete Abschnitt war das **Kuppeln von Zügen**. Beide Datenaustauschformate können diesen Spezialfall abbilden, wobei *railML* hier klarer vorgeht als *NeTeX*. In *railML* werden die Teile der Fahrt, in denen sich nichts ändert als einzelner Fahrplanteil angelegt. Anschließend werden diese Züge jeweils einmal aus Sicht des Fahrgastes und dann aus betrieblicher Sicht zusammengesetzt. Es entstehen dadurch vier Züge.

*NeTeX* definiert das Kuppeln in mehreren Etappen. Zuerst wird ein Treffen angelegt, bei dem angegeben wird, an welchem Bahnhof bestimmte Fahrten zusammengeslossen beziehungsweise getrennt werden. Anschließend wird die Fahrt im jeweiligen Fahrplan in Abschnitte unterteilt. Ähnlich wie bei *railML* werden hierfür die Strecken ausgewählt, auf denen sich die Zusammensetzung der Züge nicht ändert. Als letzten Schritt muss im Fahrplan im jeweiligen Bahnhof das anfangs definierte Treffen referenziert werden. Erst dann ist eine Kupplung bzw. Trennung in *NeTeX* vollständig beschrieben.

Im Anhang A.2 ist eine Zusammenstellung dieser Informationen sowie jeweils auch den dazu passenden Stellen im Code zu finden.

## 6 Erweiterungsvorschläge

Die beiden in dieser Arbeit behandelten Datenaustauschformate haben eine sehr unterschiedliche Struktur. Während *railML* im Wesentlichen drei große Teilschemen enthält, besitzt *NeTEx* eine Vielzahl unterschiedlicher Frames. Diese Frames müssen nicht zwingend verwendet werden – das ist jeweils abhängig vom verwendeten Verkehrsträger sowie auch den Vorgehensweisen in unterschiedlichen Ländern beziehungsweise Verkehrsunternehmen und der gewünschten Detailtiefe.

Diese unterschiedliche Struktur erschwert es, ein Format sinnvoll mit dem jeweils anderen zu erweitern. Die Möglichkeiten, die sich bei der Erarbeitung der Beispieldateien ergeben haben, sind in Tabelle 6-1 kurz dargestellt und werden in den beiden Kapiteln näher erläutert

Tabelle 6-1: Zusammenfassung möglicher Erweiterungen

	mögliche Erweiterung	Begründung	Abschnitt im Code	
			<i>railML</i>	<i>NeTEx</i>
<i>railML</i>	Informationen von „StopPlace“ übernehmen	ermöglicht das Hinterlegen fahrgastrelevanter Informationen an Haltestellen	<ocp>	<StopPlace>
	Beschreibung der Fußgängerwege bzw. Fußgängernavigation	ermöglicht Beschreibung von Wegen und Navigation innerhalb eines Bahnhofs (nur möglich, wenn Informationen von „StopPlace“ übernommen wurden)	<ocp>	<pathLinks> <NavigationPath>  unter: <StopPlace>
	Hinzufügen von Attributen zur Zuordnung zu bestimmtem Verkehrsmittel	Umstiege zwischen Zügen lassen sich in <i>railML</i> zwar beschreiben, es fehlt jedoch die Möglichkeit, andere Verkehrsträger zu definieren	<ocp> <trainPart>	<StopPlace> <ServiceJourney>
<i>NeTEx</i>	technische Informationen zu den Fahrzeugen	<i>NeTEx</i> enthält nur wenige Möglichkeiten, technische Details zu den Fahrzeugen zu hinterlegen	<engine> und <vehicleBrakes>  unter: <vehicle>	<TrainComponent>  unter: <vehicleTypes> <train>

## 6.1 Erweiterungsvorschläge für railML®

Ein großer Vorteil von *NeTEx* gegenüber *railML* ist die detaillierte Beschreibung von Bahnhöfen beziehungsweise Haltestellen. In *NeTEx* lässt sich nahezu jedes Element oder Objekt eines Bahnhofs definieren.

Da beide Datenaustauschformate ein eigenes Element für Bahnhöfe beziehungsweise Haltestellen besitzen, würden sich diese Definitionen recht einfach übernehmen lassen.

