This work was submitted at the
Institute for Railway Vehicles and Transport Systems

Master Thesis

Analysis of Methods for Converting and Modelling Track Infrastructure Data in RailTopoModel and Gazebo Simulation

submitted by

Chinenye Azubuike

INSTITUTE FOR RAILWAY VEHICLES AND TRANSPORT SYSTEMS
UNIV.-PROF. DR.-ING. C. SCHINDLER

Master Thesis

Chinenye Azubuike

Matriculation number: 390823

Subject:

Analysis of Methods for Converting and Modelling Track Infrastructure Data in RailTopoModel and Gazebo Simulation

Abstract: This thesis presents an overview of data, models and data sources that aid in the digitization and automation of railway systems, with focus on the railway infrastructure. The relationship between Building Infrastrucutre Modelling, RailTopoModel and railML is explained. Different methods and sources for converting and modelling data in railML 3.1 which implements the RailTopoModel are analysed and a method to convert the data and import to gazebo is also shown.

This work was carried out at the Institute for Rail Vehicles and Transport Systems in Aachen.

Faculty supervisor:	Univ.-Prof. Dr.-Ing. Christian Schindler Hyun-Suk Jung, M.Sc.
Place and date of submission:	Aachen, 15. August 2021

Eidesstattliche Versicherung

Ich versichere hiermit an Eides Statt, dass ich die vorliegende Project Work / Bachelor Thesis / Master Thesis mit dem Titel

Analysis of Methods for Converting and Modelling Track Infrastructure Data in RailTopoModel and Gazebo Simulation

selbständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht.

Für den Fall, dass die Arbeit zusätzlich auf einem Datenträger eingereicht wird, erkläre ich, dass die schriftliche und die elektronische Form vollständig übereinstimmen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Aachen, 15. August 2021
Ort, Datum

Chinenye Azubuike

Belehrung:

§ 156 StGB: Falsche Versicherung an Eides Statt

Wer vor einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

§ 161 StGB: Fahrlässiger Falscheid; fahrlässige falsche Versicherung an Eides Statt

(1) Wenn eine der in den §§ 154 bis 156 bezeichneten Handlungen aus Fahrlässigkeit begangen worden ist, so tritt Freiheitsstrafe bis zu einem Jahr oder Geldstrafe ein.

(2) Straflosigkeit tritt ein, wenn der Täter die falsche Angabe rechtzeitig berichtigt. Die Vorschriften des § 158 Abs. 2 und 3 gelten entsprechend.

Die vorstehende Belehrung habe ich zur Kenntnis genommen:

Aachen, 15. August 2021
Ort, Datum

Chinenye Azubuike

Table of Contents

List of abbreviations

IoT	Internet of things
BDA	Big Data Analysis
UITP	International Association of Public Transport
GoA	Grade of Automation
ADAS	Advanced Driver Assistance Systems
ARS	Automatic Route Setting
DB	Deutsche Bahn
ATO	Automatic Train Operation
ETCS	European Train Control System
RFF	Rail Freight Forward
Rus	Railway Undertakings
DLR	Deutsches Zentrum für Luft- und Raumfahrt
EBO	Eisenbahn-Bau- und Betriebsordnung
RINF	Register of Infrastructure
INSPIRE	Infrastructure for Spatial Information in the European Community
CIS	Comprehensive Information System
IMs	Infrastructure Managers
XML	eXtensible Markup Language
GML	Geography Markup Language
ERIM	European Rail Infrastructure Masterplan
UIC	International Union of Railways
IDMvu	Infrastructure data management for transport companies
DLG	Digital Line Graph
OSM/ORM	OpenStreetMap/OpenRailwayMap
TAF/TAP TSI	Technical Specification for Interoperability relating to Telematics Applications for Freight/Passenger services
EVN	European Vehicle Number
HRI	Human-Robot-Interaction

EU	European Union
BIM	Building Information Modelling
SDF	Simulation Description Format
LZB	Linienzugbeeinflussung
PZB	Punktförmige Zugbeeinflussung
STDR	Simple Two Dimentional Robot
ATB	Automatische Treinbeïnvloeding
RTTP	Real-Time Traffic Plan
IRS	International Railway Standard
railML®	railway Markup Language
NEMO	Network Evaluation Model
OGC	Open Geospatial Consortium
ASCII	American Standard Code for Information Interchange
UML	Unified Modeling Language
PBF	Protocolbuffer Binary Format
GIS	Geographic Information Systems
LiDAR	Light Detection and Ranging
OCP	Operation Control Point
CAD	Computer Aided Design
SDF	Simulation Description Format
CLBG	Chiemgauer Lokalbahn GmbH & Co KG
OPs	Operational Points
SWOT	Strength Weakness Opportunity and Threat
MRDS	Microsoft Robotics Developer Studio

1 Introduction

Digitization of the railway infrastructure is expected to boost mobility of people while offering new opportunities for flexibility, effectiveness, and efficiency [2]. Automation is a key technology that has accelerated the digital transformation of the railway sector in recent years [2] [4]. There are four main grades of automation in railway system, of which grade of automation 4 corresponds to a situation where a vehicle can be safely operate automatically and also react in a safe way in case of a disruption without the need of intervention by an on-board operational staff or driver [6]. This requires lots of onboard data, as well as communication between vehicles and between the vehicle and the infrastructure. The infrastructure data is an important aspect of the data that is necessary to facilitate automated driving, but it is necessary that this data is provided following a model, so that it can be easily generated and understood. In addition to the challenge of communication, there has to be a proof of feasibility of the automated driving (irrespective of the grade of automation) [15] which is achieved by extensive testings to ensure that the requirement of safety is met.

There are various data models and formats used to store and exchange infrastructure data. The most widespread formats such as the OSM and GeoJSON are based on the extensible markup language, however, there are also non-text formats such as the ESRI shapefile, depending on the source of the data and intended use. Public sources for railway infrastrucutre data include OpenStreetMap/OpenRailwayMap, microsoft bing, google maps and the Deutsche Bahn Open data which was launched in 2015. In the interest of providing a universal representation of a railway network and all events related to the rail sector, the International Union of Railways published the standard - RailTopoModel (IRS 30100) in September 2016 as an outcome of the European Railway Infrastructure Masterplan project. Prior to this standard, different infrastructure managers and softwares in the railway industry make use of different models and data formats to represent data, leading to difficulties when interacting with each other. railML, a data format for infrastructure data, implements the RailTopoModel in its latest versions (version 3.x), making it a very attractive format for most of the stakeholders in the railway industry.

An automated train can be modelled as a robot. In addition to physical testing, engineering design processes of robots can be hastened, made more economical, and benefit from more extensive testing by using simulations [49]. Different softwares such as webots, gazebo and coppelia sim can be used for the simulation, but first, the railway infrastructure data has to be imported into the robotic simulation software.

The different infrastructure data formats lead to many incompatibilities and inconsistencies, thereby impacting negatively on duration, quality and cost of railway operations. railML 3 represents one the most widely used implementation of the RailTopoModel, nevertheless, several railway data, especially publicly available data, are not available in this format.The main goal of this research work is to examine the possibility of accessing the railway infrastructure raw data of a small railway station in Obing from publicly available sources and converting the data to the RailTopoModel using railML 3.1 format. These sources could be lacking certain information regarding the infrastructure and might significantly impact the useability of the data for different applications/ use cases. So an analysis of the various sources is needed to identify which method is preferable in terms of accuracy and ease. Finally, because the infrastructure data is needed for simulation, it is necessary to also examine if there is a possibilty to import to gazebo simulation software, without building the track from the scratch.

This research work shows that railway infrastructure data from license free sources can be converted to railML 3.1 following the RailTopoModel and can also be converted and imported in gazebo. Chapter two discusses the state of the art on automation and shunting in railways, the different data aspects in the field of railway systems, the different geospatial data types, formats and terms. In this chapter, some of the most popular railway infrastructure data sources have also been explained, as well as the RailTopoModel, railML and railVIVID; and ends with an overview of Robotic simulation. Chapter three presents a short motivation behind this research topic, while chapter four surmarises the methods that will aid in the understanding of the RailTopoModel. RailTopoModel uses Unified Modeling Language and the topology is described following connexity graph theory. This chapter ends with different methods for calculating great circle distance which is important when obtaining the linear referencing method from the geographic coordinate system.

The steps and code snippets while modelling the Obing railway station are detailed in chapter five, which also contains a Strength Weakness Opportunity and Threat (SWOT) analysis of trying different data sources for the modelling process. Chapter six deals with importing the track infrastructure data to gazebo and chapter seven contains the conclusions drawn and possible future works.

2 State of the art

2.1 Automation of railway and shunting in overview

The wider use of digital processes and automation contributes greatly to the economic, social and environmental future of Europe [1]. The digitization of railway infrastructure is expected to boost mobility of people while offering new opportunities for flexibility, effectiveness, and efficiency [2]. Therefore, the thorough understanding of the concept of digital transformation is paramount in the development of rail transport [3]. Based on a research conducted by Roland Berger [4] in 2015 on major sectors of the German and European Economies, digital transformation takes effect by four levers (see Figure 1):

- Digital data
- Automation
- Connectivity
- Digital customer access

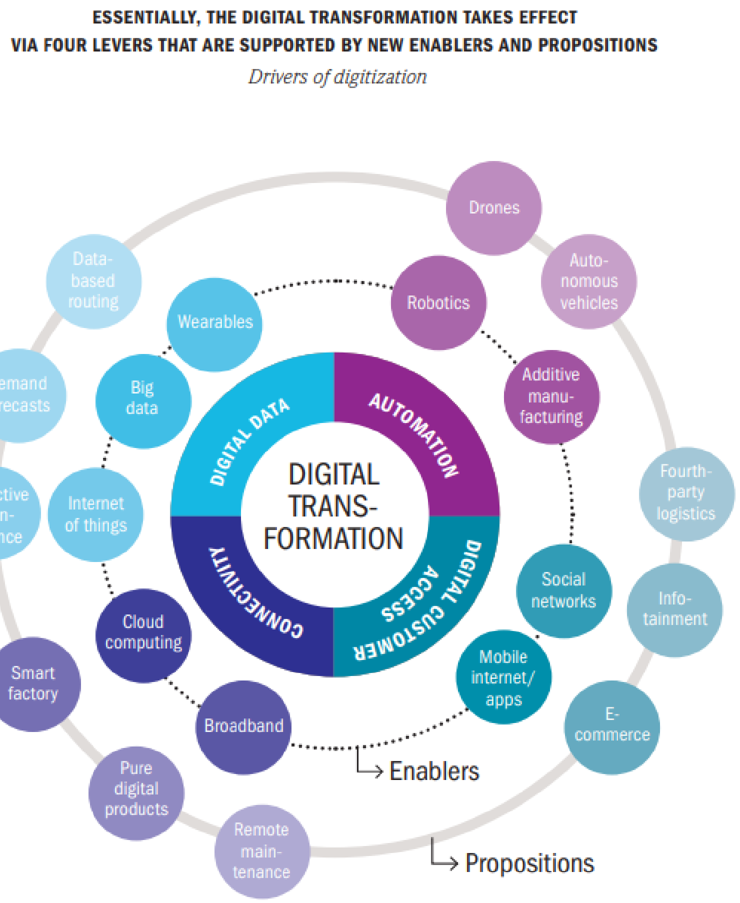


Figure 1: Drivers of Digitization (Source: [4])

Specifically for the railway sector, four main technologies and solutions were identified to have accelerated digital transformation in recent years - Internet of things (IoT), Cloud computing, Big Data Analysis (BDA), and Automation and Robotics [3].

2.1.1 Automated and Autonomous Driving

Automated driving in railway system is classified into four different grades by the International Association of Public Transport (UITP) starting from Grade of Automation (GoA) 1 to GoA 4, depending on the level of human interference either as a driver or as a train attendant (Table 1) [5]. GoA 4 represents a system where the vehicle runs fully automatic without any operational staff or driver on board of the train, and in the case of unforseen disturbances, is able to react in a safe way, which is a safe fall-back (usually standstill) position on its own [6]. There is also a GoA 0 which represents a drive on sight situation, without any form of assisted, automated or autonomous driving [6]. Autonomous driving however, is a situation where a vehicle can operate without any extraneous control and get all information required for safe operation and maneuvering (including during disruptions) from its on-board sensors without necessarily having to assume any safe fall-back state [6].

In road transport, automotive companies are already promising the possibility of fully automated cars (for a comparison of the grades of automation in automotive and rail industry, see [6]). Although several road vehicles are equipped with advanced driver assistance systems (ADAS), due to high social interaction involved in road transport, safety remains a high milestone if they must become fully automated.

Table 1: Grades of Automation in Rail Transport (Source: [5])

Grade of Automation	Type of train operation	Setting train in motion	Stopping train	Door closure	Operation in event of disruption
GoA1	ATP* with driver	Driver	Driver	Driver	Driver
GoA2	ATP and ATO* with driver	Automatic	Automatic	Driver	Driver
GoA3	Driverless	Automatic	Automatic	Train attendant	Train attendant
GoA4	UTO	Automatic	Automatic	Automatic	Automatic

**ATP - Automatic Train Protection; ATO - Automatic Train Operation*

On June 14 2019, Rio Tinto completed the transition to entirely automated (unmanned) operation of its 1500 km railway in the Pilbara region of Western Australia, making it the first heavy-haul railway in the world to operate on GoA 4 [7]. Vancouver in Canada, and Dusseldorf airport in Germany both have Sky trains which also run fully automated. Figure 2 shows the length of automated metro lines globally as of 2018. There were over 1000 kilometres of automated metro lines with a projection of 4000 kilometres by 2030 [5].

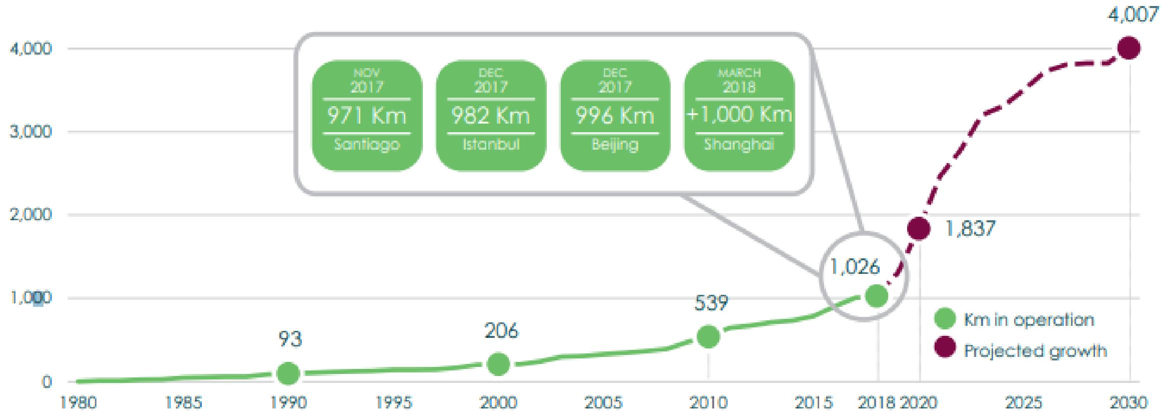


Figure 2: Growth of automated (at least GoA 2) metro globally, measured as km in operation (Source: [5])

In 2018, Deutsche Bahn (DB), Siemens and the city of Hamburg signed an agreement to develop an Automatic Train Operation (ATO) with GoA 3 (using European Train Control System (ETCS) level two for normal operations) for route section and GoA 4 for driving from and to depot for a 23 kilomter S-Bahn line (Line S21) between Berliner Tor and Aumühlein in the city of Hamburg by 2021 [8].

The European rail sector has committed itself to the goal of 30 % rail modal share by 2030, from the present 18 % in 2020 [9]. Currently, this goal is not set to be achieved. Therefore, the members of the Rail Freight Forward (RFF) initiative (which represents 90 % of the European rail freight market), has proposed that to achieve this goal, Railway Undertakings (RUs) should fully adopt three key technologies until 2030 – Digital Automatic Coupling, Automatic Train Operations (ATO) and Digital Platforms. For the ATO technology, it was proposed to fully deploy driving with supervision by a driver (GoA 2) on long haul and unattended train operations without driver (GoA 4) in shunting yards, on the first and last mile, and for fenced-in main line infrastructure [9].

DB is carrying out investigations into driverless operation on the railway, which is being tested in a pilot project at the Paderborn maintenance depot for shunting work. “Due to the considerable amount of shunting work within a relatively short time, automating the shunting process presents an opportunity to optimize the process in the depot and dispatch of the wagons so that shunting around the clock is possible” [10]. Alstom and ProRail plan to test automatic train operation in 2021, using Lineas shunting locomotive [11]. Rail Vision [12] in a case study recognised three major challenges facing the shunting yard operation – manpower shortages, safer work environment and a growing need for automation. In 2018, SBB Cargo AG entered partnership with Rail Vision to tackle these challenges by deploying an automated assisted shunting process with obstacle detection to reduce the required number of workers on the track from two to one. Shunting yards operations are usually at low speed and are confined to limited areas of yards, and therefore have less human interaction compared to passenger traffic, which increases its potential for automation [12].

2.1.2 Advantages of Automation in Railway

Experts of UITPs Observatory of Automated Metros [5] Identifies mobility, safety, affordability, ecology and humanity as the five key benefits of fully automated metro operation. These benefits are further declined in ten areas as seen in Figure 3.

Figure 3: Five key benefits of fully automated operation declined in 10 areas, as identified by the experts of the Observatory of Automated Metros (Source: [5])

Recent studies by Deutsches Zentrum für Luft- und Raumfahrt (DLR) prove that in adopting ATO in train operations, ETCS Level 3 should be the focus instead of ETCS Level 2 limited supervision, because the initial cost reductions based on the reduction in infrastructure, and large performance increases due to the possibility of Moving Block operations are only possible with ETCS level 3 [13]. Rail Vision also assert that ATO increases efficiency and safety and leads to a reduction in number of workers required to be on tracks which indirectly translates to an increase in profit [12]. A case study for the application of automated railcars in terminal Rotterdam Maasvlakte shows a reduction of 30 % in container handling time and inter-terminal transport time [14]. A project termed the “brain-train concept” which is expected to provide a structural change in the Aachen-Mönchengladbach region and address innovative rail mobility in rural areas will use fully automated (GoA 4) passenger operations within the rural areas of the Aachen-Düren-Jülich region of Northrhein Westphalia [15].

The benefits of automated driving in railway cuts across all sectors from passenger operation (metro, mainline, etc) to freight transport and most significantly, shunting yard operations. However, to roll-out automated driving on a large scale and harness these benefits, several corresponding challenges still need to be addressed.

2.1.3 Challenges of Automated Driving in Railways

Automation in the United Kingdom (UK) mainline railways began in four major stages: 1989 - Standard electronic signalling control system first incorporated Automatic Route Setting (ARS), 2008 - ARS specification published to open up market, 2009 - Development of Traffic Management systems commenced and 2012 - Development of Automatic Train Operation (ATO) commenced [76]. A study conducted by network rail has shown the human factor challenges that affect each stage of the automation of UK railway, some of which include [76]:

- ARS-specific timetable data is difficult to come by at a high enough quality level
- The automation does not cover everything, leaving some tasks for the signaller to carry out. For instance, shunting, including signalling trains into an occupied section is still left to signaller
- Where timetabled operation is disrupted, the ARS is least effective and the task can fall back to a signaller
- Planners and signallers of the traffic management system can both independently change the plan
- It is still unclear what the impact of ATO would be on vigilance, driving skills and route knowledge for train drivers.

