

Stadler Dok.-ID PA_2272526
Status In Bearbeitung
Index 1.3
Vom 29.06.2020
Sprache De
Anzahl Seiten 17

Spezifikation RailML Schnittstelle zur FIS Grunddatenversorgung

Spezifikation der Datenstrukturen auf
Basis von RailML

Revisionen

Index	Datum	Änderung	Erstellt	Geprüft (STAP)	Freigegeben (STAP)
1.0	18.12.19	Erstausgabe	Kabisch	Frank	Zander
1.1	10.03.20	Umläufe und Dateikonventionen	Kabisch	Frank	Zander
1.2	27.03.20	Ansagenverfeinerung, Linien, Ziele	Kabisch	Frank	
1.3	29.06.20	Ansagentypen, Ansagereferenzen, Version	Kabisch	Frank	

Inhalt

1	Einleitung	4
1.1	Abgrenzung	4
1.2	Zweck und Geltungsbereich	4
1.3	Zielgruppe	4
1.4	Aufbau des Dokumentes	4
2	Technologie	5
2.1	Kommunikationsmechanismus	5
2.2	Validierung	5
2.3	Versionshandling	5
2.4	Datentypen	6
3	Beschreibung zu verwendeten RailML-Struktur	7
3.1	Allgemeine neudefinierte Elemente	7
3.1.1	Sichtbarkeitsbereiche (<stap:scopes>)	7
3.1.2	Anzeigeinhalte (<stap:designator>)	8
3.1.3	Ansagen (<stap:announcement>).....	9
3.2	Subschema Infrastructure (<infrastructure>/).....	10
3.2.1	Allgemeine Infrastrukturangaben	10
3.2.2	Infrastrukturmetainformationen (<metadata>).....	10
3.2.3	Stationsstammdaten (<infrastructure>/<operationControlPoints>/<ocp>)	10
3.3	Subschema Rolling Stock (<rollingstock>).....	12
3.3.1	Fahrzeuge (<vehicles>/<vehicle>)	12
3.3.2	Fahrzeugverbände (<formations>/<formation>).....	12
3.4	Subschema Timetable (<timetable>)	13
3.4.1	Fahrplangültigkeiten (<timetablePeriods>/<timetablePeriod>)	13
3.4.2	Verkehrstage (<operatingPeriods>/<operatingPeriod>)	13
3.4.3	Zuggattungen (<categories>/<category>)	13
3.4.4	Zugteile (<trainParts>/<trainPart>/).....	13
3.4.5	Züge (<trains>/<train>)	15
3.4.6	Umlaufpläne (<rosterings>/<rostering>).....	15
4	Dateiaustausch-Struktur	16
4.1	Textdateien (/txt).....	16
4.2	Webseiten (/html)	16
4.3	Bilddateien (/pic).....	16
4.4	Audiodateien (/audio).....	16
4.5	Videodateien (/video).....	17
5	Dokumentenverweise	17

Abkürzungen

STAP	Stadler Pankow GmbH
STAG	Stadler Bussnang AG
STAR	Stadler Altenrhein AG
STARe	Stadler Reinickendorf GmbH
AN	Auftragnehmer
AG	Auftraggeber
FIS	Fahrgastinformationssystem
RBL	Rechnerbetriebsleitsystem
ITCS	Intermodal Transport Control System <small>in Deutschland oft noch RBL genannt</small>
GPS	Global Positioning System
HTTPS	HyperText Transfer Protocol Secure
URL	Uniform Resource Locator
EVU	Eisenbahnverkehrsunternehmen
TF	Triebfahrzeugführer
ELA	Elektronische Lautsprecheranlage
DFIWeb	Dynamische Fahrgastinformation Webtechnologie
STADD	Stadler Dateidrehscheibe

1 Einleitung

1.1 Abgrenzung

Die Datenversorgung der FIS-Kernapplikation und der Landseite erfolgt grundsätzlich über zwei Kommunikationskanäle: i) der Grunddatenversorgung mit Hilfe auf Basis eines erweiterten RailML-Standards gemäß RailML-Spezifikation 2.4 und ii) der online-Datenversorgung über JSON-Telegramme (vgl. [2].)

Diese Spezifikation deckt den Grunddatenteil ab.

1.2 Zweck und Geltungsbereich

Das nachfolgende Dokument beschreibt die Fileschnittstelle zur Datenversorgung des Steuerrechners der Fahrgastinformation auf dem Fahrzeug.