Es wäre damit beispielsweise möglich, die Bahnsteige und Einstiegspositionen auf den Plattformen zu beschreiben, genauso wie Eingänge zu Bahnhöfen oder auch die Lifte und Treppen. Dadurch wiederum wäre es möglich, auch den Teil von *NeTEx*, der die Beschreibung von Fußwegen innerhalb einer Haltestelle beinhaltet zu übernehmen. Dieser ist ebenfalls unter „StopPlace“ eingeordnet. Anschließend wäre mit Hilfe von „NavigationPath“ eine Navigation durch den Bahnhof möglich.

Ein weiterer allgemeiner Vorschlag wäre, Elementen von *railML*, die unverändert auch zur Beschreibung anderer Verkehrsträger genutzt werden könnten, näher bestimmende Attribute zuzuordnen. So ist es in *NeTEx* beispielsweise möglich, zu definieren, um welche Art von Plattform es sich handelt – um einen Bahnsteig, einen Bussteig oder gar eine Fähranlegestelle. Ebenso kann einer Haltestelle ein „StopPlaceType“ zugewiesen werden, der aussagt, um welche Art von Haltestelle es sich handelt. Die eben beschriebenen Beispiele sind in *NeTEx* alle unter „StopPlace“ eingeordnet. Jedoch gibt es zum Beispiel auch die Möglichkeit einem Fahrplan („ServiceJourney“) ein Verkehrsmittel zuzuordnen, indem mit Hilfe von „TransportMode“ der entsprechende Verkehrsträger ausgewählt wird (z.B. Metro, Bus, Taxi, etc.).

Durch diese Zuordnung in *railML* wäre es eventuell möglich, auch andere Verkehrsmittel zu beschreiben und beispielsweise auch den Umstieg zwischen ihnen zu definieren. Eine Software könnte diese Informationen anschließend eindeutig zuordnen und auswerten.

## 6.2 Erweiterungsvorschläge für NeTeX

Im direkten Vergleich zu *railML* fällt auf, dass *NeTeX* an einigen Stellen nicht die gleichen Möglichkeiten bietet. Dies ist zum Beispiel bei der Beschreibung der Infrastruktur aufgefallen, denn hier konnten bahnspezifische Anforderungen, wie das Hinterlegen der eigenen Streckenkilometrierung und -bezeichnung nicht erfüllt werden. Speziell bei der Infrastruktur ist es jedoch schwierig, passende Erweiterungen zu finden, da die beiden Formate hier grundlegend verschieden aufgebaut sind. Beispielsweise bei der Kilometrierung: *NeTeX* definiert die Strecke immer als Verbindung zwischen zwei Punkten mit einer bestimmten Länge. *RailML* beschreibt die Gleise („tracks“) zwar auch als Verbindung zwischen zwei Punkten, es ist jedoch noch möglich weitere Punkte dazwischen aufzuzählen. Die Distanzen werden hier relativ zum Beginn des „track“ angegeben und alle dazwischen eingefügten Punkte beziehen sich in ihrer Längenangabe ebenfalls zu diesem Anfang.

Ein weiterer Punkt in dem *NeTeX* einige betriebliche Informationen außer Acht lässt ist der Abschnitt zum Rollmaterial. Hier ist es nicht möglich, bestimmte technische Informationen zu den Fahrzeugen, wie zum Beispiel Informationen zu Achsen oder der Funkausstattung, zu hinterlegen. Daher wäre es sinnvoll und durchaus möglich, die Beschreibung von *railML* zu übernehmen. Die Vorgehensweise bei der Zusammensetzung von Fahrzeugen ist zwar komplementär, jedoch haben beide Formate die Beschreibung des einzelnen Fahrzeugteils gemein – nur jeweils an unterschiedlichen Stellen. Es wäre dadurch möglich, Elemente wie „engine“ und „vehicleBrakes“ aus *railML* zu übernehmen und sie unter „TrainComponent“ in *NeTeX* anzuordnen.

## 7 Zusammenfassung

Ziel dieser Arbeit war es, zwei Datenaustauschformate auf XML-Basis zu analysieren, miteinander zu vergleichen und mögliche gegenseitige Erweiterungen vorzuschlagen.