In Germany, some of the major challenges identified for implementing the brain-train project include [15]:

- Definition and development of operating concepts of an autonomous or fully automated passenger train and monitoring level crossings for automated driving
- Proof of feasibility and demonstration of driverless driving
- Clarification of liability and licensing aspects for operation with driverless vehicles. Presently in Germany, it is mandated by regulation (Eisenbahn-Bau- und Betriebsordnung (EBO)) that a locomotive must be manned by a driver while driving

- Obstacle detection
- Communication between the infrastructure and the vehicle, as well as between the vehicles.

Automation in railway is dependent on different kinds of data and its effective communication. In addition to communication of data between the infrastructure and vehicle and between vehicles, information regarding the infrastructure and the vehicles needs to be monitored to ensure safety [16]. The completeness and complexity of the monitoring process is also linked with the collection of data which is either done automatically or manually [16].

2.2 Data aspects in Railway Systems

The railway network, though an independent system, connects with different departments, authorities and companies, and must therefore seamlessly work together in the conception, approval, design, manufacturing, testing, operation and maintenance phases along the value chain. In all these stages, information and data need to be shared among the different stakeholders for the requirements of safety, comfort and efficiency to be achieved. This sharing could be done locally (within the same organisation), nevertheless, there are situations where it is necessary to share such data with other stakeholders including the customers (like the timetable information). This means that the data must not just be in a format that is recognizable and readable by the generating party but should also be understandable and useable by third parties. However, in most European member states, some railway related data are collected only at company level and shared only when strictly necessary or for legal reasons [16].

There have been several specialized data formats created for both rail and non-rail focused purposes such as the Register of Infrastructure (RINF) and the Infrastructure for Spatial Information in the European Community (INSPIRE). The design of each format was specific in both requirement and discipline [17].

As of 2002, there were more than six management systems on the Chinese railway network, which were developed independently and differently for solo purposes before a Comprehensive Information System (CIS) was put forward [18]. A CIS considers a rail network as an integrated system from rail bed including bridge and tunnel, rails and points to vehicle, locomotive, and train operation [18]. Under this system, there are six parts of a railway network: (1) the data center and control center (the headquarter); (2) the depots (cars, vehicles and locomotives); (3) the logistics centers (materials for rails, bridge, tunnel and rolling stocks etc.); (4) the maintenance centers; (5) the stations (marshalling yards, normal station, junction station) and (6) the wayside systems in blocks.

The 2016 EU Interoperability Directive [19] breaks the railway system into subsystems: infrastructure, trackside control-command and signaling, on-board control-command and signaling, energy, rolling stock, operation and traffic management, maintenance, and telematics applications for passenger and freight services. Each of these subsystems can then be further classified into fixed, mobile or managerial elements as shown in Table 2.

Table 2: Classification of the railway subsystems into elements (Source: [19])

Sub-system	**Classification**
Infrastructure	Fixed elements
Trackside control-command and signaling	Fixed elements
Energy	Fixed elements
On-board control-command and signaling	Mobile elements
Rolling stock	Mobile elements
Operation and traffic management	Managerial elements
Maintenance	Managerial elements
Telematics applications for passenger and freight services	Managerial elements

railML (see 2.6) classifies the railway system in four different areas – Infrastructure, rolling stock, timetable and interlocking [20] which is adopted in this work to represent the main data aspects relevant for railway operations. In Table 3, the different formats, interface and/or models that are used to store data are surmarised.

Table 3: Models, formats and interfaces for railway data

Data Aspect	**Name**	**Model/Format/Interface**
INFRASTRUCTURE	RINF	Format
	GEOJSON	Format
	railML	Format
	NeTEx	Model and format
	RTM	Model
	IDMvu	Format
	OSM	Format
	OVERPASS API	Interface (OSM)
TIMETABLE	CIF	Format (Network Rail)
	RTTP	Format
	REST	Interface (Deutsche Bahn)

	railML	Format
	NeTEx	Model and format
ROLLING STOCK	railML	Format
INTERLOCKING	PlanPro	Model and format
	EULYNX DP	Model
	railML	Format

2.2.1 Infrastructure

A railway infrastructure data contains information on the topology, geometry, coordinates, tangible assests (e.g. balises, level crossings etc) and intangible assets (e.g. speed profile) of a railway network [21] [22]. In railML, these assets are classified as functional infrastructure (see 2.6). UIC defines a network topology as “the arrangement of the iron network and the related infrastructure data, and its relationship with objects and events located along the network” [22]. Infrastructure managers (IMs) need to have information about the status, characteristics, and utilization of their assets. Therefore, the challenge of collecting, storing, maintaining, and presenting infrastructure data must be overcome. Data models are used to organise the data in a structured way. The structure of the data model should support – besides the actual storage - the semantic dependencies of entities and the over-all coherence [24].

A typical infrastructure data should contain properties of the network elements (tracks, platforms, stations, tunnels, bridges etc) such as length, height, curve radius, etc; the relations (connection between them); the network entities which comprise the physical assets (like balise, switch, signals, level crossings etc), speed limits, gradients, radio coverage etc; and the location. There are different levels of description of the topology (covered in 2.5.2) depending on the requirement and scope [23] [21].

Various data models and formats are being used to store and exchange infrastructure data. The most widespread formats are based on the extensible markup language (XML, or expandable markup language in German). European Rail Infrastructure Masterplan (ERIM) Task Force of the UIC [25] has conducted a feasibility study on several existing national models, their technical characteristics, topology elements, features, structure and specialities are outlined. In this feasibility study, a clear distinction was made between a model and format and how they work together. A model specifies how to describe infrastructure elements (including its attributes) using any topological representation, whereas a format is one of many possible representations of a model usually in form of a text, for exchanging model objects [25]. In other words, a model tells how to describe an object, while a format follows the rules of the model to describe the object in text form. It is also possible to use the same exchange format to share data of several models, provided there is an adapter (conversion interface) as shown in Figure 4 [25].

In the result of the work package 3.4 of the EU project Capacity4Rail [26], a comparison of the different XML and non-XML-based data models and formats was made. Some XML-based data formats and models used for infrastructure data are the railML, NeTEx, RailTopoModel (RTM), Register of Infrastructure (RINF), Infrastructure data management for transport companies (IDMvu) etc., while OpenStreetMap (OSM)/ OpenRailwayMap (ORM) is the most well-known non-XML-based database and format used in railway operation [26]. However, of all the above-mentioned formats, only the railML3 and the OSM allow for the aggregation from microscopic to mesoscopic level [26].

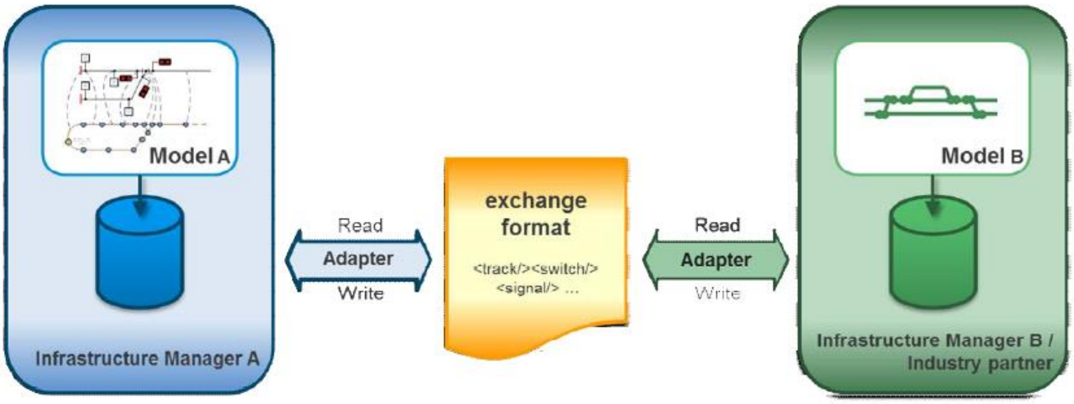


Figure 4: Model, Format, and adapter (Source: [25])

2.2.2 Rolling Stock

This refers to all kinds of vehicles that run on the infrastructure. Information about the propulsion system, dynamic characteristics of the rolling stock (braking, acceleration, etc) and all other data concerning the train are classified here [22]. Data describing the properties of the rolling stock help in its maintenance and is essential for vehicle management as well as for detailed runtime calculations. Different levels of detail of the rolling stock data could be described [22]. Data related to the rolling stock are necessary to, and collected by, both the IM and RU [16].

Some data collected by the IM for each train include [16]:

- From the Technical Specification for Interoperability relating to Telematics Applications for Freight/Passenger services (TAF/TAP TSI) (TSI OPE – Reg. 2015/995) - train identification, identification of reporting point line on which the train is running, scheduled time at reporting point, actual time at reporting point (and whether depart, arrive or pass — separate arrival and departure times must be provided in respect of intermediate reporting points at which the train calls), former train identification(s) if any, train cancelled for a whole or a part of its journey etc

- European Vehicle Number (EVN) of the engine and EVN of each vehicle composing the train
- weight and length of the train
- running speed and maximum speed
- type of goods
- weight of the rolling stock etc

While some data collected by RU related to rolling stock include data collected via:

- pre-departure checks
- maintenance reports
- on-board monitoring systems
- on-board diagnostic systems
- on-board recording devices etc

railML and TAF TSI are formats currently being used for rolling stock data and real time monitoring and whereas the TAF TSI is targeted for the cross-modal exchange of information in freight transport [20][26].

2.2.3 Timetable

A railway timetable is a representation of all movements of the rolling stock on a railway network consisting of at least the passing times at a specific location and routes to be passed. The different organisations involved in operating the railway system view the timetable in different ways. For the RU who owns and operates the trains (passenger or cargo trains), the timetable describes the planned movements; while for the IMs, the timetable describes the planned utilization of the infrastructure [27]. For the passengers, the arrival and departure time of trains, and the number of possible transfers are most important in a timetable [27].

Like infrastructure and rolling stock data, there are different data formats and models used by different countries and IMs to save and share timetable data or for automatic generation and simulation of timetables. DB provides its timetable data for long distance travel via an XML/JSON-based Representational State Transfer (REST) interface [28] (Figure 6). In the UK, the Common Interface File (CIF) data format is used [29] as seen in Figure 5. Other common XML-based formats used in the timetabling are Real-Time Traffic Plan (RTTP) and NeTEx (also a model), while TAP TSI and GTFS are non-XML-based.

```
/ Association between P39948 and P39725 occurring at Paddington

AANP39948P39725980830980830000001    PADTON   T                                C

/ Train schedule for Train C53290 between Bedford and Croydon

BSNC53290170524170920000001 POO2T07     124207004 EMU319 100D     B            P
BX         TLYTL123400
LOBEDFDM  0841 08411  SL     TBH
LIBEDFDS            0843 00000000
LIFLITWCK 0850 0850H     08500850         T
LIHRLG    0854 0854H     08540854         T
LILEAGRVE 0859H0900      09000900         T           1
LILUTON   0904 0905      090409041        T             1H
LIHRPNDN  0912H0913      09120912         T           1
LISTALBCY 0919 0920      091809181        T
LIHDON              0930H00000000
LIWHMPSTM           0933 00000000
LIKNTSHTN           0935H00000000   MOL               2
LIKNSXMCL 0941H0943      09420943A        T
CRFRNDNLT OO2T07    124612004 EMU319 100D     B
LIFRNDNLT 0945H0946H     094609463        T
LICTMSLNK           0947H000000002
LIBLFR    0949 0950      094909504        T
LIMTRPLTJ           0953H00000000
LILNDNBDE 0955 0956      095509565  5     T
LISPAROAD           0958H00000000
LIBRCKLAJ           1000 00000000   FL                3
LINORWDJ            1010 000000004
LINORWDFJ           1011 00000000   SL
LIWNDMLBJ           1011H00000000   SL
LTECROYDN 1013 10136     TF
```

Figure 5: Excerpt of Basic timetable data in CIF format (Source: [29])

```
XML:
<JourneyDetail xsi:noNamespaceSchemaLocation="https://DOMAINE-TOBE-
DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/xsd?name=hafasRestJourneyDetail
.xsd">
    <Stops>
        <Stop name="Frankfurt(Main)Hbf" id="8000105" lon="8.663785"
lat="50.107149" routeIdx="0" depTime="15:02" depDate="2016-02-22"
track="13"/>
        <Stop name="Erfurt Hbf" id="8010101" lon="11.038501"
lat="50.972549" routeIdx="1" arrTime="17:10" arrDate="2016-02-22" dep-
Time="17:12" depDate="2016-02-22" track="10"/>
        <Stop name="Halle(Saale)Hbf" id="8010159" lon="11.987088"
lat="51.477509" routeIdx="2" arrTime="17:46" arrDate="2016-02-22" dep-
Time="17:48" depDate="2016-02-22" track="6"/>
        <Stop name="Berlin Südkreuz" id="8011113" lon="13.365314"
lat="52.475042" routeIdx="3" arrTime="18:54" arrDate="2016-02-22" dep-
Time="18:56" depDate="2016-02-22" track="8"/>
        <Stop name="Berlin Hbf (tief)" id="8098160" lon="13.369548"
lat="52.525589" routeIdx="4" arrTime="19:01" arrDate="2016-02-22" dep-
Time="19:04" depDate="2016-02-22" track="6"/>
        <Stop name="Berlin Gesundbrunnen" id="8011102" lon="13.388515"
lat="52.548961" routeIdx="5" arrTime="19:09" arrDate="2016-02-22"
track="9"/>
    </Stops>
    <Names>
        <Name name="ICE 1631" routeIdxFrom="0" routeIdxTo="5"/>
    </Names>
    <Types>
        <Type type="ICE" routeIdxFrom="0" routeIdxTo="5"/>
    </Types>
    <Operators>
        <Operator name="DPN" routeIdxFrom="0" routeIdxTo="5"/>
    </Operators>
    <Notes>
        <Note key="BR" priority="450" routeIdxFrom="0" rou-
teIdxTo="5">Bordrestaurant</Note>
    </Notes>
</JourneyDetail>
```

Figure 6: Excerpt of XML journey details from DB Timetable API (Source: [28])

For automatic generation of timetables, the RailSys uses the Network Evaluation Model (NEMO) to construct timetables on a macroscopic level (see 2.5.2), especially to investigate the impact of new infrastructure and operating program [30]. The Periodic Event Scheduling Problem (PESP) is a mathematical model that is also used in periodic railway timetabling. NEMO and PESP are not models used for storing data. There are also other train time tabling models as shown in surveys by [31] [32]. However, in principle, each model falls under one of three categories [27]:

(i) trip based models where continuous variables are used for determining departure and arrival times of trains

(ii) trip based models with a time discretization and variables are used for determining departure and arrival times of trains

(iii) path based models where for each train a set of potential paths is generated, and a set of non-conflicting paths is selected based on a set covering model

2.2.4 Interlocking

In Railway Operation and Control, Interlocking can be described in two meanings: "the interlocking plant where points and signals are interconnected in a way that each movement follows the other in a proper and safe sequence" and secondly, "the principles to achieve a safe inter connection between points and signals" [33]. Important aspects of the interlocking technology are the data points, the signals and switches. Data relating to these aspects are required in the planning of ETCS compliant tracks.

To promote the data-based planning of interlocking systems, PlanPro - a model for the signalling systems - was created as an XML schema [34]. This was done by a joint working group under the lead of DB Netz AG with the goal to exchange all future planning data in this common file format (see Figure 7). PlanPro is expected to have four major effects [34]:

- Digital storage of planning and inventory data in a signalling database
- Quality assurance for planning data and planning process
- Electronic transfer of planning data to the manufacturer of interlockings
- Using of inventory data in the life cycle

Eclipse Signaling Engineering Toolbox was therefore developed by Eclipse foundation to provide essential components of the tool chain to manage initialization, testing, validation and visualization of the PlanPro model for the railway interlocking signalling technology [35].

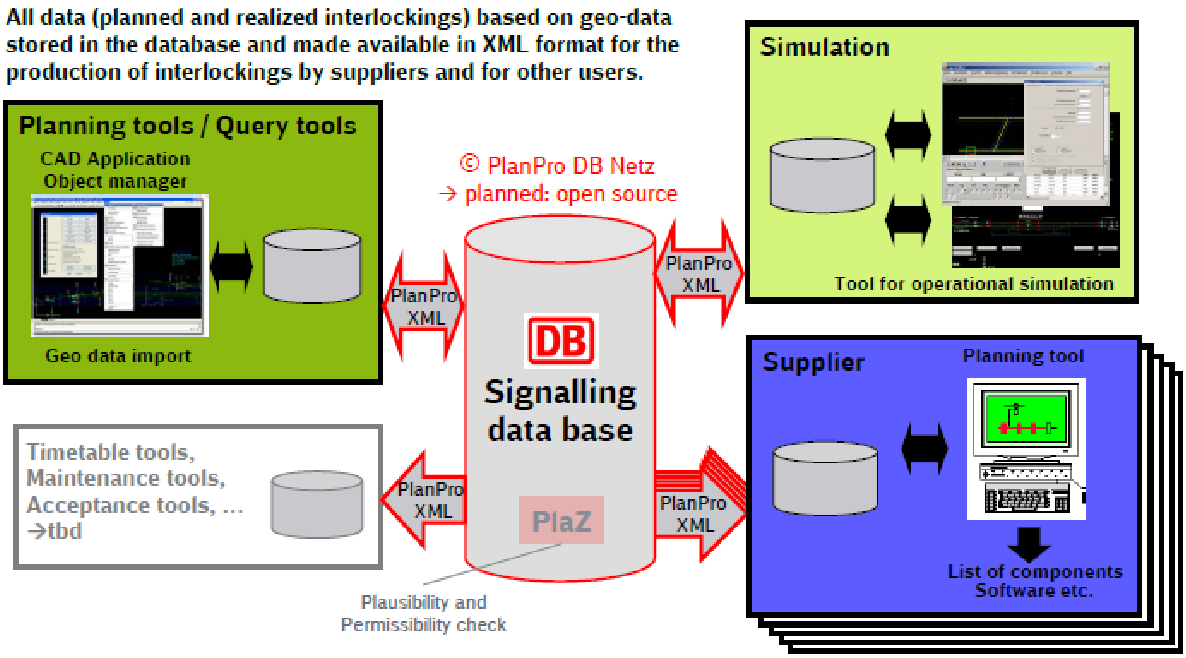


Figure 7: Architecture of the users and use cases of PlanPro (source: [34])

Also, EULYNX, which is a cooperation of thirteen European IMs including DB and ProRail, aims to create a standardized interface and field elements of the railway signalling systems [36]. Conventionally, the IM shares signalling data with the suppliers by means of plans, texts and tables in propriety formats as seen in Figure 8 [68] [69]. This process results in lots of waste mostly caused while resolving errors from ingesting and interpreting data (see Figure 9).