1.3 Zielgruppe

Dieses Dokument ist für folgende Zielgruppe vorgesehen

- Kundenvertreter verantwortlich für das Projekt SBH und folgende.
- Stadler Mitarbeiter, welche am Design und an der Ausführung des Projektes 'FIS' beteiligt sind.
- Mitarbeiter von Zulieferfirmen die mit der Implementierung dieser Schnittstelle betraut sind

1.4 Aufbau des Dokumentes

Die Beschreibung der Datenschnittstelle besteht aus folgenden Punkten:

- Einleitung
- Grundsätzliches zur verwendeten Technologie
- Beschreibung aller verwendeter Strukturen
- Beispiele

2 Technologie

Kern des Grunddatenaustausches zwischen landseitigem RBL und der FIS-Kernapplikation auf dem Fahrzeug bildet die in dieser Spezifikation beschriebene XML-Struktur. Diese basiert auf der RailML-Spezifikation 2.4 [1] welche um FIS-spezifische Besonderheiten erweitert ist.

2.1 Kommunikationsmechanismus

Die Übertragung der FIS Grunddaten erfolgt als ein gepacktes Archiv, bestehend aus der hier beschriebenen XML-Datei und zusätzlichem externen Content (beispielsweise Audiodateien für die Bahnhofsansagen). Beide Bestandteile des Datenpakets werden verpackt, zusätzlich wird eine md5-Checksummendatei mit übergeben, um eine Verifizierung des Dateiinhalts zu ermöglichen.

Grundsätzlich ist aufgrund des Stadler-Security-Konzepts ein Kommunikationsaufbau nur vom Fahrzeug aus vorgesehen. Zur Sicherstellung dieses Zieles wird eine Client-Server-Architektur verwendet, in der landseitig ein FTPS-Server zwischengeschaltet ist.

Der Zugriff auf diesen Server erfolgt für landseitige Applikationen (hier RBL) über ein von Stadler zu definierendes FTPS Nutzer- und Rollenkonzept, wobei die aktive Komponente die Applikation ist. Die Adressierung der Zielfahrzeuge erfolgt hierbei über die Verzeichnisstruktur auf dem FTPS-Server. Es können sowohl einzelne Fahrzeuge adressiert werden, oder aber ein gesamte Flotte in einem einzigen Transfer.

Eine Stadler-Komponente sorgt für die Übertragung auf die Fahrzeuge.

Files, die auf das Fahrzeug übertragen wurden, können bei einem fahrzeuginternen FTP-Server abgeholt werden bzw. Dateien, die zum Landseitentransfer vorgesehen sind, dort abgelegt werden. Die Abgrenzung der einzelnen fahrzeugseitigen Systeme erfolgt über ein geeignetes FTP-Rechtekonzept.

2.2 Validierung

Die beschriebene RailML-Struktur erweitert den RailML-Standard dergestalt, dass die resultierenden Dokumente grundsätzlich gültig im Sinne von RailML 2.4 [1] sind. Daher ist eine Prüfung mit den Standard-RailML-Schemadefinitionen (XSD-Dateien) möglich. Zusätzlich wird im Rahmen der Projektentwicklung eine XSD für die Erweiterungen erstellt, welche eine Validierung der Erweiterungen ermöglicht.

2.3 Versionshandling

Im Header der RailML-Dateien werden Versionsnummern des RailML-Standards und Datenversionsnummern ausgetauscht.

Erkennt das Fahrzeug, dass die Zentrale mit eine neuere Schnittstellenversion vorliegen hat, so wird dies als Fehlermeldung dem Diagnosesystem im Fahrzeug mitgeteilt.

Folgende Konventionen werden getroffen: Im Element <dc:identifier> wird der String „STAP Passenger Information System Data Exchange“ geschrieben, im Element <dc:format> die aktuelle Versionsnummer des unterstützten Schnittstellenstandards, bspw. „1.3“.