Die beiden Austauschformate waren:

- *railML*®, welches bisher ausschließlich den Schienenverkehr betrachtet
- *NeTEx*, welches den gesamten öffentlichen Verkehr abdeckt, von der Eisenbahn, über Bus und Straßenbahn, bis hin zu Flügen und Fährreisen

Dabei sollte vor allem auf die Fähigkeit, intermodale Fahrgastinformationen bereitzustellen, eingegangen werden.

Dazu wurden Beispielfahrten auf der Strecke von München nach Nürnberg ausgewählt, die sich vor allem durch verschiedene Verkehrsmittel (Regional- und Fernverkehrszüge, Fernbus, Flug) kennzeichnen. Bei den Zugfahrten wurden zusätzlich unterschiedliche Fahrwege und einige betriebliche Besonderheiten (Kuppeln von Zügen) ausgewählt.

Zur besseren Bearbeitung und als Leitfaden für diese Arbeit wurden vier Fragen entwickelt, die sich ein Fahrgast vor Antritt seiner Reise stellen sollte:

- Welche Verkehrsmittel stehen zur Auswahl?
- Wie genau komme ich von A nach B?
- Wo muss ich hin?
- Wie hoch ist der Preis?

Während der Bearbeitung sollten die beiden Datenaustauschformate auf ihre Eignung überprüft werden, die Informationen, welche zur Beantwortung dieser Fragen notwendig sind, bereitzustellen.

Aufgrund der Auslegung von *railML*® für den Schienenverkehr wäre eine multimodale Beschreibung nicht interoperabel. Eine Software, die Daten mit Hilfe von *railML*®

auslesen soll, könnte diese Informationen nicht verarbeiten, da nicht eindeutig wäre, welcher Verkehrsträger beschrieben werden soll. Mit *NeTEx* hingegen lassen sich die meisten Verkehrsmittel gut beschreiben. In dieser Arbeit hat sich die Darstellung von Flugreisen jedoch als schwierig herausgestellt.

Intermodalität ist mit *railML*® ebenfalls nicht umsetzbar, jedoch können sehr gut Umstiege zwischen Zugfahrten beschrieben werden. *NeTEx* ermöglicht das Definieren von intermodalen Umsteigebeziehungen in verschiedenen Detailtiefen. Es können Umstiege zwischen zwei bestimmten Fahrten sowie zwischen Fahrtmustern beschrieben werden. Außerdem kann noch die Information, in welche Verkehrsmittel ein Umstieg an einer Haltestelle möglich ist, hinterlegt werden.

Bahnhöfe lassen sich mit *NeTEx* äußerst genau beschreiben, vor allem die festen Objekte und Elemente, die ein Fahrgast benutzt. *RailML*® beschreibt vorrangig betriebliche Details.

Fußwege zwischen den Elementen eines Bahnhofs lassen sich nur mit *NeTEx* beschreiben. Ebenso verhält es sich mit dem Hinterlegen von Fahrpreisen. Hierfür soll es bald einen neuen Standard von *NeTEx* geben, der auch komplexe Tarifstrukturen abbilden kann.

Bei der Infrastruktur ist *railML*® klarer definiert. *NeTEx* unterscheidet zwischen einem geografischen Verlauf der Strecke und einer topologischen Reihenfolge der bedienten Haltestellen. Es ist hierbei jedoch nicht möglich, spezielle Streckenkilometrierungen und -bezeichnungen zu hinterlegen, was *railML*® jedoch beherrscht.

Um die verwendeten Fahrzeuge zu beschreiben gehen beide Formate komplementär vor. *RailML*® legt zuerst einzelne Zugteile an und setzt sie anschließend zu einem gesamten Zug zusammen, wohingegen *NeTEx* erst den Zug definiert und dann direkt darin die einzelnen Teile beschreibt.

Zuletzt wurde das Kuppeln von Zügen betrachtet, wobei festgestellt wurde, dass beide Datenaustauschformate diesen speziellen Fall beschreiben können, jedoch wieder unterschiedlich an die Definition herangehen.

Zusammenfassend ist festzustellen, dass *railML*® oft eher die betriebliche Sichtweise wählt, wohingegen *NeTEx* viele Möglichkeiten der Beschreibung aus Fahrgastsicht bietet.

Es hat sich gezeigt, dass eine Erweiterung der beiden Formate miteinander aufgrund der unterschiedlichen Strukturen schwierig ist.