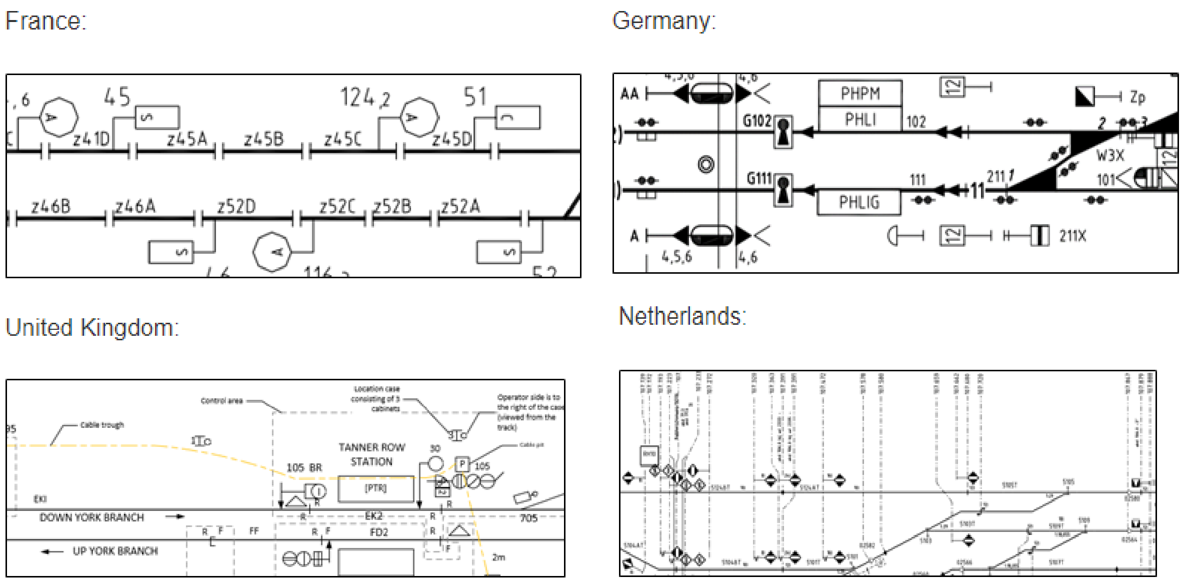


Figure 8: Traditional shapes and formats used by different IMs for presenting signalling information (Source: [69])

Therefore EULYNX began creating the EULYNX Data Preparation (DataPrep or DP) which is a platform independent UML class model of the information needed to engineer the signalling system for specific yards, which solves the problem of waste. Table 4 shows the packages contained in EULYNX DP. Its main benefit is that the information that meets the EULYNX DP schema is compatible for Machine-to-Machine communication [68]. This implies that the IM's EULYNX DP engineering information feeds directly into the suppliers's engineering tool chain. The supplier then designs, tests and carries out the signalling work, and returns the as-built information directly into the IM's IT systems (Figure 9) [68].

Figure 9: IT Processes within the signalling supply industry (Source: [69])

This lack of a uniform standard for data format and models in the railway industry pose a challenge for the future development of the industry, especially while driving digitization and automation of processes. Furthermore, the different formats lead to many incompatibilities and inconsistencies, thereby impacting negatively on duration, quality and cost of railway operations. Hence, there is the need for developing standards such as the IRS 30100 by UIC described in 2.5.

Table 4: Packages in EULYNX Data prep (Source: [69])

<table>
<tr><th>Work Packages in current snapshot</th><th>Work packages in development</th></tr>
<tr><td rowspan="5">Assets
• Block
• Key locks
• Level crossing
• Local operating area
• Local operation devices
• Object controller
• Platform
• Point, crossing and derailer
• Signal
• Support structures
• Track
• Train detection
• ETCS</td><td>Operation</td></tr>
<tr><td>Project management</td></tr>
<tr><td>Automatic Route Setting</td></tr>
<tr><td>Train</td></tr>
<tr><td></td></tr>
</table>

• Geo data • Power supply • Workers' safety	
Common classes	
Flank protection	
Route	
Speed profiles	
Train protection	

2.3 Geospatial Data Formats and Terms

Geospatial data (or simply spatial data) refers to any information that relates to a location on the surface of the earth. A spatial data usually has other information (other than location) that describes its features. Such an information is called attributes. Geographic coordinate system is used to identify the positions of objects on the surface of the earth. Georeferencing is the process of assigning the coordinates of a map frame (which represents a model of the earth) to the geospatial data so that they can be accurately oriented and queried. The field of processing and analyzing spatial data is called Geographic Information Systems (GIS).

There are two main categories of Geospatial data namely: vector data and raster data. Other types of Geospatial data include Light Detection and Ranging (LiDAR) data and 3D data (Table 5).

- **Vector data**: is a graphical representation of the real-world and are not made up of grids of pixels. The main symbol types for a vector data are points, lines, and polygons. Some formarts for vector data include: Geography Markup Language (GML) which is associated with the file type .gml, Digital Line Graph (DLG), GeoJSON, ESRI Shapefile OpenStreetMap (OSM) XML (which is associated with the file type .osm) etc. An alternative to the .osm which is XML-based is the smaller Protocolbuffer Binary Format (PBF).

- **Raster data**: is a Geospatial data that is made up of pixels. They are Geospatial images that are taken by satellite imagery or other aerial devices. File formats associated to this data include ERDAS Imagine (IMG) in .img; American Standard Code for Information Interchange (ASCII) Grid in .asc; GeoTIF in .tif, .tiff and .ovr; and IDRISI Raster in .rst and .rdc.

- **LiDAR data**: also known as point cloud data, is a network of coordinate points and elevation values (Figure 10). The LiDAR file format include the extensions ASPRS LiDAR Data Exchange Format (with the file types .las, .lasd, and .laz) and Point Cloud XYZ (with the file type .xyz).

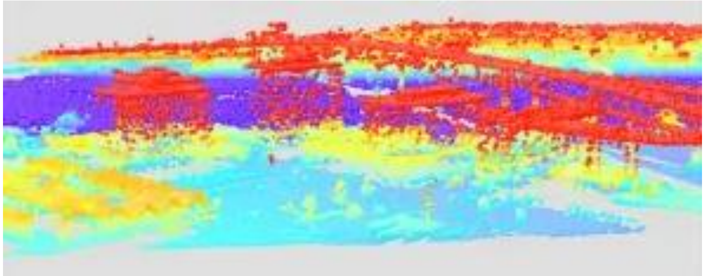


Figure 10: Example of a LiDAR data (Source: https://gisgeography.com/gis-formats/)

- **3D data:** adds the elevation and/or depth to the conventional 2D longitude and latitude coordinate data. They are graphic representations of objects in the real world developed using 3D modelling software. Examples of 3D formats are the Collada (.dae) and Trimble Sketchup (.skp).

Table 5: Geospatial data types

Type of Data	Name	Extensions	Supports multi-object	Specific coordinate reference system
Vector	Geojson	.geojson, .json	yes	WGS84 reference
	Esri shapefile	.shp, .shx, .dbf, .prj	no	non-specific
	GML	.gml	yes	non-specific
	KML	.kml, .kmz	yes	WGS84 reference
	DLG	.dlg	yes	non-specific
	MapInfo TAB	.tab, .map, .ind, .id	yes	non-specific
	OSM	.osm, .pbf	yes	WGS84 reference
Raster	ERDAS Imagine	.img		
	ASCII Grid	.asc		
	GeoTIF	.tif, .tiff, .ovr		
	IDRISI Raster	.rst, .rdc		
3D	Collada	.dae		
	Trimble sketchup	.skp		
LiDAR	ASPRS LiDAR Data Exchange Format	.las, .lasd, and .laz		
	Point Cloud XYZ	.xyz		

2.3.1 Geographic Java Script Object Notation (GeoJSON)

GeoJSON is a geospatial data interchange format based on JavaScript Object Notation (JSON). It's file type includes .GEOJSON and .JSON. A GeoJSON object is a JSON object and is either a geometry, feature, or collection of features (see 2.3.1.3). Since its first publication is 2008, it has increasingly grown in popularity. It is widely used in JavaScript web mapping libraries and JSON-based document databases [37]. With GeoJSON format, vector data and its attribute(s) can be stored in a single file. Geospatial data from DB Open data portal and OSM are available in this format.

2.3.1.1 Cordinate Reference System

GeoJSON uses the World Geodetic System of 1984 (WGS84) datum as the default reference system with longitude and latitudes expressed in decimal degrees. Although geographic data coordinates can be expressed in sexagesimal (base 60, eg 15° 12’ 45’’) or other referencing systems, they are not recommended in GeoJSON format. For geographic coordinates with units of degree, it is recommended to use six decimal places, which amounts to a precision of 10 centimeters and is well within the precision of current gps systems [37]. Increasing the precision than necessary could come at the cost of double the number of GeoJSON text, depending on the number of detailed polygons and additional decimal points [37].

2.3.1.2 Position

This is an array of coordinates. The coordinates must have at least two elements and must follow the order longitude (or easting) followed by latitude (or northing). A third element – altitude (or elevation), may be included. This ensures consistency with the x, y, z sequence in math.

```
[longitude, latitude, elevation]
```

2.3.1.3 Geometry

Geometries are simply shapes. There are seven Geometry “type” in GeoJSON namely: Point, MultiPoint, LineString, MultiLineString, Polygon, MultiPolygon and GeometryCollection. Each Geometry in GeoJSON must have a “type” member and a “coordinates” member, except for the “type” GeometryCollection which do not need to have “coordinates” member.

Points: For this type of geometry, the “coordinates” member is a single position. It is therefore suitable to describe non-linear geospatial elements lying anywhere in space.

```
{“type”: “Point”, “coordinates”: [50.1, 30.1]}
```

MultiPoint: For this type, the “coordinates” member is an array of positions.

```
{“type”: “MultiPoint”, “coordinates”: [ [50.1, 30.1], [40.8,30.5] ] }
```

LineString: In this type of Geometry, the “coordinates” member is an array of two or more positions. These positions connect each other to form a line.

```
{“type”: “LineString”, “coordinates”: [ [50.1, 30.1], [50.2,29.9] ] }
```

MultiLineString: The “coordinates” member is an array of LineString coordinate arrays.

```
{“type”: “MultiLineString”, “coordinates”: [ [ [50.1, 30.1], [50.2,29.9] ], [ [50.3, 30.3], [50.4,29.8] ], [ [50.2, 30.1], [50.3,29.9] ] ] }
```

Polygon: This type of Geometry is significantly more complex; hence it is necessary to understand the concept of linear ring. A linear ring is a closed LineString with four or more positions, such that the first and last positions are identical values. This means that the linear ring represents boundaries of a surface or a hole in a surface. The right-hand rule must be followed in identifying exterior rings (anticlockwise) and holes (clockwise).

For the Geometry type “Polygon”, the “coordinates” member must be an array of linear ring coordinate arrays. When the two kinds of rings need to be described, the first array of “coordinates” for the polygon must be the exterior ring, and the other(s) must be the interior ring(s). This is useful when describing geographic information such as an Island (exterior ring) which has a lake in it (interior ring).

```
{“type”: “Polygon”, “coordinates”: [ [50.1, 30.1], [50.0,29.9], [ [50.3, 30.3], [50.1,30.1] ] }
```

In the example above, there is only one ring which signifies the outer ring, therefore the polygon has no inner ring.

MultiPolygon: This is simply an array of polygons.

```
{“type”: “MultiPolygon”, “coordinates”: [ [ [ [50.1, 30.1], [50.0,29.9], [ [50.3, 30.3], [50.1,30.1] ], [ [50.1, 30.1], [50.0,29.9], [ [50.3, 30.3], [50.1,30.1] ] ],

 [ [ [50.1, 30.1], [50.0,29.9], [ [50.3, 30.3], [50.1,30.1] ] ] ]

 }
```

The above example is a MultiPolygon (enclosed in red square brackets) that contains two polygons (enclosed in purple square brackets). The first Polygon has an outer (enclosed in yellow square brackets) and inner (enclosed in green square brackets) ring, and the second Polygon has only an outer ring (enclosed in grey brackets).

Geometry Collection: As the name implies, it is a collection of any of the smaller geometry objects. This collection can also be heterogenous, which differentiates it from all other geometry types. Geometry Collection must have a member with the name “geometries”, whose value is an array. The elements of the array represent any of the smaller GeoJSON geometry object. This type is used when different geometries all refer to the same thing, for example a continent or country with disjointed territories.

```
{"type": "GeometryCollection",

    "geometries": [{"type": "Point",

                  "coordinates": [50.1, 30.1]

                  },
```

```
                    {"type": "Point",
                    "coordinates": [49.9, 30.2]
                    },
                     {"type": "LineString",
                     "coordinates": [[50.1, 30.1], [49.9, 30.2]]
                    }]
 }
```

Features:

Most spatial data are not just ordinary shapes, rather these shapes (Geometry) represent something. Hence, features are a combination of geometry and the properties (or attributes) of a GeoJSON object. This means that every GeoJSON object whose “type” member is the value “Feature”, must have a “Geometry” and “Properties” member. However, if the feature has a commonly used identifier, it should be included as a third member with the name “id”.

```
{“type”: “Feature”,
“geometry”:{“type”: “Point”,
             “coordinates”: [50.1, 30.1]
}
“properties”: {“railway”: “switch”,
              “@id”: “4567892”
}
“id”: “node/4567892”
}
```

FeatureCollection:

This is an array of features and could be regarded as a ‘*MultiFeature*’*.* It must have the member “features” which contains an array of feature object(s).

```
{“type”: “FeatureCollection”,
“Features”: [{“type”: “feature”,
            “geometry”:{“type”: “Point”,
            “coordinates”: [50.1, 30.1]
```

```
                }
    "properties": {"railway": "switch",
                 "@id": "4567892"
                }
    "id": "node/4567892"
            }
            {"type": "feature",
            "geometry":{"type": "Point",
            "coordinates": [50.2, 30.0]
               }
    "properties": {"railway": "buffer_stop",
                  "@id": "4567893"
             }
    "id": "node/4567892"
           }
           }]
    }
```

GeoJSON uses "Points", "Lines" and "Polygons" to describe a single geometry, and uses "MultiPoint", "MultiLines", "MultiPolygons" and "GeometryCollection" to describe an array of Geometries that combine to form a single feature. All GeoJSON object has a "type" member; knowing what "type" the object is, enables an easy read of the data. With GeoJSON, there is no restriction on using only one type of Geometry as with some other GIS file formats like shapefile (shp). This flexibility makes it possible to aggregate different geometries in the same feature class.

2.3.2 ESRI Shapefile

Like GeoJSON, a shapefile also stores nontopological geometry and attribute information for the spatial features as a shape comprising a set of vector coordinates [38]. Shapefiles can support point, line, and area features (polygon). However, in contrast to GeoJSON, Shapefiles handle single features that either overlap or are non-contiguous [38]. This means that a shapefile could either be a point shapefile, line shapefile or polygon shapefile, but cannot contain two or more different geometry type in a single file (Figure 11).

The data in a standalone shapefile (.shp) cannot be meaningfully processed by most computer programs but has to be associated with an index file (.shx) and a dBASE file (.dbf) that holds the attributes of the shapes. Optionally, a project file (.prj) that stores the project information of the file could also be associated. Hence a shapefile is associated with the file types .shp, .dpf, .shx and .prj. A more technical description of shapefile and how the files are organised, are described in the ESRI Shapefile Technical Description [38].

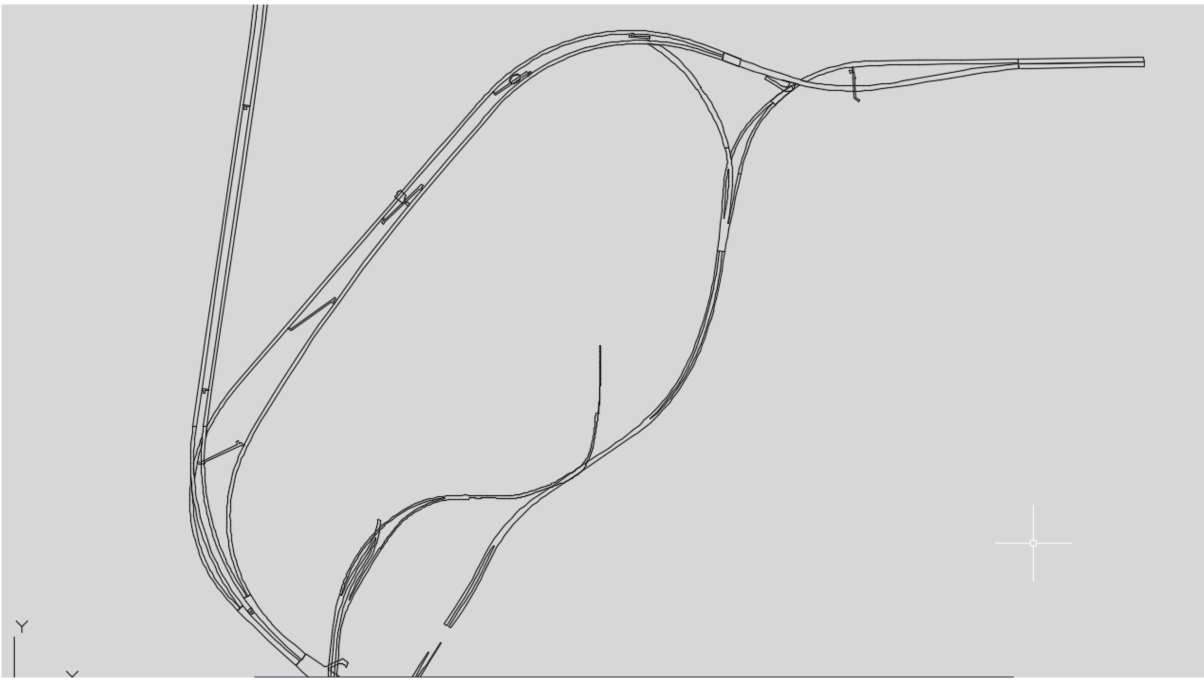


Figure 11: Shapefile made of "line"

2.3.3 Keyhole Markup Language (KML)

Open Geospatial Consortium (OGC) describes KML as an XML dialect used to encode and transport representations of geographic data for display in an earth browser, such as a 3D virtual globe, 2D web browser application, or 2D mobile application [39]. KML uses case-sensitive tag-based structure with nested elements and attributes. In 2008, KML became an international standard of the OGC [40] and has since then been supported by various softwares including AutoCAD Map 3D.

A standard KML file consists of: Placemarks, Descriptive HTML in Placemarks, Ground Overlays, Paths, and Polygons and is associated with the file types .kml and .kmz (KML zipped). Some advanced version of KML file can optionally have: Styles for Geometry, Styles for Highlighted Icons, Screen Overlays, and Network Links. Locations of elements in KML are specified with the longitude and latitude using the WGS84. There can also be additional values such as altitude, heading and tilt. KML is described in more details in [39] and [41].

2.3.4 MapInfo TAB

This is a file type used for MapInfo Software. Similar to shapefiles, it is made up of the files .TAB, .MAP, .IND, .DAT and .ID. TAB files link the associated ID, DAT, MAP and IND files; DAT files houses the tabular data associated as a dBase DBF file; MAP files store the geographic information; ID files are index files that link graphical objects to database information while IND file contains the index files for the tabular data.

2.4 Railway Infrastructure Data Sources

For independent research purposes and in simulating realistic automated train operations, it is necessary to make use of real data from available sources. This is usually challenging when seeking for railway infrastructure data as most IMs are confidential with their data or release them at high license fees. Some of the freely available data sources for railway infrastructure are described below.

2.4.1 OpenStreetMap (OSM)

OpenStreetMap (OSM) provides geological data free of charge and with less license restrictions. OSM was founded in 2004 in the United Kingdom [42] and focuses mainly on transport Infrastructure (such as rail, roads, rivers etc), however, a variety of other features such as buildings and administrative boundaries are available (Figure 12). It is also possible to navigate from one architecture level to another by simply zooming in and out.