2.4 Datentypen¹

string

Zeichenkette, die jedes gültige XML-Zeichen enthalten darf. Ist die Länge eines Stringtyps beschränkt, so wird dies in eckigen Klammern mit angegeben.

int

Dezimalwert zwischen -2147483648 bis 2147483647

unsignedInt

Dezimalwert zwischen 0 bis 4294967295

boolean

Logisch-Typ : 0 = FALSE / 1 = TRUE

dateTime

Dieser Datentyp beschreibt Zeitpunkte, die durch die Kombination eines Datums und einer Zeit identifiziert werden. Der Wertebereich wird in Kapitel 5.4 von ISO 8601 als »Kombination aus Datum und Tageszeit« beschrieben. Der lexikalische Raum ist das erweiterte Format »[]CCYY-MM-DDThh:mm:ss[Z](+|-)hh:mm]«. Die Zeitzone kann als »Z« (UTC) oder als »(+|-)hh:mm« angegeben werden. Nicht angegebene Zeitzonen werden als »unbestimmt« betrachtet.

enumeration

Definition einer Liste möglicher Werte für den Wertebereich eines Datentyps, meist vom Datentyp *string* oder *integer*.

Die konkreten Wertebereiche der beantragten Enumerationen sind in der Detailspezifikation im Kapitel 3 angegeben.

language

Der lexikalische und der Wertebereich von dem Datentyp `language` entspricht dem in XML definierten Typ `xml:language`, es ist hier sind die Menge der von RFC 1766 definierten Sprachcodes definiert.

coord

Definition eines Koordinatenpaares analog des Attributs `coord` des RailML-Elements `geoCoord`. (vgl. [1])

complex

Als Complex Type werden gemäß XML-Spezifikation Unterelemente bezeichnet, die wiederum eine eigene Struktur aufweisen.

¹ <https://www.data2type.de/xml-xslt-xslfo/xml-schema/datentypen-referenz/>

3 Beschreibung zu verwendeten RailML-Struktur

RailML verwendet verschiedene Sub-Schemata für verschiedene Teilaspekte der Modellierung im Bahnkontext. Fahrgastinformationsdaten verwenden Informationen aus dem Schemata

- Infrastructure
- Rolling Stock
- Timetable

Der größte Teil der relevanten Informationen kann mit den vorhandenen Konzepten in RailML abgebildet werden, die dabei für den hier vorliegenden Anwendungsfall nötigen Elemente werden hier aufgeführt.

An einigen Stellen der RailML-Spezifikationen wurden Erweiterungen erforderlich, um die Spezifika der Stadler-FIS abzubilden. Alle Erweiterungen wurden derart konzipiert, dass die Validität der Standard-RailML-Schemata nicht verletzt wird, da ausschließlich an vorgesehenen Extension Points neue Strukturen eingefügt werden. Alle neuen Elemente werden im Namensraum `:stap` zusammengefasst.

Neue Strukturen und Typen die subschemaübergreifende verwendet werden, werden im Kapitel 3.1 zusammengefasst, während die folgenden Kapitel jeweils ein Subschema strukturell spezifizieren.

3.1 Allgemeine neudefinierte Elemente

Hier aufgeführte Elemente werden an mehreren Stellen verwendet, in den nachfolgenden Kapiteln wird jeweils auf die hier vorliegenden Definitionen verwiesen.

3.1.1 Sichtbarkeitsbereiche (<stap:scopes>)

Ein Kernelement zur Steuerung von FIS-Prozessen bilden sog. Sichtbarkeitsbereiche, mit denen sich räumliche Umgrenzungen um Betriebsstellen definieren lassen. Diese Sichtbarkeitsbereiche ermöglichen die Festlegung von Triggern die zusammen mit Prozessdaten aus GPS bzw. Wegmeterzähler des Fahrzeugs für die Auslösesteuerung verschiedener Aktionen verwendet werden können.

Es werden verschiedene geometrische Formen von Sichtbarkeitsbereichen vorgesehen, welche alternativ verwendet werden können.

3.1.1.1 Sichtbarkeitskreis (<stap:scopeRadius>)

<stap:scopeRadius>: Sichtbarkeit, definiert über kreisförmige Fläche			
Type	enumeration: {snap, inner, outer}	1:1	Typ des Sichtbarkeitskreises (snap: Fangradius, inner: innerer Radius, outer:äußerer Radius)
Value	Int	1:1	Radius des Sichtbarkeitskreises, Angabe in Metern

3.1.1.2 Sichtbarkeitsabstand (<stap:scopeDistance>)

<stap:scopeDistance>: Sichtbarkeit definiert auf Entfernungsbasis			
Type	enumeration: {snap, inner, outer}	1:1	Typ des Sichtbarkeitsabstands (snap: Fangabstand, inner: innerer Abstand, outer:äußerer Abstand)
Value	Int	1:1	Länge des Sichtbarkeitsabstands, Angabe in Metern