Es wäre möglich, *railML*® um die detaillierte Beschreibung von Bahnhöfen und darauf aufbauend auch um die Beschreibung von Fußwegen innerhalb einer Haltestelle zu ergänzen. Im Gegenzug würde sich eine Übernahme der Beschreibung von Fahrzeugen von *railML*® zu *NeTEx* anbieten, um Züge technisch genauer definieren zu können.

## 8 Ausblick

Wie bereits in der Einleitung erwähnt, findet *NeTEx* bisher keine Anwendung. Dies ist sicherlich darauf zurückzuführen, dass es sich hierbei um einen recht jungen Standard handelt. Andererseits könnte es jedoch auch zur Verbreitung der Norm beitragen, wenn der etwas unübersichtliche und lange Standard um einige Beispiele ergänzt würde. Es wäre daher durchaus sinnvoll, ein Handbuch zu entwickeln, welches anhand von verschiedenen Szenarien die vielen Anwendungsmöglichkeiten von *NeTEx* aufzeigt. Damit könnte man beispielsweise aufzeigen, welche Möglichkeit zur Beschreibung der Infrastruktur wann verwendet werden sollte. Auch die Beschreibung einer Flugreise bedarf einem Beispiel, da sie zwar im Standard erwähnt wird, jedoch keine näheren Erläuterungen dazu vorhanden sind. Für den Fernverkehr sind Flüge durchaus von Interesse, da beispielsweise in einem Verkehrsmittelvergleich über weite Strecken hinweg der Flug eine entscheidende Rolle spielen kann.

Es wäre auch denkbar, in weiteren Arbeiten andere Verkehrsträger genauer zu betrachten, vor allem im innerstädtischen Verkehr. Hier wäre eventuell auch ein Vergleich mit der *VDV-Schrift 452* angebracht, um zu zeigen, welche Vorteile *NeTEx* gegenüber dieser, vor allem im deutschsprachigen Raum weitverbreiteten, Schnittstelle bietet.

Bei *railML* ist zu prüfen, inwieweit sich die hier getroffenen Vorschläge für eine Umsetzung eignen und wie viel Aufwand dafür notwendig ist. Wenn sich die Vorschläge als realisierbar erweisen, ist eine Integration in *railML*® 3.0 denkbar.

## 9 Literaturverzeichnis

- [1] WHITLEY, D., "Developing and evolving RTPI strategies in an era of cost-cutting," *European Railway Review*, vol. 20, no. 6, pp. 28–30, 2014.
- [2] VERBAND DEUTSCHER VERKEHRSUNTERNEHMEN (VDV), "VDV-Schrift 452: VDV-Standardschnittstelle Liniennetz/Fahrplan," *V1.5*, pp. 9–12, 07.2013.
- [3] MENTZ DATENVERARBEITUNG GMBH; BLIC GMBH, "Entwicklung und Validierung eines Standards für Daten für den Öffentlichen Verkehr," 01.2014.
- [4] "Soll-Daten-Schnittstellen: europäische Norm NeTEx (CEN)." [Online]. <https://www.vdv.de/netex.aspx>. [Abgerufen am: 15.02.2015].
- [5] MORGENSTERN, P., "Stresstest aus der Schweiz für Stuttgart 21 plus," *VDI Nachrichten*, vol. 48, Düsseldorf, p. 4, 03.12.2010.
- [6] CEN/TC 278, "DIN EN 12896:2008-02: Straßentransport- und Verkehrstelematik – Öffentlicher Verkehr – Referenzdatenmodell; Englische Fassung EN 12896:2006," 2008.
- [7] CEN/TC 278, "Road traffic and transport telematics - Public transport - Identification of fixed objects in public transport," 2007.
- [8] CEN/TC 278, "Public transport — Network and Timetable Exchange ( NeTEx ) — Part 3: Public transport fares exchange format," 2013.
- [9] CEN/TC 278, "DIN CEN/TS 16614-1 - NeTEx - Teil 1: Austauschformat für Netzwerk-Topologie im öffentlichen Verkehr," 2014.
- [10] "railML.org - Vorstellung." [Online]. <http://www.railml.org//index.php/vorstellung.html>. [Abgerufen am: 15.02.2015].