OSM provides a platform not just to visualize and export an image of the visualised area (as with other web map sources such as Google and Microsoft Bing), but the geographic or infrastructure data in different formats like .osm and .geojson can also be downloaded. To export an image in PNG, JPEG, SVG or PDF format, the ‘sharing’ sidebar on the right is used. However, the raw OSM data can be downloaded using the ‘export’ button at the top left. This feature allows the user to download all the related data of a preselected boundary area in OSM XML format. There are other sources that can be used to access OSM raw data. To access the full database as one file in OSM XML or .pbf format, planet OSM is used. Geofabrik permits the download of pre-extracted data for continents, countries and in some cases, regions in .xml, .shp and .pbf formats. A third method is to make use of the overpassturbo interface. This allows for many possibilities especially when specific infrastructure data is required or needs to be filtered.

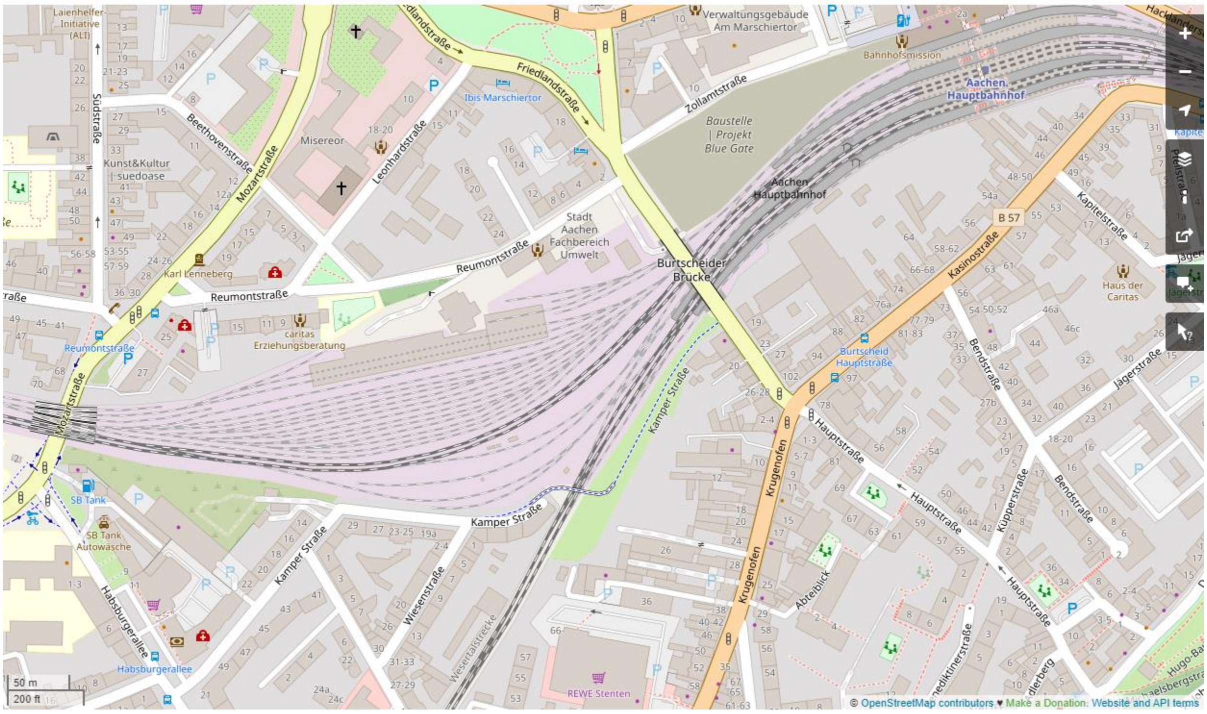


Figure 12: Geographic data from OSM (Source: OSM)

2.4.2 OpenRailwayMap (ORM)

OpenRailwayMap (ORM) is a detailed map of the worlds' railway network, based on data from OSM. The project was founded in 2011 and was then called "Bahnkarte" [43]. It aims to provide up-to-date in-depth details of the railway infrastructure. ORM is divided into three layers- Infrastructure, Maximum speed, and Signalling. In these layers, several railway features can be visualised which are not available in OSM. For example, in the infrastructure layer, it is possible in ORM to visualise the type of track (bridge, narrow gauge track, tunnel etc) and usage of the track (main line, high speed line, yard track, tram etc). The train protection systems can likewise be visualised in the signalling layer (see Figure 13) - blue lines represent the existing ETCS tracks, red and yellow lines represent the German Linienzugbeeinflussung (LZB) and Punktförmige Zugbeeinflussung (PZB) respectively, orange lines represents the Dutch Automatische Treinbeïnvloeding (ATB) or Automatic Train Control in English while black and grey lines represent areas with no train protection at all and areas where the information regarding the train protection is unknown respectively. Nevertheless, ORM is limited in its usage due to the present inability to download the raw data and subsequent conversion for use in diverse simulation applications.

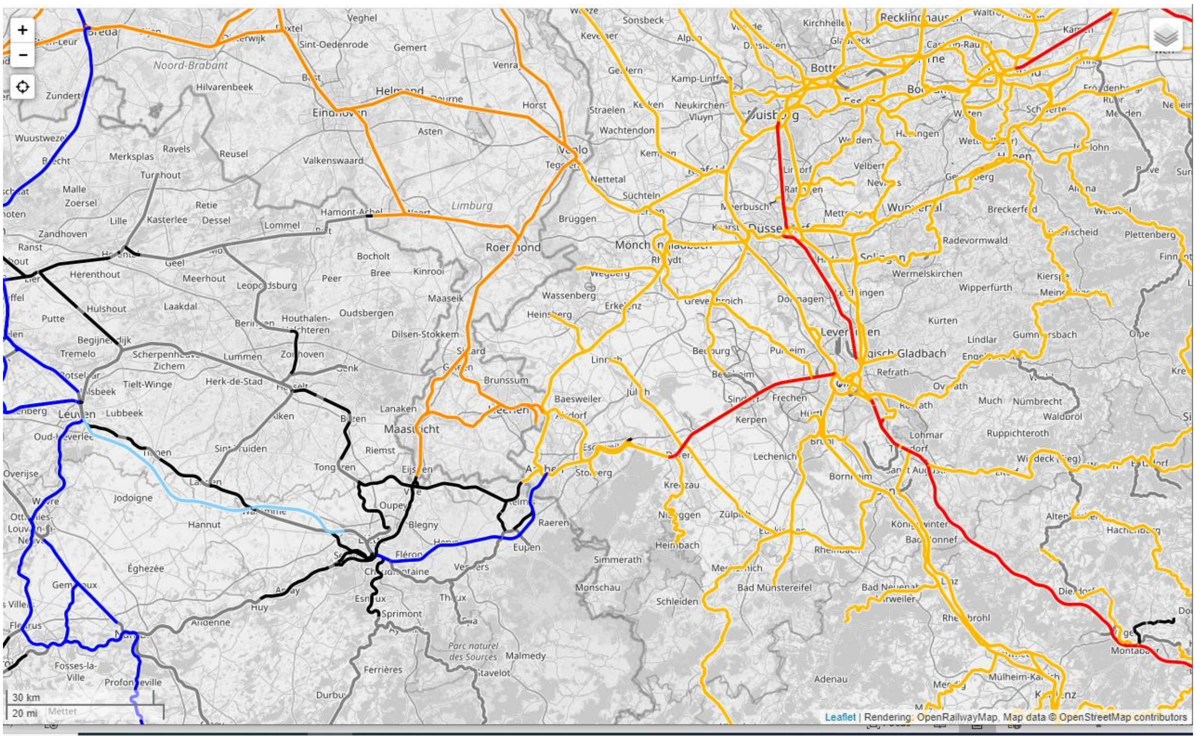


Figure 13. Map from ORM showing the various train protection systems (source: ORM)

2.4.3 Google Maps and Microsoft Bing

Google offers its own 2D map visualisation of the world geospatial information with several additional features such as 3D view, satellite view, live traffic condition, bicycle friendly routes, nature of the terrain, business contacts/organisations etc. To download the displayed data in pdf format, the print function on the menu button is used. Right-clicking on any location on the screen displays the coordinates of the point and an opportunity to measure the linear distance between two points. Google maps is very useful for navigation purposes because it can display feasible routes between two locations as well as an estimate of the time required for the transit; but in terms of railway infrastructure data, it is not very suitable for reasons mentioned in chapter 5.

Microsoft Bing is very similar to Google maps in terms of its features and functionalities and serve almost the same purposes.

2.4.4 Deutche Bahn (DB) Open Data

DB launched its own open data portal in November 2015 [44], where it makes data about certain aspects of the German railway available to the public in a machine-readable form and under a free license (creative commons licence). The portal is also being updated with more railway data, with the hope to improve data quality and further develop services [44]. Although not all railway data is being made available, this shows the willingness to finally take a step to make data open. Before the launch of the open data portal, huge usage fees had to be paid to access such data, including within the DB group where usage fees had to be paid by a part of DB in order to access and use data from another part of DB, further leading to bureaucracy, increased work, and sometimes, increased cost [45]. DB's initial reluctance for an open data could have helped to promote other sources such as OSM and ORM and made lots of volunteers to begin mapping the railway infrastructure on their own [45].

As of 2019, DB had 5 billion data elements in its open data portal under the creative commons licence (approximately 8 GByte) [44]. The mobility data available on DB open data portal comprises of:

A) Master data – Rail Network DB, station data, service facilities etc.

B) Business information

C) Logistic data – such as shipment data of containers and data from DB cargo

D) Target data – e.g. target timetable for long distance trains

E) Real-time-APIs – e.g. realtime condition of DB elevators and escalators, APIs to station masterdata etc

F) Data by further transport companies such as AVV, VRR, SNCF, KVV etc.

These data are available in various formats such as PDF, XML, ESRI shape, GeoJSON, TSV, GTFS etc. Some of the data are also made available in two or more different formats.

2.5 IRS 30100 – RailTopoModel (RTM)

The International Railway Standard (IRS) 30100 which is also called RailTopoModel (RTM) was published by UIC in September 2016 as an outcome of the European Railway Infrastructure Masterplan (ERIM) project, with the goal of providing a universal representation of a railway network and all events related to the rail sector [17]. This is intended to solve the problem that arise when different stakeholders in the railway sector use their own model to represent data. In other to come up with such a model, UIC focused on two major principles: a) Modelling network topology based on a connexity graph and b) using a systemic approach, independent of any particular use or application, thereby ensuring the durability and scalability of the model without compromising the integrity, quality, and dimensions of the data.

2.5.1 Modelling Principles

RailTopoModel offers an abstract view of the network as it describes the topology, geometry and positioning of the objects in relation to each other. RTM is based on graph theory (see 4.2) as far as topology is concerned. To describe a railway topology, two basic types of elements can be used namely: linear elements such as tracks and lines; and non-linear elements such as operational points. Traditionally, these elements would represent lines while the switches would represent nodes. However, this contradicts graph theory where nodes represent objects and edges represent relations between the nodes. Therefore, in the microscopic level, the railway topology is represented with the tracks (network elements) as nodes. Furthermore, the navigability is necessary to describe whether a train can pass from one network element to another and the feasible direction(s) of movement [23]. This makes it possible to show the feasible passage from track to track or line to line when passing an operation control point (ocp).

2.5.2 Architecture Levels

When dealing with or describing infrastructure data and railway topology, there are different levels of detail (or precision) that could be required based on the business need and/or objective as shown in Figure 14. As a rule of thumb, the bigger the area to be considered, the less details in the description and vice versa. For example, in capacity management or construction of a new line, a precise description of the network is not needed. However, in operations of timetables or interpretation of simulations, a more detailed information might be required. Therefore, as with most engineering problems, there is always a trade-off (in this case) in accuracy, quantity, and time. This trade-off is difficult to achieve with a single level of description. It is also possible and sometimes useful to combine several levels of description. For instance, if a bottleneck area along a network needs to be analysed, then a more detailed description of that area is needed. A detailed description of the whole network with highest precision could be counter-productive, especially while working with softwares since calculations are usually exponential in terms of the number of elements [46].

Implementation of RTM should include at least one level. There are three main architecture levels [23], but when needed, higher or lower levels of details are possible.

Micro level: This corresponds to the description of the network at track level and its associated elements such as switches, signals and buffer stops. It represents the network in a way that is close to the physical view. In this level, the non-linear elements are the objects (e.g. switches) on the track, while the linear elements are the tracks connecting them.

Meso level: This corresponds to the description of the network mainly at the operational point (also called operation control point) level, where several tracks are aggregated to form a station, junction etc. Here, the operational points (OPs) become the non-linear elements and the tracks connecting them are the linear elements. This level could be useful when trying to carry out capacity analysis at nodes or sections of lines.

Macro level: This is a broad description of the network at regional or national level, showing mainly the major operational points. In this level, several OPs could be aggregated to form a regional boundary describing only major OPs. These boundaries or OPs form the non-linear elements, while the lines connecting them form the linear elements.

Other useful levels that could exist include the **nano level** - which is a more detailed description of the network than the micro level, where the most minimal details of the properties of the rail network (e.g. the type of switches) are described, and the **corridor level** where only major boundaries along a route of the network could be described. In other words, each level can be aggregated to a higher level, or disaggregated to a lower level based on certain methodologies and rules as described in detail in [46].

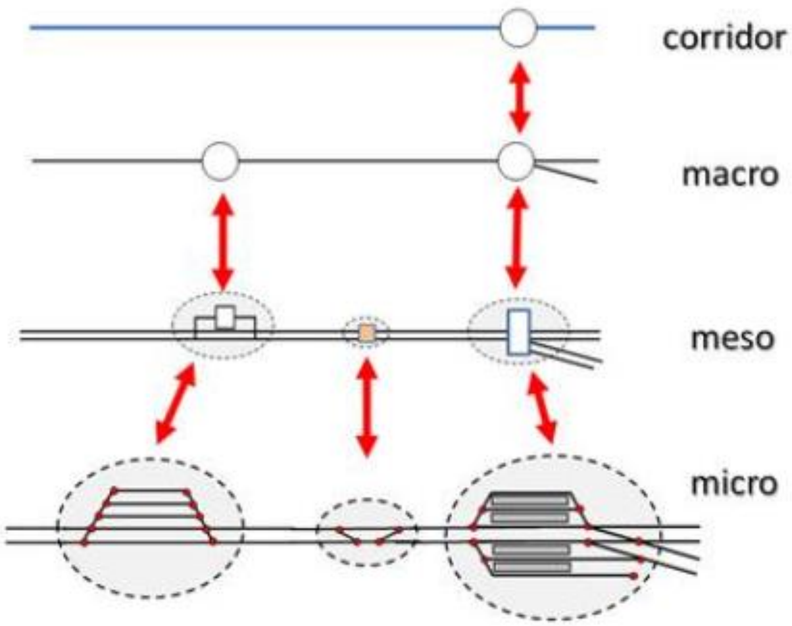


Figure 14: Various levels of description of railway line (Source: [25])

2.5.3 Brief Description of RTM Packages

RTM is described in UML notation further explained in 4.1. The modelling concepts are grouped within packages which consists of a collection of closely coupled modelling concepts and deals with a specific aspect of the model. There are four major packages: Base, Topology, Positioning System and Net Entities (Table 6).

Base package is depicted using a grey colour and contains mainly the Network and levelNetwork. It defines four properties:

- id: unique identifier
- name: natural designation of the object
- validFrom: date where the object is available for train operations. If empty, then it is valid till the validTo date

- validTo: date where the object is no longer available for usage. If empty, then it is valid since the validFrom date

shared by most objects in RTM. The class “Network” contains all the resources including all levels of description, that compose the network being considered. The “LevelNetwork” defines the resources that are required to define a certain level and therefore maintains a consistent level of granularity.

The **Topology** package is depicted with a yellow colour and contains mainly the NetElement, Relation and CompositionNetElement. This package applies graph theory concepts. The aggregation and consequently disaggregation on nodes, forming the different architecture levels is allowed by the class CompositionNetElement which defines an element that aggregates some other elements (in this case topological elements) from another level (e.g. aggregating macro elements to form a corridor).

Positioning Systems package is represented with a green colour and includes the PositioningSystem and IntrinsicCoordinate. It defines the various categories of positioning methods (linear, intrinsic and geographic) as well as the associate coordinates (linear or geometric) that can be used to spatially identify objects. Every network element must have an intrinsic positioning coordinate (with values from 0 to 1) because this defines its position in relation to a predefined reference element, whereas it is optional to use the linear or geographic positioning systems. In other to transform between the intrinsic locations and traditional (or absolute) locations, additional traditional coordinates (such as linear or geometric coordinates) must be defined.

Net Entities package is light blue in colour and contains the main classes LocatedNetEntity and EntityLocation. These classes allow to structure the functional description of the network objects that are not covered within the scope of topology. Such objects could be physical e.g. bridges, tunnels, switches, signals etc, or abstract e.g. speed limits, radio coverage area, type of control system etc. This package first associates the objects with their locations and these locations are then subsequently associated with the topology elements. This independence from the topology is a significant improvement from previous models where each entity was related directly to one (topology) element, leading to a duplication of the same entity (Figure 15) [47]. The location of an entity could either be a spot location (point locations), linear location (which defines information on the start and end point of the entity in reference to one or more element(s)) or area location (which is used to define a location when a set of net elements together represent an area of interest).

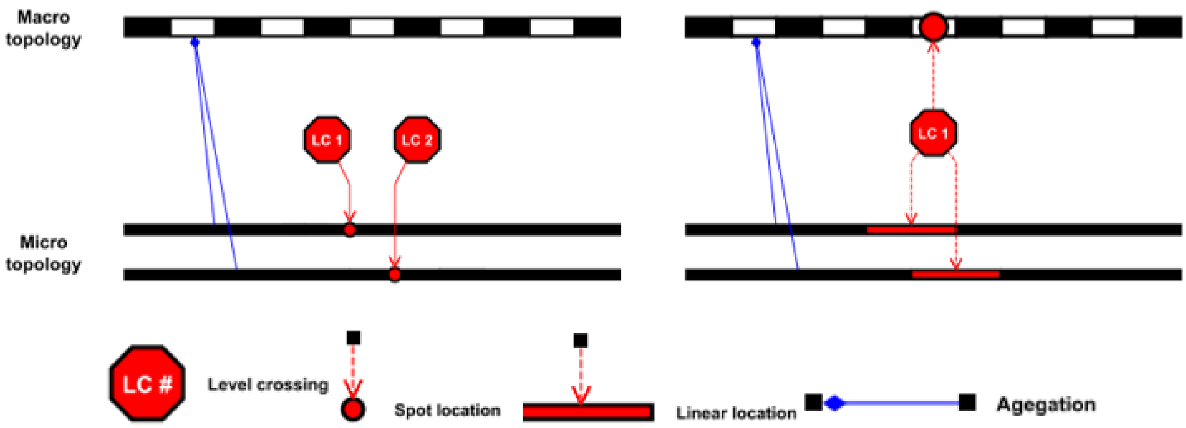
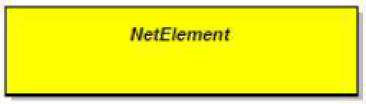
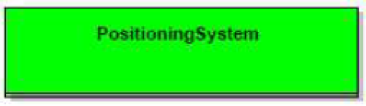


Figure 15: Comparison of railML 2 (left) and railML 3 (right) (Source: [47])

Table 6: RTM Packages (Source: [23])

Package	Colour code	Main element(s)
Base	Grey 	Network, LevelNetwork
Topology	Yellow 	NetElement, Relation, CompositionNetElement
Positioning Systems	Green 	PositioningSystem, IntrinsicCoordinate
Net Entities	Light blue 	LocatedNetEntity, EntityLocation

2.6 railway Markup Language (railML®)

railML® is an open, XML-based data exchange format for data interoperability of railway applications [22]. It is being managed by the railML.org initiative founded in 2002 with the primary objective of enabling a seamless communication between different railway applications (Figure 16). The initiative comprises of over 100 partners from different continents made up of IMs, Rus, software developers, authorities, research institutes and universities and consultancies. There have been several versions of railML, the earliest practical version being railML 1 and the latest being railML 3, wherein the concepts of RailTopoModel are implemented. One significant improvement in implementing railML 3 according to RailTopoModel remains that one object can be associated to different network elements and there is therefore no need for unnecessary duplications which sometimes can be confusing.

railML is thus far an industry standard and not a government standard. This means it is driven by the needs of the railway sector and its members following a bottom-up approach and not by a governmental norm which is usually developed by government-approved expert-groups and forced to be implemented in a top-down manner. [72]

There are four sub-schemas in railML 3 for the four most important data aspects in the railway system – Interlocking, Infrastructure, Rolling stock and Timetable [20]. The timetable sub-schema focuses on describing the railway timetable information and contains information on the operating periods, trains, paths and schedules of a trains, and rostering (rostering deals with a (cyclic) timetable for the drivers and operators of the trains). The rolling stock sub-schema is focused on describing all the necessary information related to the different rail vehicles. This information includes its characteristics such as its name, classification, car body features, propulsion system etc and its formations which describes the combination of different or similar vehicles to form a train set, or parts of a train set.