3.1.1.3 Sichtbarkeitsrechteck (<stap:scopeRectangle>)

<stap:scopeRectangle>: Sichtbarkeitsrechteck			
Type	enumeration: {inner, outer}	1:1	Typ des Sichtbarkeitsrechtecks (inner: inneres Rechteck, outer:äußeres Rechteck)
<geoCoord>	complex	2:2	Geokoordinate (Lat, Lon) obere linke und untere rechte Ecke des Sichtbarkeitsrechtecks

3.1.1.4 Sichtbarkeitspolygon (<stap:scopePolygon>)

<stap:scopePolygon>: Sichtbarkeitspolygon			
Type	enumeration: {inner, outer}	1:1	Typ des Sichtbarkeitspolygons (inner: inneres Polygon, outer:äußeres Polygon)
<geoCoord>	complex	3:n	Geokoordinate (Lat, Lon) eines Polygonpunktes, Definition gemäß RailML-Spezifikation [1], die Verbindungen der Punkte dürfen sich nicht überschneiden.

3.1.2 Anzeigehalte (<stap:designator>)

Für die Abdeckung der unterschiedlichen Anwendungsfälle zur Ansteuerung der FIS-Anzeiger am Fahrzeug (bspw. TFT bzw LED-Außenanzeiger) wurde eine zweistufige Struktur vorgesehen. Die genaue Verwendung der definierten Anzeigen ergibt sich durch die Position innerhalb des RailML-Dokuments.

Werden die unterordneten Text-Elemente nicht spezifiziert gilt der Bezeichner in allen Sprachen und für alle Anzeigegeräte. Innerhalb von zugeordneten Textelementen definierte Values überschreiben den im Designator-Element angegebenen Wert.

<stap:designator>: Bezeichner für Anzeigen			
id	String	1:1	Identifizier des Bezeichners
value	String	1:1	Wert des Bezeichners
code	unsignedInt	0:1	Code für Bezeichner
<stap:text>	complex	0:n	Textelemente, siehe Kapitel 3.1.2.1

3.1.2.1 Anzeigetexte (<stap:text>)

Die Anzeigetexten zugeordneten Textelemente ermöglichen die Spezifizierung von Anzeigetargets (Ziel-Gerät) und Sprachzuordnungen.

<stap:text>: Text für Anzeigen			
xml:lang	Language	1:1	Sprachkürzel gemäß RFC 1766, Spezifikation analog XML-Standard
target	enumeration: {D, F, I, P, S}	1:1	Zielgerät der Anzeige : D: driver, F: front, I: interior, P: passenger, S: side)
value	String	1:1	Anzuzeigender Text

3.1.3 Ansagen (<stap:announcement>)

Analog zu Anzeigen ist eine feinere Untergliederung der Ansageinhalte erforderlich. Zu unterscheiden sind hier grundsätzlich Ansagen für den FIS-Regelprozess (bspw. Bahnhofsansagen) und Sonderansagen. Die genaue Typisierung ergibt sich zum einen durch die Positionierung innerhalb der RailML-Struktur, zum anderen kann feiner mit dem Attribut `type` spezifiziert werden.

<stap:announcement>: Ansage			
id	String	1:1	Identifizier der Ansage
xml:lang	Language	1:1	Sprache gemäß Spezifikation
audio	String	0:1	Referenz auf File
tts	String	0:1	Ansagetext für TTS
description	String	0:1	Beschreibungstext (z.B. für Anzeige im MMI)
code	String	0:1	Bezeichner für Austausch Zwecke
target	enumeration: {AL, AR, AB, AD, AP}	0:1	Ausgabeziel der Ansage: AL:side left, AR: side right, AB: side both, AD: dorrelease, AP: passenger
Type	String	0:1	Typ der Ansage. Die Sonderansagen (Type <i>special</i>) werden separiert angezeigt auf dem MMI

3.2 Subschema Infrastructure (<infrastructure>/)

Das Schema Infrastructure enthält Beschreibungen der ortsfesten Infrastruktur.

Für den Anwendungszweck gliedert es sich in zwei Bereiche:

1. **allgemeine Angaben** die über die gesamte Infrastruktur gültig sind oder unabhängig von der Infrastruktur gültig sind, diese werden im ersten Teil des Schemas direkt zu Beginn des Blockes <infrastructure> definiert und werden als RailML-Erweiterungen eingeführt
2. **Stationsstammdaten**, die jeweils für eine Station (gültig sind), diese werden innerhalb Stationsstammdaten im Bereich der Betriebspunktliste (<operationControlPoints >) definiert.