- [11] "railML-Partner: Anwender." [Online].  
<http://www.railml.org//index.php/anwender.html>. [Abgerufen am: 15.02.2015].
- [12] WUNSCH, S., "Expertengespräch railML," 2015.
- [13] DB REGIO AG, "Streckenkarte Bayern," p. 80634, 2011.
- [14] GOOGLE, "Google Maps," 2009. .
- [15] UNITED NATIONS, "Terminologie des kombinierten Verkehrs," New York, Geneva, 2001.
- [16] INSTITUT FÜR REGIONAL- UND FERNVERKEHRSPLANUNG, "Allgemeines zu Datumsbezügen , Gültigkeitsperiode und Feiertagen Abbildung spezieller Verkehrstagerregelungen ohne Datumsbezug," pp. 16–19, 2012.
- [17] DÜX, W., "E-Mail vom 11. Februar 2015 - Beantwortung von Fragen," *Mentz Datenverarbeitung GmbH*, München, 11.02.2015.
- [18] DÜX, W., "E-Mail vom 2. Februar 2015 - Beantwortung von Fragen," *Mentz Datenverarbeitung GmbH*, München, 02.02.2015.
- [19] STATIONS DATEN BANK BAYERN, "Stationssteckbrief Treuchtlingen," 2015. [Online].  
<http://www.stationsdatenbank.bayern-takt.de/StationsdatenbankBEG/Steckbrief.html?lang=de&efz=8000122>.  
[Abgerufen am: 26.02.2015].

# 10 Anhang

## A.1 Ausgewählte Verkehrsmittel auf der Beispielstrecke

Verkehrsmittel	Fahrtnummer	Abfahrt		Ankunft		Fahrzeit [h]		Fahrpreis (regulär / spontan)	Bemerkung	
		Ort	Gleis	Zeit	Ort	Gleis	Zeit			
Eisenbahn	RB 59104	München Hbf	20	17:29	Nürnberg Hbf	12	20:17	<u>35,40 €</u>	langsamer direkter Regionalverkehr	
	RE 57316	München Hbf	17	17:35	Treuchtlingen	4	19:20		Nahverkehr mit Umstieg	
	RB 59104	Treuchtlingen	3	19:25	Nürnberg Hbf	12	20:17	<u>35,40 €</u>		
							gesamt:	<u>2:37</u>		
Flugzeug	ICE 732/782	München Hbf	19	18:20	Nürnberg Hbf	7	19:31	<u>55,00 €</u>	gekuppelter ICE über Schnellfahrstrecke	
	RE 4042	München Hbf	26	19:08	Nürnberg Hbf	11	20:48	<u>35,40 €</u>	schneller RE über Ingolstadt	
	S8	München Hbf (Terminal)		17:35	München Flughafen Terminal		18:15	10,80 €	Direktflug (inklusive Zubringer und Wartezeiten)	
	LH 2162	MUC		18:55	NUE		19:30	82,49 €		
	U2	Nürnberg Flughafen		19:52	Nürnberg Hbf (U-Bahn)		20:05	2,50 €		
							gesamt:	<u>2:05</u>	<u>95,79 €</u>	
Bus	Mein Fernbus N05	München ZOB		21:30	Nürnberg ZOB		23:40	<u>19,50€</u>	Fernbus	