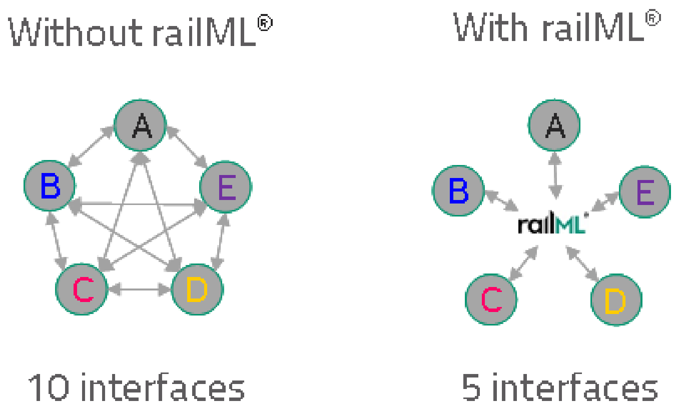


Figure 16: Number of interfaces with and without railML (Source: [66])

The infrastructure sub-schema describes the infrastructure that make up the railway network. It contains information on the topology, coordinates, geometry, infrastructure elements of the rail network and other entities that relate to the network such as gradients and speed profiles. The interlocking sub-schema is focused on describing the information necessary for signal planning and route locking along the network, therefore it re-uses most on the information in the infrastructure sub-schema.

2.7 Building Information Modelling (BIM), RTM and railML®

BIM is simply a digital form of construction and asset operations [1]. According to ISO (ISO/TS 12911:2012 (en)) [73], it is the process of managing information related to the facilities and projects (using shared digital representation of physical and functional characteristics of any built object, including buildings, bridges, roads, process plant) in order to coordinate multiple inputs and outputs, irrespective of specific implementations. This means that BIM is more of a “process” and goes beyond simply using softwares in the design/construction of infrastructures.

The extent of sharing digital representations is referred to as BIM maturity level. There are different BIM maturity models described in [75] however Figure 17 represents the BIM maturity model adopted in the UK. In this model, there are four levels of maturity explained in [74]. Level 0 is the least of all the stages wherein there is almost no collaboration. Information is produced and communicated using paper-based documents. Even though CAD drawings are utilized, but the model's information is not shared.

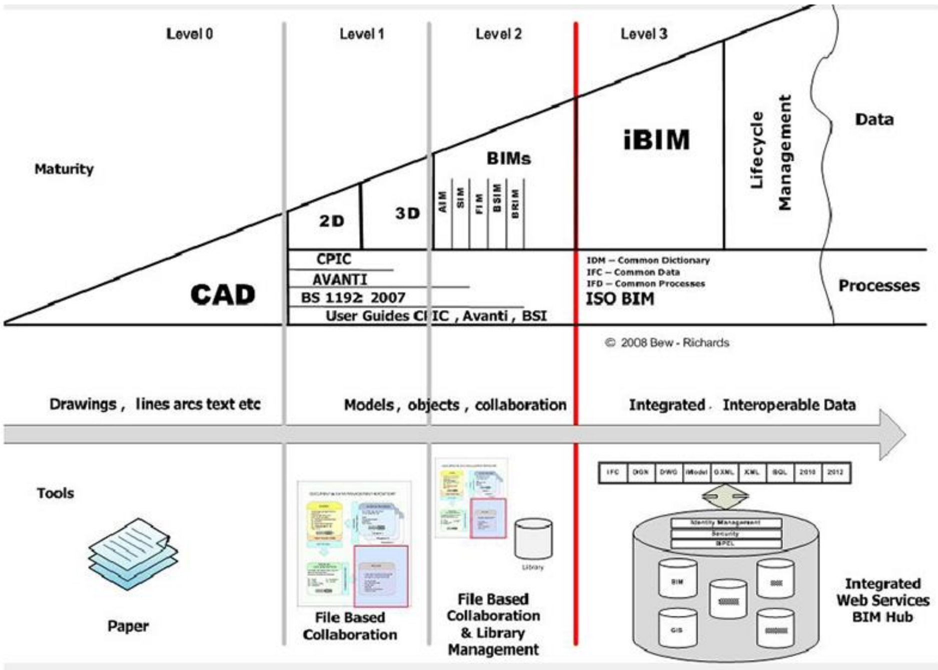


Figure 17: BIM Maturity levels (Source: [74])

BIM level 1 emphasises the move from CAD to 2D and 3D information and utilization of basic standards, however, the created models are not distributed among the many stakeholders, despite the presence of a common data environment. BIM Level 2 focuses on how information is communicated among the project's many members. Project teams use 3D modelling to develop their projects, produce and share information. At this point, two more project dimensions are introduced - the 4D, time management, and the 5D, budget computation. BIM level 3 is the final goal of the construction sector where there is a full integration of information in a cloud-based environment. This is made possible by utilising a common shared model that will be available to all the stakeholders of the project who can add or modify their own information.

The European Union (EU) introduced a “Common EU performance level” (Figure 18) which specifies the activities that should be consistently performed on a project for it to be considered an EU BIM Project. This is described in detail by EU BIM Task group in a Handbook for the Introduction of Building Information Modelling by the European Public Sector [1]. Some characteristics of the policy and technical definition areas include the application of the fundamental principle of avoiding over-generation and over processing of data; specification of information requirements for data in vendor-neutral, non-proprietary formats; and the basic principle of specifying, modelling and organising data should be object-oriented.

RTM and railML reduces the over-generation and over processing of data by setting standards for the generation of data (using RTM to generate data) and an interface through which various stakeholders can directly make use of the data without requiring many interfaces (using railML to store data). RTM and railML are also vendor-neutral (not specific to just one IM), free of restrictive licenses and makes use of object-oriented modelling. This makes them very helpful tools in the BIM process and digitalization in general.

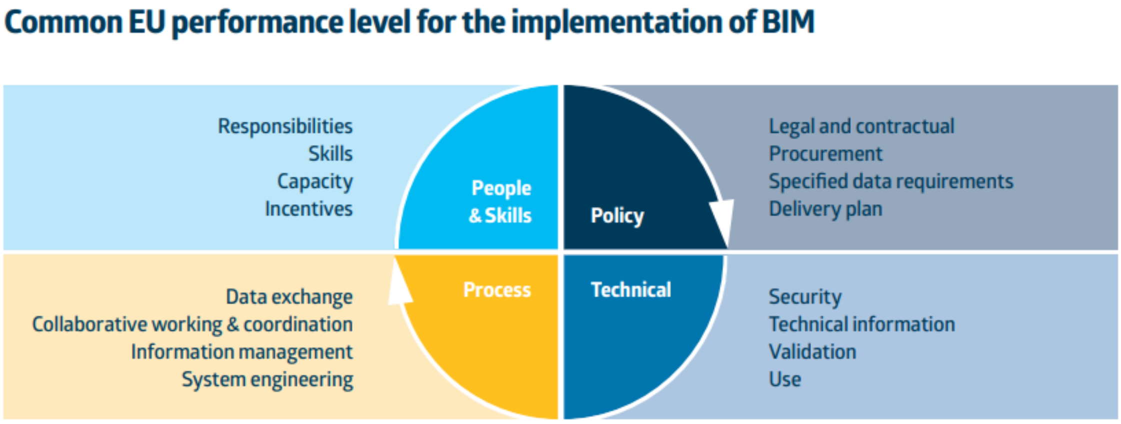


Figure 18: Common EU Performance level for the implementation of BIM (Source: [1])

2.8 railVIVID

railML stores data in text form and therefore relatively easy to understand, however there is still the need to validate and visualize the data. railVIVID was developed and presented by TU Dresden and railML.org as the railML visualisation and validation tool. With railVIVID, all kinds of data in railML could be checked for syntax correctness, visualised, and understood even without understanding the XML and railML syntax in which they were written. railVIVID provides a platform to have a graphic and tabular view of timetable data, view of the rolling stock data, topologic and geographic view of infrastructure data as well as a schema validation, provided the file was written in railML (Figure 19 and Figure 20). Only the railML 3 versions are unable to be visualised in railVIVID, nonetheless, validation is possible [48].

```
<infrastructure id="is1">
  <tracks>
    <track id="tr26301" code="2630" type="mainTrack">
      <trackTopology>
        <trackBegin id="tB26301" pos="0" absPos="91050">
          <openEnd id="oEb26301"/>
        </trackBegin>
        <trackEnd id="tE26301" pos="61366" absPos="152415">
          <macroscopicNode ocpRef="ocpFBGK"/>
        </trackEnd>
        <crossSections>
          <crossSection id="cs263011" pos="182" type="station" absPos="91232" ocpRef="ocpKKO">
            <geoCoord coord="50.350834 7.589025" epsgCode="urn:ogc:def:crs:EPSG::4326"/>
          </crossSection>
          <crossSection id="cs263012" pos="3303" type="station" absPos="94353" ocpRef="ocpKKOB">
            <geoCoord coord="50.323351 7.586081" epsgCode="urn:ogc:def:crs:EPSG::4326"/>
```

Figure 19: Excerpt of data written in railML 2.3

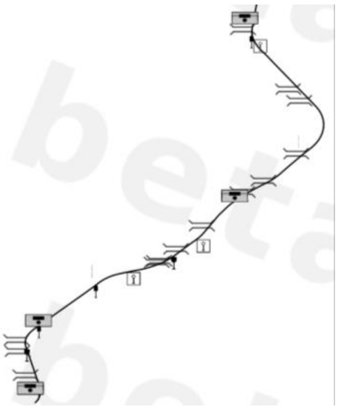


Figure 20: Excerpt of railVIVID visualization

2.9 Robotic Simulation

Implementing automated driving in railway entails that the rail vehicle must perform some operations or even drive on its own (depending on the GoA) in an unstructured, uncertain, and constantly changing environment. Such a rail vehicle is no different from a modern-day smart robot. A case study by DB Systel on driverless operation in the Paderborn maintenance depot utilized a driverless rail-road vehicle for smaller shunting duties and pushing freight wagons [10]. To prevent accidents and ensure safety, it is intended to install a clearance monitor on the vehicle [10]. These kinds of projects require a lot of testing to ensure reliability and safety, beginning on a small scale and then on a large scale before being completely rolled out.

In addition to physical testing, engineering design processes of robots can be hastened, made more economical, and benefit from more extensive testing by using simulation [49]. Simulation provides the following opportunities [49]:

- large amount of data can be quickly generated and at a low cost
- possibility a safe and fully controllable test and verification environment so that as many robot-robot and robot-human interactions can be tested without much complexity in the design and verification
- facilitate the development of more intelligent robots
- accelerating the design cycle and cost by reducing the need for manufacturing several prototypes before the acceptable version
- facilitates the understanding of human-robot-interaction (HRI) which reduces the risks to persons working in dangerous environments or with mobile robots

Robotic simulation is not free from challenges or barriers, some of which are [49]:

- developing a simulation platform for robots necessitates a wide range of multidisciplinary expertise and a long-term commitment to software development
- existing modelling languages are in their infancy at a time when robotics simulation ontology is emerging
- model calibration can be time-consuming
- challenges in determining the required level of model complexity
- sometimes the simulation is not fast enough

Simulations are done using softwares. Although software development poses considerable obstacles - the development of the software infrastructure and licensing being some of the key aspects - it is not a core research activity in the field of robotic simulations [49]. If simulation must be more useful in the field of robotics, there must be more research and attention given to the softwares used. Some of the softwares being used for simulation in robotics include Microsoft Robotics Developer Studio (MRDS), Webots, Roboguide, Gazebo, MotoSim, CoppeliaSim, Simple Two Dimentional Robot Simulator (STDR Simulator) etc.

Webots [50]**:** is a free open-source 3D robot simulation platform developed by Cyberbotics. It supports programming in C, C++, Java, ROS, MATLAB, and Python, and it runs on Windows, Linux, or macOS. In Webots, indoor and outdoor environments can easily be created. It has a large asset library which includes sensors, actuators, objects etc and it is possible to import existing CAD models (from Blender, Solidworks etc) as well as OSM maps.

Gazebo [51]**:** is another free open-source 3D robot simulator that supports the precise and quick modelling of robot populations in both indoor and outdoor contexts. It runs on Linux. It provides realistic rendering of environments including high-quality lighting, shadows, and textures. Gazebo has a wide range of sensors which can generate data with or without noise, and supports the development of custom plugins for robots, sensors, and environmental control. In gazebo, existing robot models (such as PR2, iRobot create and TurtleBot) can be used or new models can be created in other CAD softwares and imported.

There are two types of files involved in running a gazebo simulation: a world file and a model file. All the elements in a simulation, including robots, lights, sensors, and static objects, are contained in the world description file which has the `.world` extension while the model file contains only a single model. Model files help to simply the world file and facilitate reuse of models. Both files are formatted using the Simulation Description Format (SDF).

An SDF model is a collection of links, joints, collisions, visuals, objects, and plugins and can range from simple shapes to complex robots.

A link contains the physical properties of one body of the model and could have many collision and visual elements. A collision element is a geometry component that is used to check for collisions. Visual element is used to visualise parts of a link and could be zero or more. The link's dynamic attributes, such as mass and rotational inertia matrix, are described by the inertial element. A sensor gathers information from the outside world for use in plugins while a light element is a light source that is connected to a link. There may be zero or more sensors and lights in a link. A joint is used to connect two links. A plugin is a third-party shared library used to control a model and can either be loaded in the command line or specified in an SDF file.

CoppeliaSim [52]**:** is the continuation version of the discontinued V-REP. It is a 3D robot simulator which has integrated development environment. It supports a variety of features such as collision detection, minimum distance calculation, customizable particles, proximity and vision sensor simulation, etc., which makes it suitable for different applications including factory automation simulations, remote monitoring and safety double-checking. CoppeliaSim is versatile and ideal for multi-robot simulation, owing to its distributed control architecture. This means each object/model can be separately controlled - by an embedded script, a plugin, ROS, a remote API client, or some other custom solution. The controllers can be written in a variety of programming language such as C/C++, Python, Java, Lua, Matlab or Octave. CoppeliaSim supports the formats URDF, COLLADA, DXF, OBJ, STL, glTF, etc. It also offers a free version for educational purposes and can be run on Windows, macOS and Linux.

Simple Two Dimentional Robot Simulator (STDR Simulator) [53] [54]**:** is a free simple, flexible, and scalable 2D multi-robot simulator that runs on Linux (Figure 21). This simulator was developed with two main goals – to have a platform for easy multi-robot 2D simulation and to be ROS compliant. STDR supports functions such as: load a map or robot; delete a map or robot; move a robot; create a map, robot, or sensors; etc.

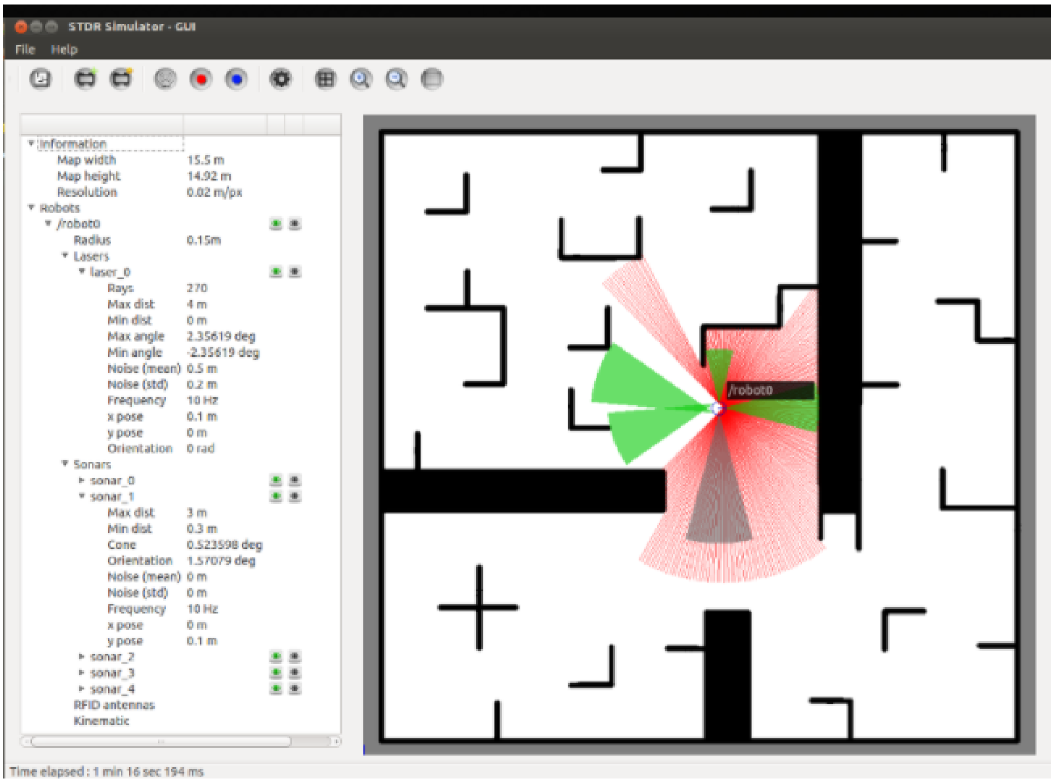


Figure 21: STDR GUI with one loaded robot, equipped with a laser and five sonar sensors (Source: [53])

3 Motivation

The benefits of automated driving in railway cuts across all sectors from passenger operation (metro, mainline, etc) to freight transport and most significantly, shunting yard operations. However, to roll-out automated driving on a large scale and harness these benefits, several corresponding challenges still need to be addressed. One major aspect is the effective communication of data because this aspect deals a lot with safety. For independent research and simulation purposes, a fairly accurate information on the railway infrastructure data is needed.