Es erfolgt eine auf einzelne Attribute und Unterelemente eingeschränkte Umsetzung gemäß RailML-Spezifikation [1], zusätzlich werden weitere Unterelemente eingeführt.

Minimal folgende Attribute von <infrastructure> sind umzusetzen

- Id

Es werden alle im Kapitel „Allgemeine Infrastrukturangaben“ (0) genannten Unterelemente von <infrastructure> neu definiert, diese werden direkt am Anfang des Infrastrukturteils eingeführt. Von den vorhandenen Unterelementen gemäß RailML-Spezifikation [1] sind die Metainformationen (Kapitel) und Stationsstammdaten (Kapitel) umzusetzen.

3.2.1 Allgemeine Infrastrukturangaben

3.2.1.1 Allgemeine Sichtbarkeitsangaben (<stap:scopes>/)

Spezifikation gemäß Kapitel 3.1.1

3.2.1.2 Allgemeine Anzeigen (<stap:designations>/)

Liste von Anzeigehalten, spezifikation gemäß Kapitel 3.1.2

3.2.1.3 Allgemeine Ansagen (<stap:announcements>/)

Liste von Ansagehalten, Spezifikation gemäß Kapitel 3.1.3

3.2.2 Infrastrukturmetainformationen (<metadata>)

Es erfolgt die vollständige Umsetzung gemäß RailML-Spezifikation [1].

3.2.3 Stationsstammdaten

(<infrastructure>/<operationControlPoints>/<ocp>)

Es erfolgt eine auf die folgenden Attribute und Unterelemente eingeschränkte Umsetzung gemäß RailML-Spezifikation [1]

Minimal folgende Attribute sind umzusetzen

- Id
- name

Von den Unterelementen von <ocp> sind die folgenden umzusetzen:

3.2.3.1 Stationstyp (<propOperational>)

Es erfolgt die vollständige Umsetzung gemäß RailML-Spezifikation [1].

3.2.3.2 Servicetyp (<propService>)

Es erfolgt die vollständige Umsetzung gemäß RailML-Spezifikation [1].

3.2.3.3 Stationskoordinate (<geoCoord>)

Es erfolgt die vollständige Umsetzung gemäß RailML-Spezifikation [1].

3.2.3.4 Stationsbezeichner(<designator>)

Es erfolgt die vollständige Umsetzung gemäß RailML-Spezifikation [1].

Zusätzlich werden folgende Unterelemente von <ocp> definiert:

3.2.3.5 Stationssichtbarkeitsbereich (<stap:scopes>)

Liste von stationsspezifischen Sichtbarkeitsbereichen, Spezifikation gemäß Kapitel 3.1.1

3.2.3.6 Stationsanzeige (<stap:designation>)

Spezifikation gemäß Kapitel 3.1.2

3.2.3.7 Stationsansagen (<stap:announcements>)

Liste von stationsspezifischen Ansagen, Spezifikation gemäß Kapitel 3.1.3

3.3 Subschema Rolling Stock (<rollingstock>)

Dieses Subschema definiert die Fahrzeuge und Fahrzeugverbände.

Es erfolgt eine auf einzelne Attribute und Unterelemente eingeschränkte Umsetzung gemäß RailML-Spezifikation [1], zusätzlich werden weitere Unterelemente eingeführt.

Minimal folgende Attribute von <rollingstock> sind umzusetzen

- Id

Von den Unterelementen sind die folgenden umzusetzen:

3.3.1 Fahrzeuge (<vehicles>/<vehicle>)

Es erfolgt eine auf folgenden Attribute und Unterelemente eingeschränkte Umsetzung gemäß RailML-Spezifikation [1]

Minimal folgende Attribute sind umzusetzen

- Id
- Name
- axleSequence
- numberDrivenAxles
- numberNonDrivenAxles
- length
- speed
- bruttoWeight
- nettoAdhesionWeight

Von den Unterelementen von <formation> sind die folgenden umzusetzen:

3.3.1.1 Klassifizierung (<classification>)

Es erfolgt eine auf folgenden Attribute und Unterelemente eingeschränkte Umsetzung gemäß RailML-Spezifikation [1]

3.3.2 Fahrzeugverbände (<formations>/<formation>)

Es erfolgt eine auf folgenden Attribute und Unterelemente eingeschränkte Umsetzung gemäß RailML-Spezifikation [1]

Minimal folgende Attribute sind umzusetzen

- id
- length
- speed
- weight

Von den Unterelementen von <formation> sind die folgenden umzusetzen:

3.4 Subschema Timetable (<timetable>)

Dieses Subschema definiert die Fahrplan- und Umlaufdaten.