abgerufen am 09.12.14

Stichtag: 16.12.14

## A.2 Tabellarische Zusammenfassung der Analyse

		<i>railML</i>	<i>NeTEx</i>
Multi-modalität	Beschreibung	<ul style="list-style-type: none"> <li>- Auslegung für Schienenverkehr</li> <li>- eine multimodale Verwendung wäre nicht interoperabel</li> </ul>	<ul style="list-style-type: none"> <li>- Beschreibung vieler verschiedener Verkehrsträger möglich (Zug, Bus, Straßenbahn, Flugzeug, Fähre, ...)</li> </ul>
Umstiege	Beschreibung	<ul style="list-style-type: none"> <li>- Umstieg zwischen Zügen</li> </ul>	<ul style="list-style-type: none"> <li>- Umstieg zwischen jeglichen Verkehrsträgern</li> </ul>
	Definition im Code unter	timetable: <trainPart> <ocpTTs> <ocpTT> <connections>	ServiceFrame: <connections>  TimetableFrame: <ServiceJourneyInterchange>
Fahrplan	Beschreibung	<ul style="list-style-type: none"> <li>- Möglichkeit, spezielle Gültigkeiten für Güterzüge zu beschreiben (Versatz zu einem Feiertag)</li> </ul>	<ul style="list-style-type: none"> <li>- mehrere Möglichkeiten:               <ul style="list-style-type: none"> <li>• Definition aller Bedingungen in „contentValidityCondition“ oder</li> <li>• im jeweiligen Fahrplan</li> </ul> </li> </ul>
	Definition Gültigkeiten im Code unter	timetable: <timetablePeriod> <operatingPeriod>	ServiceCalendarFrame: <DayType>  TimetableFrame: <contentValidityCondition> <AvailabilityCondition>
	Definition Fahrplan im Code unter	timetable: <trainPart>  anschließend Zusammensetzen als <train>	TimetableFrame: <ServiceJourney>
Bahnhof	Beschreibung	<ul style="list-style-type: none"> <li>- wenige Möglichkeiten, fahrgastrelevante Informationen zu hinterlegen</li> </ul>	<ul style="list-style-type: none"> <li>- zwei Arten von Bahnhöfen:               <ul style="list-style-type: none"> <li>• aus Sicht des Fahrgastes</li> <li>• zur Beschreibung der Infrastruktur</li> </ul> </li> <li>- Definition einer Vielzahl von Objekten und Elementen an Haltestelle, die ein Fahrgast nutzen kann</li> </ul>
	Definition im Code unter	infrastructure: <ocp>	unter SiteFrame: <StopPlace>  unter ServiceFrame: <ScheduledStopPoint> <PassengerStopAssignment>

Fußwege	Beschreibung	Beschreibung nicht möglich	Definition von Fußwegen zwischen den Objekten, die unter „StopPlace“ angelegt wurden, möglich
	Definition im Code unter	-----	SiteFrame: <StopPlace> <PathLink> <NavigationPath>
Fahrpreise	Beschreibung	Beschreibung nicht möglich	- möglich, jedoch noch in der Entwicklung - Möglichkeit, komplexe Tarifstrukturen abzubilden
Infrastruktur	Beschreibung	- Hinterlegen betrieblicher Informationen, wie bahninterne Streckenkilometrierung und -bezeichnung möglich	- zwei unterschiedliche Varianten: • geografischer Verlauf einer Strecke • Topologie der bedienten Haltestellen
	Definition im Code unter	infrastructure: <track> <trackTopology>	ServiceFrame: - geografischer Verlauf: <routePoints> <routeLinks> <routes> - topologischer Verlauf <servicePatterns> <journeyPatterns>
Rollmaterial	Beschreibung	- erst Definition des einzelnen Zugteile - anschließend Zusammensetzung zu einem kompletten Fahrzeug - detaillierte technische Beschreibung der Fahrzeuge möglich	- Anlegen eines Fahrzeuges - nacheinander Beschreibung der einzelnen Zugteile innerhalb des Gesamtzuges - keine technische Details definierbar
	Definition im Code unter	unter rollingstock: <vehicle> <formation>	unter TimetableFrame <Train> <trainComponents>
Kupplung von Zügen	Beschreibung	- Anlegen einzelner Fahrplanteile - Zusammensetzung einzelner Teile zu gesamten Fahrplan bzw. Zug - Betrachtung einmal aus betrieblicher, einmal aus Fahrgastsicht je Zugnummer —> 4 Züge	- Definition, wo sich zwei Fahrten zu welchem Zweck treffen, außerhalb des Fahrplans (Referenz auf Fahrplan und Haltestelle) - Eintrag direkt im jeweiligen Fahrplan - Unterteilung der Fahrt in die Abschnitte, in denen sich nichts ändert - Eintrag des Treffens in der jeweiligen Haltestelle im Fahrplan
	Definition im Code unter	timetable: <trainPart> anschließend Zusammensetzen als <train>	TimetableFrame <JourneyMeeting> <ServiceJourney> <parts> <Call>

## **Eidesstattliche Erklärung**

Hiermit erkläre ich, dass ich die von mir am heutigen Tage eingereichte Studienarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Jessica Pöpping

Dresden, den 3. März 2015