There are presently different types of formats and models used by the various groups that make up the railway industry. This lack of a uniform standard poses a challenge for the future development of the industry, especially while driving digitization and automation. Furthermore, the different formats lead to many incompatibilities and inconsistencies, thereby impacting negatively on duration, quality and cost of railway operations. Hence, the standard IRS 30100 was developed by UIC.

railML 3 represents one the most widely used implementation of the RailTopoModel, nevertheless, several railway data, especially publicly available data, are not in this format. Unfortunately, these public data sources represent a greater percentage of the freely available data. Thus far, these data can be downloaded in different formats except the railML 3, thereby requiring many interfaces while trying to exchange.

The main goal behind this work is to examine the possibility of accessing the railway infrastructure raw data from publicly available sources and conversion to the RailTopoModel using railML. These sources could be lacking certain information regarding the infrastructure and might significantly impact the useability of the data for different applications/ use cases. So an analysis of the various sources is needed to identify which method is preferable in terms of accuracy and ease. Finally, because the infrastructure data is needed for simulation, it is necessary to also examine if there is a possibilty to import to gazebo simulation software, without building the track from the scratch.

4 Methods

4.1 Unified Modelling Language

Models are important for software and non-software systems because they help to manage complexity, detect errors and omissions early, communicate with stakeholders, understand requirements, drive implementation, and to understand the impact of change [55]. For most systems, modelling languages were developed to define the systems using consistent set of rules. The language could be textual or graphical [56].

RailTopoModel is described using the Unified Modelling Language (UML) notation [23], which is a general-purpose graphical modelling language that uses graphs/ shapes for descriptions. UML provides a standard way to visualize the design of systems. There are different diagrams in UML which is a consequence of the different viewpoints that a system (or software) may have, depending on the stakeholder involved. Different stakeholders might be interested in different information or detail; however, UML tries to provide a modelling language, such that all stakeholders can benefit from at least one diagram [57].

A UML diagram is a graphical representation of the parts of a UML model, which consists of elements such as packages, classes, and associations [58]. Each diagram has a contents area and may have a frame and a heading as seen in Figure 22.

Figure 22: UML Diagram

The contents area contains the graphical symbols while the heading represents the kind, name, and parameters of the namespace enclosing or the model element that owns the elements that are described by the graphic symbols in the content area. There are two major types of diagram types shown in Figure 23, however, several diagram types can be mixed, such as when showing a state machine (behaviour diagram) nested inside an internal structure (structure diagram) [58].

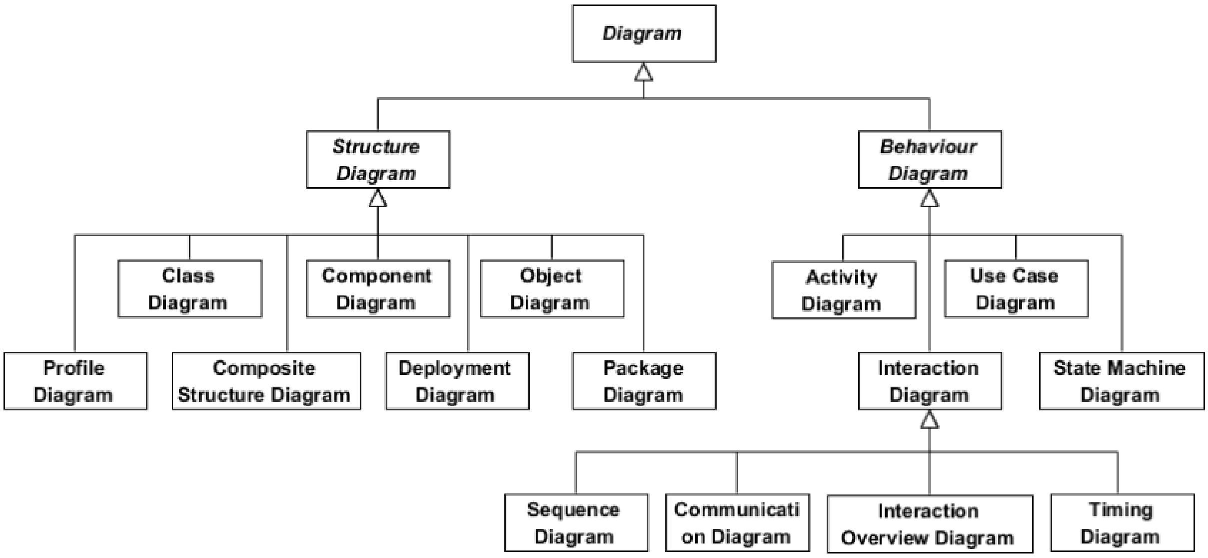


Figure 23: Taxonomy of structure and behaviour diagrams (Source: [57])

Structure diagrams show the static structure of the objects in a system (i.e. they depict those elements in a specification that are irrespective of time). Behavior diagrams show the dynamic behavior of the objects in a system, including their methods, collaborations, activities, and state histories. The dynamic behavior of a system can be described as a series of changes to the system over time [58].

RailTopoModel is a structure diagram grouped within four packages and all its concepts are depicted as UML classes. This means the diagram primarily provides a means of describing the static state of the railway infrastructure topology. In combination with railML, the dynamic states of aspects of railway such as real-time train run data could be described [59].

Package diagram (Figure 24) is a kind of structural diagram that describes the arrangement and organization of model elements in middle to large scale projects. Its ability to show both structure and dependencies between sub-systems or modules [57] makes it possible to achieve different description levels when modelling railway topology. Package diagrams are helpful in simplifying complex class diagrams because they can group classes into packages.

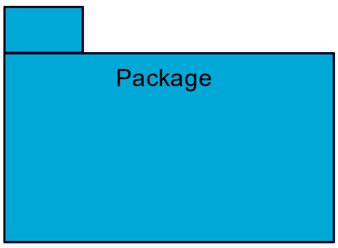


Figure 24: Shape of a Package

Class diagram describes the static structure of a system by showing the classes, attributes, operations (or methods) and the relationships among objects of the system [57]. In other words, it consists of a set of classes and a set of the relationships between the classes. A class notation is made up of the three parts (Figure 25): class name (which appears in the first partition), class attribute (which defines what the objects of the class "know" and appears in the second partition) and class operation or method (which defines what the class "can do" and are shown in the third partition where available).

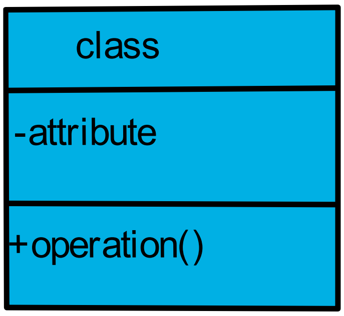


Figure 25: Shape of a class

The different class relationships are described in Table 7

Table 7: Graphical representation of class relationships

Class Relationship	**Graphical Representation**
Association: encompass any type of logical connection between classes. This relationship can be directed (dependency) – B can call A's properties or methods but not vice versa or undirected (reflexive) – B can call A's properties and vice versa	B A Reflexive B  A Dependency
Inheritance: shows that a class 'is a' child of another class and inherits its features. B is a child of A and inherits its features.	B  A
Realization: shows that a class implements the functionalities or methods specified in another class (called the interface). B implements A	B  A
Aggregation: shows that a class is formed by building up 'parts of' another class. A can exist without B	B  A

Composition: is similar to aggregation, except that it shows a relationship such that when the whole class is destroyed, the parts are also destroyed. A cannot exist without B.	

Another aspect of class relationship is cardinality, which represents the size or number of values that can be contained in a collection [58]. It could be one (1), zero or one (0.. 1), zero or more (*) or one or more (1.. *) and written at the head of the connection which represents the relationship.

The class diagram of RailTopoModel (Appendix **1**) makes use of four (composition, aggregation, directed association and inheritance) out of the five described relationships.

4.2 Graph Theory

Graph theory is the field of mathematics that study graphs. A graph consists of a set of vertices (sometimes called nodes) and edges (sometimes called links, lines, relations, or arcs) [71]. The vertex represents an object that has at least one quantifiable property (e.g., weight, length, colour, height etc) while the edge represents the connection between the vertices. For example, to describe the network between a house and a school in graph theory, the house and the school would be represented as nodes, while the roads that connects the house to the school would be represented as edges.

Simple graphs only shows that there is a possible connection between two nodes and therefore do not contain multiple edges (Figure 26), whereas multi-graphs show the complete network with all possible connections between the nodes and therefore contain multiple edges (Figure 27) [71].

Figure 26: Simple Graph

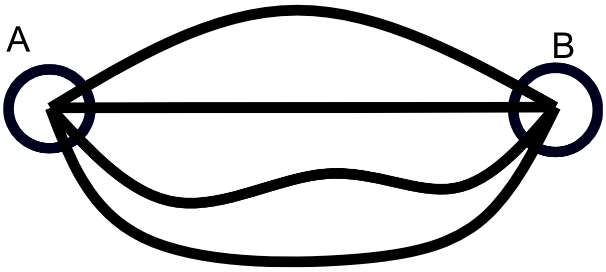


Figure 27: Multi-graph

A graph can also be directed (Figure 29) or un-directed (Figure 28) [71]. An un-directed graph is an un-ordered set of nodes and as such its arrangement makes no difference. In contrast, a directed graph is an ordered set that have directions associated to the edges which is indicated by arrows.

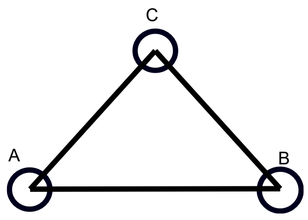


Figure 28: Undirected graph

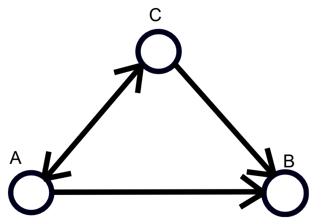


Figure 29: Directed graph

4.3 Great Circle Distance

Using Pythagoras theorem, the shortest distance between two points can easily be calculated. However, on the surface of the earth, the shortest distance is not a straight line, but part of a great circle (a circle whose plane passes through the centre of the earth) [60]. This means that in a 2D map of the world, the distance between two points would not be the same as the real distance on the earth-surface. For the Linear referencing method in RailTopoModel, the absolute or relative location is commonly used. Absolute location uses one of different kinds of independent pieces of information (such as coordinates, address, unique identifiers etc) to locate an object as a separate, identifiable entity whereas a relative location uses several interdependent pieces of information to locate objects to their relative geographic position [61]. Relative location requires a known point, the direction, and distance from the known point to specify a location and is the basis for many Geographic Information Systems (GIS) [61].

Most formats for geographic data from the public sources provide information on the position or location of an object using only its longitude and latitude coordinates, making the calculation of the great circle distance a necessary step for the linear referencing method.

Three methods that can be used to estimate the linear distance between two points on the earth surface are described below. All the described methods are not perfectly accurate when compared with the distance from Global Positioning Systems (GPS). This is because the earth is not a perfect sphere rather an oblate sphere, meaning it is a bit flatter at the equator than at the poles due to its rotation [60]. This results in a slightly larger radius at the equator when compared with the radius at the poles [60]. However, for simplification, it is assumed in this work that the shape of the earth is a sphere, and the average radius of the earth is 6371 kilometers.

4.3.1 Northing and Easting Method

This method uses the angles subtended by the difference between the longitudes and latitude of two points to calculate the respective longitudinal and latitudinal linear distance and then the Pythagoras theorem is applied to get the distance between the two points [70].

Figure 30 represents two points A and B located at certain points on the earth. The blue and red dotted lines represent the lines of longitude 0° (prime meridian) and latitude 0° (equator) respectively.

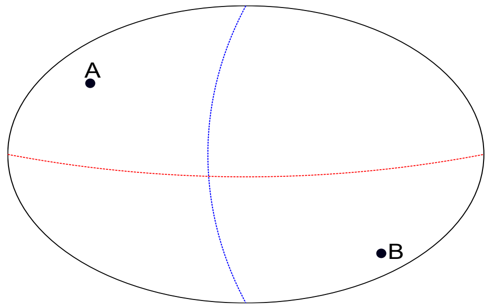


Figure 30: Earth surface showing prime meridian and equator

Figure 31 and Figure 32 shows how to calculate the linear distance between these two points using the northing and easting method. A line is drawn from A and B to the centre of the earth O. Arc AC represents how far north Point A is from point B with Θ_{NS} as the angle subtended at the centre of the earth, while arc BC represents how far east point A is from point B, with Θ_{EW} being the angle subtended at the centre of the earth. If the length of the arc AC is d_{NS} and the length of the arc BC is d_{EW}, the great circle distance d is calculated using Pythagoras theorem.

$$d_{NS} = \frac{\Theta_{NS}}{360°} 2\pi r_E$$

$$d_{EW} = \frac{\Theta_{EW}}{360°} 2\pi r_E$$

$$d = \sqrt{d_{NS}^2 + d_{EW}^2}$$

Θ_{NS}: difference of the latitudes of the two points in degrees

Θ_{EW} : difference of the longitudes of the two points in degrees

r_E: average radius of the earth

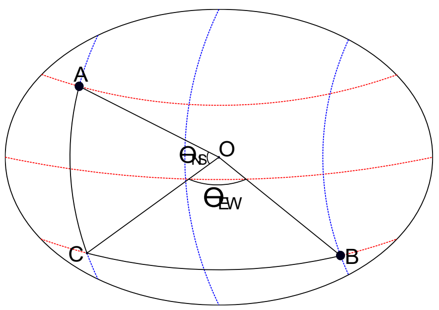


Figure 31: Linear distance between two points on the earth surface

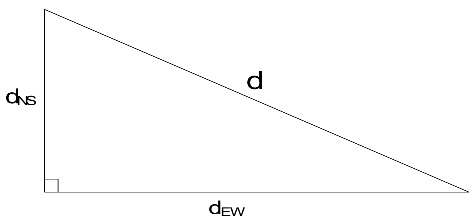


Figure 32: Pythagoras theorem

Over long distances, this method is significantly inaccurate compared to the real distance as would be obtained from a GPS system. This is because flattening a sphere on a 2D paper will result in some distortions.

4.3.2 Haversine Formular Method

The haversine (half of a versine (Figure 33)) is a trigonometric function useful in navigation. The haversine formular uses longitude (λ) and latitude (φ) to directly calculate the great circle distance between two points [62]. Figure 33 depicts the earth as a sphere with radius r (r = r_E) while d is the length of the arc AB (great circle distance) subtended by the angle θ.

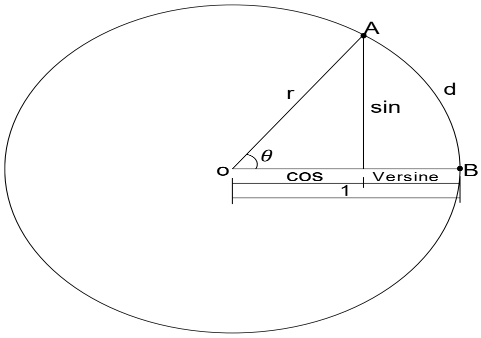


Figure 33: Haversine formula method

$$\theta = \frac{d}{r} \tag{1}$$

versine = 1 – cos (θ)

Haversine (half of versine) = $\frac{1-\cos\ (\theta)}{2}$ = $\sin^2\frac{\theta}{2}$

$$Let h = hav\ (\theta) = sin^2\frac{\theta}{2} \tag{2}$$

$$\Rightarrow\ \theta\ =\ archav(h) = 2.arcsin(\sqrt{h}) \tag{3}$$

Haversine formular is written as [62]

$$hav\ (\theta) = hav\ (\varphi_2 - \varphi_1) + cos(\varphi_1)\,cos(\varphi_2)\,hav(\lambda_2 - \lambda_1) \tag{4}$$

Where, φ_1, φ_2 are the latitude of point A and latitude of point B respectively in radians,

λ_1, λ_2 are the longitude of point A and longitude of point B respectively also in radians.

Substituting (3) into (1), the great circle distance d is given as:

$$d = r.archav(h) = 2r.arcsin(\sqrt{h}) \tag{5}$$

From (2) and (4), (5) can then be written in a more general form as

$$d = 2r.arcsin\left(\sqrt{hav\ (\varphi_2 - \varphi_1) + cos(\varphi_1)\,cos(\varphi_2)\,hav(\lambda_2 - \lambda_1)}\right) \tag{6}$$

$$= 2r.arcsin\left(\sqrt{sin^2(\frac{\varphi_2 - \varphi_1}{2}) + cos(\varphi_1)\, cos(\varphi_2)\, sin^2(\frac{\lambda_2 - \lambda_1}{2})}\right)$$

4.3.3 Central Subtended Angle Method

This method applies the spherical law of cosines to calculate the angle subtended at the centre [62]. In Figure 34, points A, B and C form a triangle. If a, b and c are the lengths of the sides opposite A, B and C respectively, and A, B and θ represent the angle subtended at points A, B and C respectively, then the spherical law of cosines can be written as [62]

$$cos\ (\theta) = cos\ (A) cos\ (B) + sin(A)\, sin(B)\, cos(c) \quad (7)$$

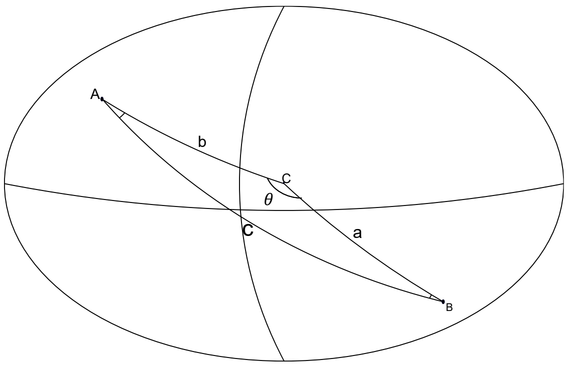


Figure 34: Central subtended angle method

If φ_1, φ_2 are the latitude of point A and latitude of point B respectively in degrees and λ_1, λ_2 are the longitude of point A and longitude of point B respectively also in degrees, then on the surface of the earth,

A = 90°- φ_1 and B = 90°- φ_2

(7) is then re-written as

$$cos\ (\theta) = cos\ (90 - \varphi_1) cos\ (90 - \varphi_2) + sin(90 - \varphi_1)\, sin(90 - \varphi_2)\, cos(\lambda_2 - \lambda_1) \quad (8)$$

But $\cos(90 - x) = \sin\,(x)$ and $\sin(90 - x) = \cos\,(x)$

Therefore, from (8), the central subtended angle on the surface of the earth becomes:

$$cos\ (\theta) = sin\ (\varphi_1) sin\ (\varphi_2) + cos(\varphi_1)\, cos(\varphi_2)\, cos(\lambda_2 - \lambda_1) \quad (9)$$

The great circle distance d, is then calculated using the equation: $d = \frac{\theta}{360°} 2\pi r_E$

The haversine formular is a derivative of the law of cosines [62].

5 Modelling of Obing Railway Station in railML following RTM

To model in railML following the RTM, three basic details are needed – elements of the network, referencing method (which makes it possible to position items on the elements) and the topology (which defines the connections (or relationships) and navigability between the elements). The available data must be in a format that these basic details can be read before it can be modelled in RTM.