Es erfolgt eine auf einzelne Attribute und Unterelemente eingeschränkte Umsetzung gemäß RailML-Spezifikation [1], zusätzlich werden weitere Unterelemente eingeführt.

Minimal folgende Attribute von <timetable> sind umzusetzen

- Id

Von den Unterelementen sind die folgenden umzusetzen:

3.4.1 Fahrplangültigkeiten (<timetablePeriods>/<timetablePeriod>)

Es erfolgt die vollständige Umsetzung gemäß RailML-Spezifikation [1].

3.4.2 Verkehrstage (<operatingPeriods>/<operatingPeriod>)

Es erfolgt die vollständige Umsetzung gemäß RailML-Spezifikation [1].

3.4.3 Zuggattungen (<categories>/<category>)

Es erfolgt die vollständige Umsetzung gemäß RailML-Spezifikation [1].

3.4.4 Zugteile (<trainParts>/<trainPart>/)

Es erfolgt die vollständige Umsetzung gemäß RailML-Spezifikation [1].

3.4.4.1 Verkehrstagerferenz (<operatingPeriodRef>)

Es erfolgt die vollständige Umsetzung gemäß RailML-Spezifikation [1].

3.4.4.2 Zughalt (<ocpsTT>/<ocpTT>)

Es erfolgt eine auf folgende Attribute und Unterelemente eingeschränkte Umsetzung gemäß RailML-Spezifikation [1]

Minimal folgende Attribute sind umzusetzen

- ocpRef
- ocpType
- sequence

Von den Unterelementen von <ocpTT> sind die folgenden mit den beschriebenen Erweiterungen umzusetzen:

3.4.4.2.1 Fahrzeiten (<ocpTT>/<times>)

Es erfolgt die vollständige Umsetzung gemäß RailML-Spezifikation [1].

3.4.4.2.2 Abschnitt (<ocpTT/sectionTT>)

Es erfolgt eine auf einzelne Attribute eingeschränkte Umsetzung gemäß RailML-Spezifikation [1]

Minimal folgende Attribute sind umzusetzen

- Distance

3.4.4.2.3 Haltbeschreibung (<ocpTT>/<stopDescription>)

Es erfolgt eine auf einzelne Attribute und Unterelemente eingeschränkte Umsetzung gemäß RailML-Spezifikation [1]

Minimal folgende Attribute sind umzusetzen

- commercial
- stopOnRequest

Von den Unterelementen von <stopDescription> sind die folgenden umzusetzen:

3.4.4.2.3.1 Gleisinformationen (<ocpTT>/<stopDescription>/<trackInfo>)

Es erfolgt die vollständige Umsetzung gemäß RailML-Spezifikation [1].

Zusätzlich werden folgende Unterelemente als Erweiterung von <ocpTT> vorgesehen:

3.4.4.2.4 Viahalt (<stap:via>)

<stap:via>: Auszeichnung von expliziten Via-Halten			
value	boolean	1:1	Definition ob Viahalt oder nicht

3.4.4.2.5 Fahrtziel (<stap:dest>)

<stap:dest>: Auszeichnung von expliziten Fahrtzielangaben			
designatorRef	String	0:1	Referenz auf den Identifier eines Bezeichners (vgl. Kapitel 3.1.2)
value	String	0:1	Bezeichner
announcementRef	String	0:1	Referenz auf eine Ansage (vgl. Kapitel 3.1.3)

3.4.4.2.6 Ausstiegsseite (<stap:exitside>)

<stap:exitside>: Ausstiegsseite			
value	enumeration: {left, right, both}	1:1	Wert für Ausstiegsseite
announcementRef	String	0:1	Referenz auf eine Ansage (vgl. Kapitel 3.1.3)