To begin, a layout of the station with all the necessary elements is needed. To obtain this layout, one can look to the publicly available maps on the web to obtain a pictorial view of the station. The most convenient available web map is the ORM because it can display only the railway data. To do this, in the layer menu, “no background map” should be selected. Microsoft Bing and Google maps (Figure 36 and Figure 35 respectively), can also provide information related to the railway elements, but it does this alongside other infrastructures such as streets, stores, rivers, parks, banks, hospitals etc. The required layout of all the tracks and how they connect each other is only possible at the microscopic (or lower) description level. However, in small stations which may contain some discontinued tracks for public transport - such as Obing station - Google maps displays only the lines of railway (Figure 38) and is therefore not sufficient for a microscopic view of the rail network while Microsoft Bing does not provide a detailed view of the tracks for both large and small stations (Figure 36 and Figure 37). Network entities such as platforms, signals, level-crossings, buffer stops, (ETCS) balise, speed limits etc are also not displayed in both sources, although they are not basic for describing the topology.

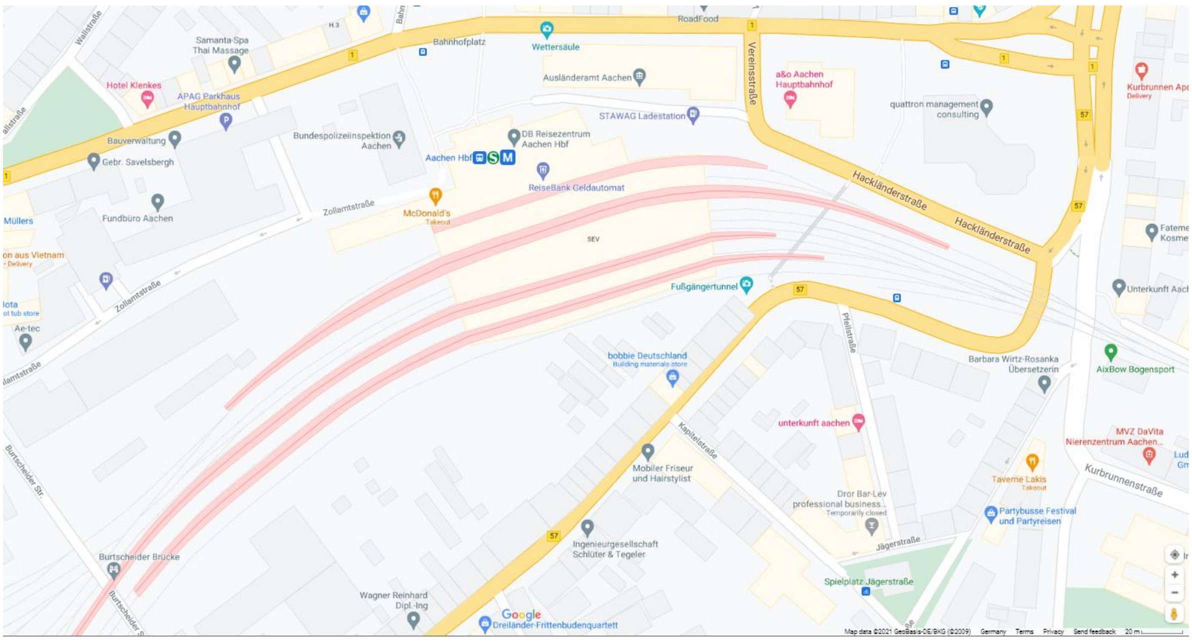


Figure 35: Google maps (Source: Google maps)

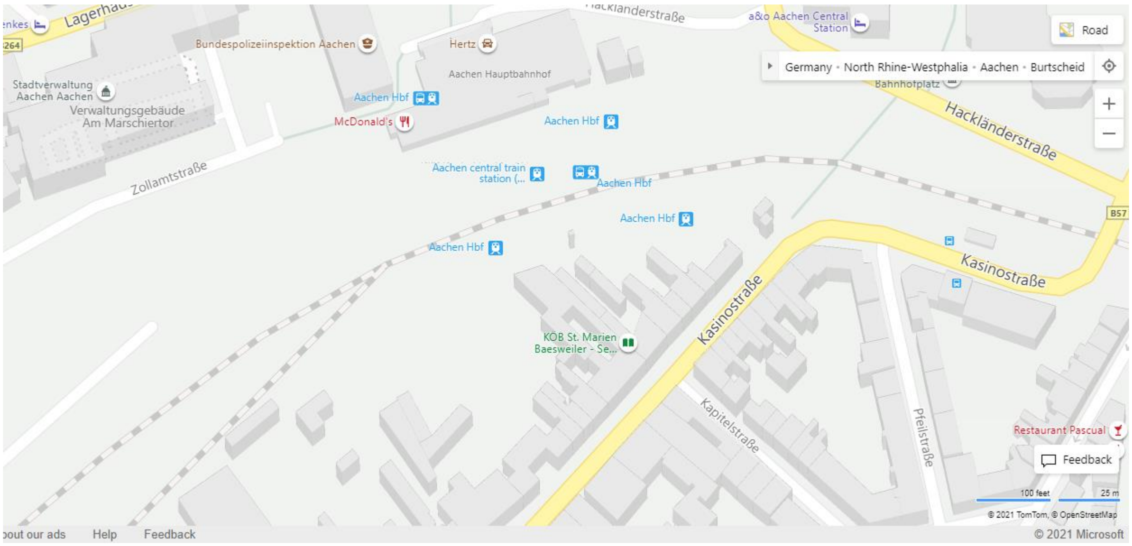


Figure 36: Microsoft Bing (Source: Microsoft Bing)

OpenStreetMap can display the tracks (including discontinued tracks) that make up a railway station, but unlike ORM, it cannot filter out all other details on the display. The shapefile containing only the railway data in OSM can as well be downloaded from Geofabrik. The challenge with this method is that one must know in advance, the exact location of Obing (or any other desired location). This is because there is no “search” function with a shapefile which displays only one type of geometry and cannot display texts. Another shortfall is that to obtain the data for only Obing, the complete data for the folder containing the location of Obing must be downloaded which is quite large. A visual comparison then must be made between the .shp file and another web map source (preferably OSM or ORM) on the possible location of Obing, implying that it is very difficult (if possible) to independently use Geofabrik to access the data for a specific location or station. When Obing is located, all other irrelevant data part can be manually deleted. The advantage of this method is that several CAD softwares can directly read .shp files, which means that the geometries describing the railway can easily be gotten without writing a script, unlike while using overpass-turbo. The layout of the network is then simply a schematic linear representation of the tracks and its elements. Figure 39 represents the layout of Obing station as viewed from ORM.

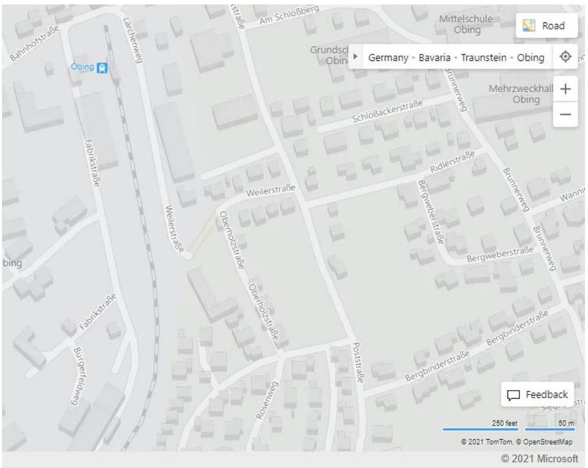


Figure 37: Obing Area from Microsoft Bing (Source: Microsoft Bing)

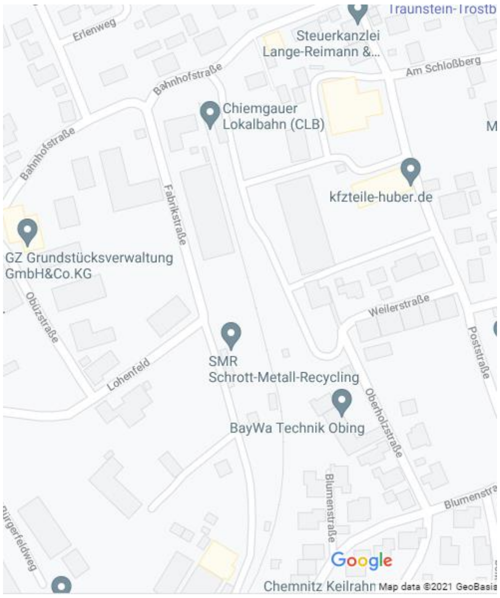


Figure 38: Obing Area from Google Maps (Source: Google maps)

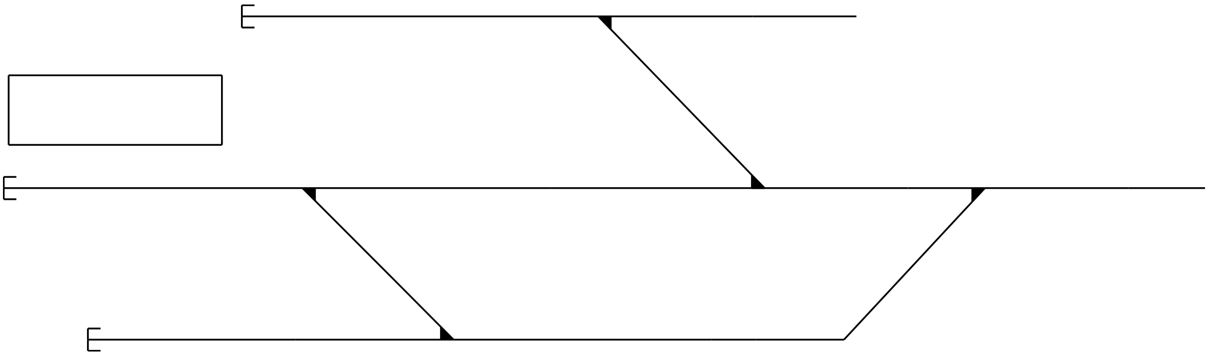


Figure 39: Layout of Obing

Although the web sources provide visual information on elements of the network, they do not provide any information on its positions or dimensions which are necessary in RTM. Therefore, more than just the visual representation from web sources is needed. Hence, the need for the raw data that describes the station. The raw data and map in ORM can be obtained by a query written in overpass-turbo using the necessary tags (**Appendix 2**). The data can be downloaded in GeoJSON format (see **Appendix 3**). In this format, it is easy to read the various geometries that describe the network elements and entities alongside their geometric coordinates (in longitude and latitude) and attributes.

Having obtained the raw data, we have sufficient information to model in RTM. Nevertheless, in other to include the linear coordinates in the associated positioning system, the linear distances have to be obtained from the geometric coordinates. Any of the described methods for great circle distances can be applied to obtain the linear distances of the elements in Obing station. The reason is that the distances are very small and even the northing and easting method would give significantly accurate values. But for longer distances, northing and easting method should not be used. A simple set of formulars in Microsoft excel (or any other programming language) can be used to convert the geometric coordinates to linear distances in kilometers using the haversine or subtended angle method by simply inserting the values of the longitude and latitude. Alternatively, some websites (such as [63] [64]) can be used to make the same calculations by simply inputting the longitude and latitude values in decimal degrees or degrees/minutes/seconds.

The detailed layout of the station Obing with its dimensions can now be schematically represented as in Figure 40. Below the layout, the first set of numbers represent the linear coordinate system in meters written at the left side of the projected line from node. Below it is the geometric coordinates with longitudes at the left and latitudes at the right. In the last row, a name is assigned to each node representing a network entity at the left, and in the right is its id as is in ORM.

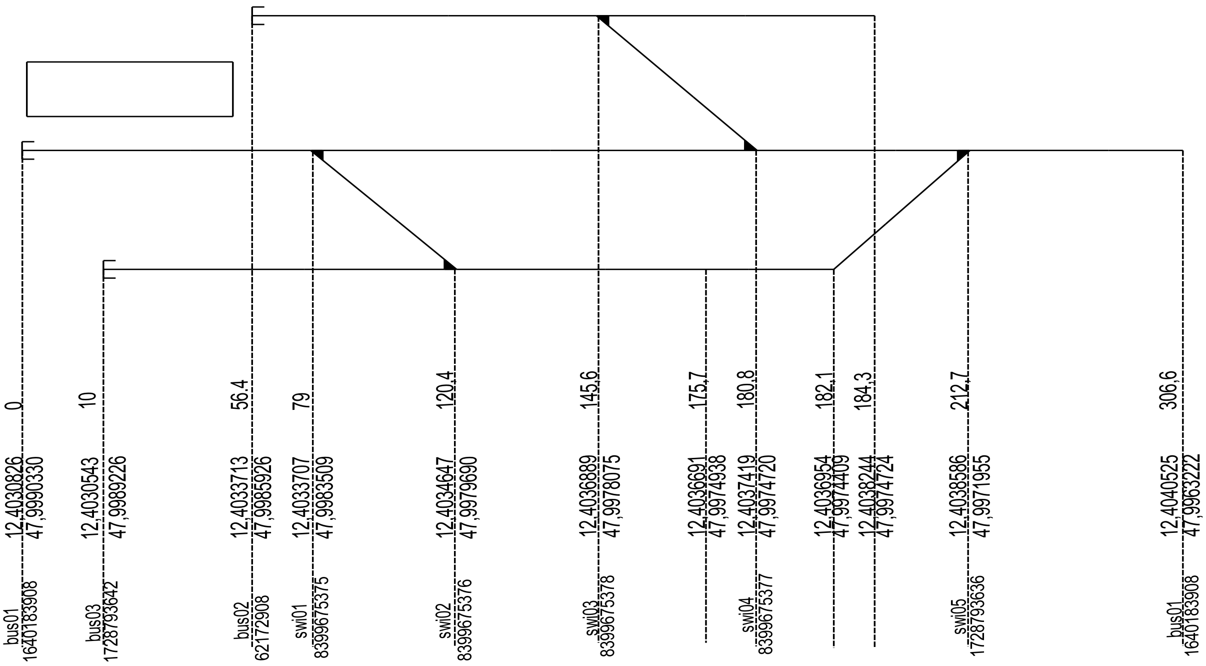


Figure 40: Schematic layout of the station Obing with its dimensions

The next step is to generate the topology following RTM. The topology for Obing would be described at microscopic level only. To go to a higher description level (meso), two OPs are needed. However, the elements and relations of the microscopic topology can easily be aggregated to form the mesoscopic topology. In describing the microscopic topology, only the tracks and connections between them are relevant. To do this, graph theory is applied. This means that the nodes are the tracks, and the edges are mainly the switches or level-crossings which connect the nodes. In ORM, the tracks are usually identified with the tag “ways”. Normally, tracks in a railway station are divided in such a way that they begin from a network element along the line and end in the next network element [65]. For example, a track can begin from a switch or buffer stop and end in another switch or buffer stop. But in ORM, these tracks are divided in such a way that they sometimes do not follow this convention. Hence, we can have a “way” that begins from a switch or buffer stop and extends through one or more switches; and we could also have two or more “ways” that are in between two switches or buffer stops (the latter can be seen between swi02 and swi05 in Figure 40). It is better to make use of as many “nodes” as possible when modelling the topology. This is because, in reality, railway tracks have sections that are curved which cannot be described in most GIS data formats as a curve. GIS data formats described in 0 only support straight lines, therefore a curved section of track would be described in GeoJSON as a multiline with many nodes, whereas a more accurate description would be a single track, its coordinate, and its radius. It is possible to ignore the node division used in ORM and model the topology following the normal convention, however, this would mean that several nodes (with their coordinate attributes) would be lost when the conversion to railML 3.1 is completed. These losses significantly jeopardize the accuracy of the converted data.

Therefore, the topology for Obing described in Figure 41 would include nodes (which represents tracks on a microscopic level) which may not have any switch at both ends.

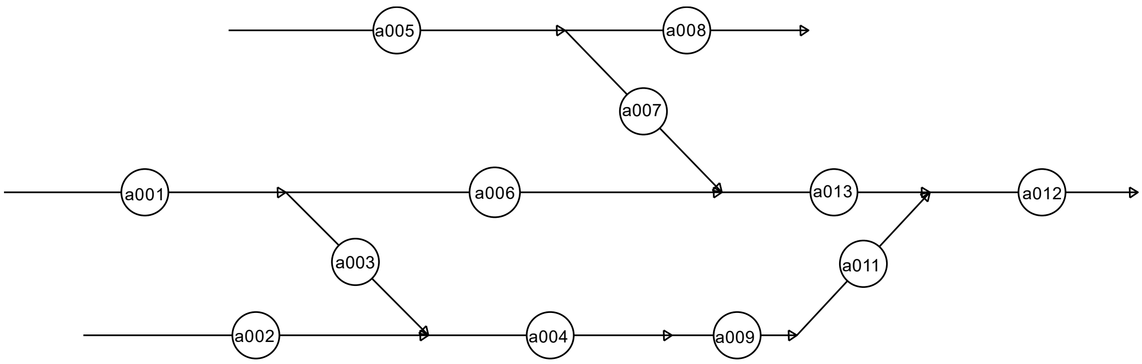


Figure 41: Topology of Obing

Figure 40 and Figure 41 of Obing will form the basics for all other steps in converting the railway data of Obing to RTM.

RTM shall be implemented using railML 3.1 format. A learning tutorial on the implementation of the infrastrucuture part of railML is avaiable on railML.org [65].

5.1 The file skeleton:

follows the same framework structure based on XML syntax. The railML version is also included as an attribute. The code will look thus:

```
<?xml version="1.0" encoding="utf-8"?>
<railML xmlns="https://www.railml.org/schemas/3.1"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:gid="http://www.gpsinfradat.de/schemas/gid"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="https://www.railml.org/schemas/3.1 https://www.railml.org/schemas/railml-
```

```
3.1/railml3.xsd"
      version="3.1">
        ...
</railML>
```

The next element is a <metadata> wherein the source of the railML file, author, copyright, timestamp etc is defined. The code snippet is:

```
<metadata>
  <dc:format>3.1</dc:format>
  <dc:identifier>2</dc:identifier>
  <dc:source>The data behind this model is from www.openstreemap.org and generated by overpass-
turbo.eu</dc:source>
  <dc:title xml:lang="en">Obing Station in railML 3.1</dc:title>
  <dc:language>en</dc:language>
  <dc:date>2021-05-10T14:35:01Z</dc:date>
  <dc:creator xml:lang="ig-NG">Chinenye Azubuike</dc:creator>
  <dc:description>This file has been coded manually and therefore may not be free of errors. Several
assumptions are also made in this file when necessary</dc:description>
  <dc:rights>Copyright (c) Chinenye Azubuike, Institute of Rail Vehicles and Transport Systems (ifs) RWTH
Aachen, Germany. All Rights Reserved</dc:rights>
</metadata>
```

The container <common> contains information that can be used by different data aspects/ elements. railML uses codes to represent infrastrucutre managers. Obing is being managed by Chiemgauer Lokalbahn GmbH & Co KG (CLBG) which is supervised by Rhein-Sieg-Eisenbahn GmbH in Bonn [67]. The positioning systems can be used by both the infrastructure and rolling stock and therefore need to be defined in the <common> element. The geometric positioning system follows the EPSG:4326 CRS definition while the relative positioning is used as the linear referencing method. As seen from Figure 40, the station has the name “Obing” and stretches from metre 0.0 to metre 306.6. The code snippet will look as follows:

```
<common id="co_01">
  <organizationalUnits>
    <infrastructureManager id="im_01" code="RSE" />
  </organizationalUnits>
  <positioning>
    <geometricPositioningSystems>
      <geometricPositioningSystem id="gps01" crsDefinition="epsg:4326">
        <name name="WGS84" language="en" />
        <isValid from="2021-01-01" to="2021-08-18" />
      </geometricPositioningSystem>
    </geometricPositioningSystems>
    <linearPositioningSystems>
      <linearPositioningSystem id="lps01" units="m" startMeasure="0.0" endMeasure="306.6"
linearReferencingMethod="relative">
        <name name="railway obing station" language="en" />
        <isValid from="2021-01-01" to="2021-12-31" />
      </linearPositioningSystem>
    </linearPositioningSystems>
  </positioning>
</common>
```

The next step is to model the infrastructure root element which contains all the information regarding the infrastructure.