3.4.4.2.7 Haltansagen (<stap:announcements>)

Liste von haltspezifischen Ansagen, Spezifikation gemäß Kapitel 3.1.3

3.4.5 Züge (<trains>/<train>)

Es erfolgt die vollständige Umsetzung gemäß RailML-Spezifikation [1].
Zusätzlich werden folgende Unterelemente als Erweiterung von <train> vorgesehen:

3.4.5.1 Linienangabe

<stap:line>: Linienangabe für eine Zugfahrt; Definition direkt oder detaillierter über Verweis auf Designator bzw. Ansage möglich			
designatorRef	String	0:1	Referenz auf den Identifier eines Bezeichners (vgl. Kapitel 3.1.2)
value	String	0:1	String
announcementRef	String	0:1	Referenz auf eine Ansage (vgl. Kapitel 3.1.3)

3.4.5.2 Zugziel (<stap:dest>)

<stap:dest>: Auszeichnung von expliziten Fahrtzielen, Angabe direkt oder detaillierter über Verweis auf Designator bzw. Ansage möglich.			
designatorRef	String	0:1	Referenz auf den Identifier eines Bezeichners (vgl. Kapitel 3.1.2)
value	String	0:1	Bezeichner
announcementRef	String	0:1	Referenz auf eine Ansage (vgl. Kapitel 3.1.3)

3.4.6 Umlaufpläne (<rosterings>/<rostering>)

Es erfolgt die vollständige Umsetzung aller Attribute gemäß RailML-Spezifikation [1].
Von den Unterelementen sind mindestens die folgenden umzusetzen.

3.4.6.1 Umlaufelemente (<blockParts>/<blockPart>)

Es erfolgt die vollständige Umsetzung aller Unterelemente gemäß RailML-Spezifikation [1].

3.4.6.2 Umläufe (<blocks>/<block>)

Es erfolgt die vollständige Umsetzung inklusive aller Unterelemente gemäß RailML-Spezifikation [1].

3.4.6.3 Umlaufketten (<circulations>/<circulation>)

Es erfolgt die vollständige Umsetzung gemäß RailML-Spezifikation [1].

4 Dateiaustausch-Struktur

Teil einer FIS-Datenversorgung sind neben den Definitionen innerhalb des RailML-Dokuments Dateien, die bestimmten Content liefern. Für diese wird eine Filestruktur vorgegeben.

Grundsätzlich können Textdateien (bspw. für Konfigurationszwecke), Webseiten ohne externe Referenzen, Bilder-, Audio- und Videofiles übertragen werden.

Daher werden die folgenden Unterordner auf oberster Ebene vorgesehen:

/txt
/pic
/audio
/video
/html

Die jeweilige RailML-Definitionsdatei liegt dabei im übergeordneten Ordner. Wenn Files im RailML referenziert werden, so erfolgt die Dateinamensangabe relativ zum RailML-Basisordner, bspw. „*audio* = „*audio/gong.wav*“.

4.1 Textdateien (/txt)

Keine weiteren Festlegungen.

4.2 Webseiten (/html)

Es können komplette Webseiten (HTML-Code inklusive eingebetteter Elemente) hinterlegt werden. Diese Webseiten dürfen jedoch keine externen Verweise oder Nachladungen enthalten.

4.3 Bilddateien (/pic)

Keine weiteren Festlegungen.

4.4 Audiodateien (/audio)

Der Audiounterbaum enthält alle vordefinierten Ansagedateien die für FIS-Ausgaben benötigt werden, sofern keine reine TTS-Lösung zur Anwendung kommt.

Zu unterscheiden sind hier: Allgemeine Ansagedateien (bspw. Jingles), Stationsnamen und Sonderansagen. Für diese drei Typen werden drei Unterverzeichnisse definiert:

/common
/station
/special

Die Namen der Dateien orientieren sich an den Namenskonventionen, die im RailML festgelegt werden. Ist als primärer Stationsidentifikator bspw. eine Kodierung gemäß den Bahnnummern vorgesehen, so werden die Dateien entsprechend mit diesen Nummern benannt.

Für /common bzw. /special Dateien werden keine weiteren Konventionen getroffen.

4.5 Videodateien (/video)

Keine weiteren Festlegungen.

5 Dokumentenverweise

- [1] RailML-Spezifikation Version 2.4
 - a. <http://www.railml.org/>
- [2] Schnittstelle zwischen FIS und ITCS, Spezifikation der Datentelegramme und Ablaufsteuerung,
Stadler-Dokumenten-ID PA_2272527