5.2 Topology:

There are twelve network elements which form **<netElement>** Objects. The elements of the topology in RTM are only logical and can represent anything (e.g. a track on microscopic level, station on mesoscopic level etc). However, knowing that these objects represent a track (linear element), the associated relationships and the length of each <netElement> Object is added as its attribute in railML. The intrinsic, linear and geometric coordinates are then subseqently defined.

The network elements are connected by seventeen possible relations which form the **<netRelation>** Objects. In RTM, these relations are positioned using intrinsic coordinates and occur at the extremities of each <netElement> object. Therefore, the position is forced to have a value of either 0 or 1, meaning that the <netRelation> occurs at the beginning or end respectively of the <netElement> object [23]. Also, RTM uses the attribute @navigability to define which direction(s) movement is posible. AB means a train can move from element A to B but not vice versa, BA means a train can move from element B to A but not vice versa, Both means movement is possible from both A to B and B to A, while None means movement is not possible at all from element A to B. Note that a <netRelation> instance must show the relationship between only two <netElement> objetcs, one is named “element A” and the other “element B”.

The topology is described in only the microscopic level and all objects in the <netElement> and <netRelation> make up the **<networkResource>** objects for this level of description. Therefore the code snippet will look as follows:

```
<topology>
 <netElements>
  <netElement id="ne_a001" length="79">
   <relation ref="nr_a001a006" />
   <relation ref="nr_a001a003" />
   <associatedPositioningSystem id="ne_a001_aps01">
    <intrinsicCoordinate id="ne_a001_aps01_ic01" intrinsicCoord="0">
     <linearCoordinate positioningSystemRef="lps01" measure="0.0" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4030826" y="47.9990330" />
    </intrinsicCoordinate>
    <intrinsicCoordinate id="ne_a001_aps01_ic02" intrinsicCoord="1">
     <linearCoordinate positioningSystemRef="lps01" measure="79" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4033707" y="47.9983509" />
    </intrinsicCoordinate>
   </associatedPositioningSystem>
  </netElement>
  <netElement id="ne_a002" length="110.4">
   <relation ref="nr_a002a004" />
   <relation ref="nr_a002a003" />
   <associatedPositioningSystem id="ne_a002_aps01">
    <intrinsicCoordinate id="ne_a002_aps01_ic01" intrinsicCoord="0">
     <linearCoordinate positioningSystemRef="lps01" measure="10" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4030543" y="47.9989226" />
    </intrinsicCoordinate>
    <intrinsicCoordinate id="ne_a002_aps01_ic02" intrinsicCoord="1">
     <linearCoordinate positioningSystemRef="lps01" measure="120.4" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4034647" y="47.9979690" />
    </intrinsicCoordinate>
   </associatedPositioningSystem>
  </netElement>
```

```xml
  <netElement id="ne_a003" length="41.4">
   <relation ref="nr_a001a003" />
   <relation ref="nr_a002a003" />
   <relation ref="nr_a003a004" />
   <relation ref="nr_a003a006" />
   <associatedPositioningSystem id="ne_a003_aps01">
    <intrinsicCoordinate id="ne_a003_aps01_ic01" intrinsicCoord="0">
     <linearCoordinate positioningSystemRef="lps01" measure="79" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4033707"
y="47.9983509"></geometricCoordinate>
    </intrinsicCoordinate>
    <intrinsicCoordinate id="ne_a003_aps01_ic02" intrinsicCoord="1">
     <linearCoordinate positioningSystemRef="lps01" measure="120.4" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4034647" y="47.9979690" />
    </intrinsicCoordinate>
   </associatedPositioningSystem>
  </netElement>
  <netElement id="ne_a004" length="55.3">
   <relation ref="nr_a002a004" />
   <relation ref="nr_a003a004" />
   <relation ref="nr_a004a009" />
   <associatedPositioningSystem id="ne_a004_aps01">
    <intrinsicCoordinate id="ne_a004_aps01_ic01" intrinsicCoord="0">
     <linearCoordinate positioningSystemRef="lps01" measure="120.4" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4034647" y="47.9979690" />
    </intrinsicCoordinate>
    <intrinsicCoordinate id="ne_a004_aps01_ic02" intrinsicCoord="1">
     <linearCoordinate positioningSystemRef="lps01" measure="175.7" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4036691" y="47.9974938" />
    </intrinsicCoordinate>
   </associatedPositioningSystem>
  </netElement>
  <netElement id="ne_a005" length="89.2">
   <relation ref="nr_a005a007" />
   <relation ref="nr_a005a008" />
   <associatedPositioningSystem id="ne_a005_aps01">
    <intrinsicCoordinate id="ne_a005_aps01_ic01" intrinsicCoord="0">
     <linearCoordinate positioningSystemRef="lps01" measure="56.4" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4033713" y="47.9985926" />
    </intrinsicCoordinate>
    <intrinsicCoordinate id="ne_a005_aps01_ic02" intrinsicCoord="1">
     <linearCoordinate positioningSystemRef="lps01" measure="145.6" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4036889" y="47.9978075" />
    </intrinsicCoordinate>
   </associatedPositioningSystem>
  </netElement>
  <netElement id="ne_a006" length="101.8">
   <relation ref="nr_a001a006" />
   <relation ref="nr_a006a013" />
   <relation ref="nr_a006a007" />
   <associatedPositioningSystem id="ne_a006_aps01">
    <intrinsicCoordinate id="ne_a006_aps01_ic01" intrinsicCoord="0">
     <linearCoordinate positioningSystemRef="lps01" measure="79" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4033707" y="47.9983509" />
    </intrinsicCoordinate>
    <intrinsicCoordinate id="ne_a006_aps01_ic02" intrinsicCoord="1">
     <linearCoordinate positioningSystemRef="lps01" measure="180.8" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4037419" y="47.9974720" />
```

```xml
    </intrinsicCoordinate>
   </associatedPositioningSystem>
  </netElement>
  <netElement id="ne_a007" length="35.2">
   <relation ref="nr_a005a007" />
   <relation ref="nr_a006a007" />
   <relation ref="nr_a007a008" />
   <relation ref="nr_a007a013" />
   <associatedPositioningSystem id="ne_a007_aps01">
    <intrinsicCoordinate id="ne_a007_aps01_ic01" intrinsicCoord="0">
     <linearCoordinate positioningSystemRef="lps01" measure="145.6" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4036889" y="47.9978075" />
    </intrinsicCoordinate>
    <intrinsicCoordinate id="ne_a007_aps01_ic02" intrinsicCoord="1">
     <linearCoordinate positioningSystemRef="lps01" measure="180.8" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4037419" y="47.9974720" />
    </intrinsicCoordinate>
   </associatedPositioningSystem>
  </netElement>
  <netElement id="ne_a008" length="38.7">
   <relation ref="nr_a005a008" />
   <relation ref="nr_a007a008" />
   <associatedPositioningSystem id="ne_a008_aps01">
    <intrinsicCoordinate id="ne_a008_aps01_ic01" intrinsicCoord="0">
     <linearCoordinate positioningSystemRef="lps01" measure="145.6" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4036889" y="47.9978075" />
    </intrinsicCoordinate>
    <intrinsicCoordinate id="ne_a008_aps01_ic02" intrinsicCoord="1">
     <linearCoordinate positioningSystemRef="lps01" measure="184.3" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4038244" y="47.9974724" />
    </intrinsicCoordinate>
   </associatedPositioningSystem>
  </netElement>
  <netElement id="ne_a009" length="6.4">
   <relation ref="nr_a004a009" />
   <relation ref="nr_a009a011" />
   <associatedPositioningSystem id="ne_a009_aps01">
    <intrinsicCoordinate id="ne_a009_aps01_ic01" intrinsicCoord="0">
     <linearCoordinate positioningSystemRef="lps01" measure="175.7" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4036691" y="47.9974938" />
    </intrinsicCoordinate>
    <intrinsicCoordinate id="ne_a009_aps01_ic02" intrinsicCoord="1">
     <linearCoordinate positioningSystemRef="lps01" measure="182.1" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4036954" y="47.9974409" />
    </intrinsicCoordinate>
   </associatedPositioningSystem>
  </netElement>
  <netElement id="ne_a011" length="30.6">
   <relation ref="nr_a009a011" />
   <relation ref="nr_a011a012" />
   <relation ref="nr_a011a013" />
   <associatedPositioningSystem id="ne_a011_aps01">
    <intrinsicCoordinate id="ne_a011_aps01_ic01" intrinsicCoord="0">
     <linearCoordinate positioningSystemRef="lps01" measure="182.1" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4036954" y="47.9974409" />
    </intrinsicCoordinate>
    <intrinsicCoordinate id="ne_a011_aps01_ic02" intrinsicCoord="1">
     <linearCoordinate positioningSystemRef="lps01" measure="212.7" />
```

```xml
      <geometricCoordinate positioningSystemRef="gps01" x="12.4038586" y="47.9971955" />
     </intrinsicCoordinate>
    </associatedPositioningSystem>
   </netElement>
   <netElement id="ne_a012" length="93.9">
    <relation ref="nr_a011a012" />
    <relation ref="nr_a012a013" />
    <associatedPositioningSystem id="ne_a012_aps01">
     <intrinsicCoordinate id="ne_a012_aps01_ic01" intrinsicCoord="0">
      <linearCoordinate positioningSystemRef="lps01" measure="212.7" />
      <geometricCoordinate positioningSystemRef="gps01" x="12.4038586" y="47.9971955" />
     </intrinsicCoordinate>
     <intrinsicCoordinate id="ne_a012_aps01_ic02" intrinsicCoord="1">
      <linearCoordinate positioningSystemRef="lps01" measure="306.6" />
      <geometricCoordinate positioningSystemRef="gps01" x="12.4040525" y="47.9963222" />
     </intrinsicCoordinate>
    </associatedPositioningSystem>
   </netElement>
   <netElement id="ne_a013" length="31.9">
    <relation ref="nr_a006a013" />
    <relation ref="nr_a007a013" />
    <relation ref="nr_a011a013" />
    <relation ref="nr_a012a013" />
    <associatedPositioningSystem id="ne_a013_aps01">
     <intrinsicCoordinate id="ne_a013_aps01_ic01" intrinsicCoord="0">
      <linearCoordinate positioningSystemRef="lps01" measure="180.8" />
      <geometricCoordinate positioningSystemRef="gps01" x="12.4037419" y="47.9974720" />
     </intrinsicCoordinate>
     <intrinsicCoordinate id="ne_a013_aps01_ic02" intrinsicCoord="1">
      <linearCoordinate positioningSystemRef="lps01" measure="212.7" />
      <geometricCoordinate positioningSystemRef="gps01" x="12.4038586" y="47.9971955" />
     </intrinsicCoordinate>
    </associatedPositioningSystem>
   </netElement>
  </netElements>
  <netRelations>
   <netRelation id="nr_a001a006" positionOnA="1" positionOnB="0" navigability="Both">
    <elementA ref="ne_a001" />
    <elementB ref="ne_a006" />
   </netRelation>
   <netRelation id="nr_a001a003" positionOnA="1" positionOnB="0" navigability="Both">
    <elementA ref="ne_a001" />
    <elementB ref="ne_a003" />
   </netRelation>
   <netRelation id="nr_a002a004" positionOnA="1" positionOnB="0" navigability="Both">
    <elementA ref="ne_a002" />
    <elementB ref="ne_a004" />
   </netRelation>
   <netRelation id="nr_a002a003" positionOnA="1" positionOnB="1" navigability="None">
    <elementA ref="ne_a002" />
    <elementB ref="ne_a003" />
   </netRelation>
   <netRelation id="nr_a003a004" positionOnA="1" positionOnB="0" navigability="Both">
    <elementA ref="ne_a003" />
    <elementB ref="ne_a004" />
   </netRelation>
   <netRelation id="nr_a003a006" positionOnA="0" positionOnB="0" navigability="None">
    <elementA ref="ne_a003" />
```

```xml
    <elementB ref="ne_a006" />
  </netRelation>
  <netRelation id="nr_a004a009" positionOnA="1" positionOnB="0" navigability="Both">
    <elementA ref="ne_a004" />
    <elementB ref="ne_a009" />
  </netRelation>
  <netRelation id="nr_a005a007" positionOnA="1" positionOnB="0" navigability="Both">
    <elementA ref="ne_a005" />
    <elementB ref="ne_a007" />
  </netRelation>
  <netRelation id="nr_a005a008" positionOnA="1" positionOnB="0" navigability="Both">
    <elementA ref="ne_a005" />
    <elementB ref="ne_a008" />
  </netRelation>
  <netRelation id="nr_a006a013" positionOnA="1" positionOnB="0" navigability="Both">
    <elementA ref="ne_a006" />
    <elementB ref="ne_a013" />
  </netRelation>
  <netRelation id="nr_a006a007" positionOnA="1" positionOnB="1" navigability="None">
    <elementA ref="ne_a006" />
    <elementB ref="ne_a007" />
  </netRelation>
  <netRelation id="nr_a007a013" positionOnA="1" positionOnB="0" navigability="Both">
    <elementA ref="ne_a007" />
    <elementB ref="ne_a013" />
  </netRelation>
  <netRelation id="nr_a007a008" positionOnA="0" positionOnB="0" navigability="None">
    <elementA ref="ne_a007" />
    <elementB ref="ne_a008" />
  </netRelation>
  <netRelation id="nr_a009a011" positionOnA="1" positionOnB="0" navigability="Both">
    <elementA ref="ne_a009" />
    <elementB ref="ne_a011" />
  </netRelation>
  <netRelation id="nr_a011a012" positionOnA="1" positionOnB="0" navigability="Both">
    <elementA ref="ne_a011" />
    <elementB ref="ne_a012" />
  </netRelation>
  <netRelation id="nr_a011a013" positionOnA="1" positionOnB="1" navigability="None">
    <elementA ref="ne_a011" />
    <elementB ref="ne_a013" />
  </netRelation>
  <netRelation id="nr_a012a013" positionOnA="0" positionOnB="1" navigability="Both">
    <elementA ref="ne_a012" />
    <elementB ref="ne_a013" />
  </netRelation>
</netRelations>
<networks>
  <network id="nw01">
    <level id="Lv0" descriptionLevel="Micro">
      <networkResource ref="ne_a001" />
      <networkResource ref="ne_a002" />
      <networkResource ref="ne_a003" />
      <networkResource ref="ne_a004" />
      <networkResource ref="ne_a005" />
      <networkResource ref="ne_a006" />
      <networkResource ref="ne_a007" />
      <networkResource ref="ne_a008" />
```

```xml
    <networkResource ref="ne_a009" />
    <networkResource ref="ne_a011" />
    <networkResource ref="ne_a012" />
    <networkResource ref="ne_a013" />
    <networkResource ref="nr_a001a006" />
    <networkResource ref="nr_a001a003" />
    <networkResource ref="nr_a002a004" />
    <networkResource ref="nr_a002a003" />
    <networkResource ref="nr_a003a004" />
    <networkResource ref="nr_a003a006" />
    <networkResource ref="nr_a004a009" />
    <networkResource ref="nr_a005a007" />
    <networkResource ref="nr_a005a008" />
    <networkResource ref="nr_a006a013" />
    <networkResource ref="nr_a006a007" />
    <networkResource ref="nr_a007a008" />
    <networkResource ref="nr_a007a013" />
    <networkResource ref="nr_a009a011" />
    <networkResource ref="nr_a011a012" />
    <networkResource ref="nr_a011a013" />
    <networkResource ref="nr_a012a013" />
   </level>
  </network>
 </networks>
</topology>
```

5.3 Functional Infrastrucure:

The obtained data for Obing station contains three buffer stops, five switches and five tracks which are modelled under <functionalinfrastructure> in railML. The buffer stops and switches are of the type “node” in overpass-turbo, while tracks are of the type “way”.

In modelling the switch, information on the name of the switch, type and branch (branching direction and continue course direction) are given. The left and right branch of the switch references a <netRelation> object, thereby building the connection between the switch model and the topology. The switch’s continue course describes the mainly used branch. The default course describes the standby position of the switch which can sometimes be different from the continue course [65]. Information on the default course is omitted in the railML model for Obing since it is unknown from GeoJSON data, and cannot easily be inferred. All switches in Obing are assumed to be ordinary switches and of the same radius. The values for the radius of branch course is based on assumption since this information is not given but an important part of a switch model. It is assumed as 100m considering that the maximum speed possible in the station is given as 40 meters per second. Likewise, the branching and joining speed are all based on assumptions using the maximum allowed speed of the track section closest to it. Care must be taken not to define the permissible speed at switches as extra line or track speed changes when provided [65]. The buffer stops are also assumed to all be of the type “fixed buffer stop”.
Finally, the tracks are the last set of functional elements modelled for Obing station. Each track references at least one <netElement> object (but can reference more than one <netElement> objects as seen in the main track “badEndorf-Obing”) on the microscopic level, thereby building the connection between the track and the topology. Obing station has only one “mainTrack”.

The two tracks with service “crossover” in OSM are designated the type “connectingTrack” in railML, while the other two tracks with service “yard” in OSM are designated the type “sidingTrack”. The elements <trackBegin> and <trackEnd> tells the orientation of the track. Each instance of the child element <associatedNetElement> references a <netElement> object where the attribute @posBegin and @posEnd tells which part of the referenced <netElement> object is part of the track, and @keepsOrientation tells that the track has the same orientation as the referenced <netElement> [23]. In addition to @type, other optional attributes that could be added include the radius, main direction, gradient etc. These lead to the code snippet:

```
<functionalInfrastructure>
 <bufferStops>
  <bufferStop id="bus01" type="fixedBufferStop">
   <spotLocation id="bus01_sloc01" netElementRef="ne_a001" applicationDirection="reverse"
pos="0.0">
    <linearCoordinate positioningSystemRef="lps01" measure="0.0" />
    <geometricCoordinate positioningSystemRef="gps01" x="12.4030826" y="47.9990330" />
   </spotLocation>
  </bufferStop>
  <bufferStop id="bus02" type="fixedBufferStop">
   <spotLocation id="bus02_sloc01" netElementRef="ne_a005" applicationDirection="reverse"
pos="0.0">
    <linearCoordinate positioningSystemRef="lps01" measure="56.4" />
    <geometricCoordinate positioningSystemRef="gps01" x="12.4033713" y="47.9985926" />
   </spotLocation>
  </bufferStop>
  <bufferStop id="bus03" type="fixedBufferStop">
   <spotLocation id="bus03_sloc01" netElementRef="ne_a002" applicationDirection="reverse"
pos="0.0">
    <linearCoordinate positioningSystemRef="lps01" measure="10" />
    <geometricCoordinate positioningSystemRef="gps01" x="12.4030543" y="47.9989226" />
   </spotLocation>
  </bufferStop>
 </bufferStops>
 <switchesIS>
  <switchIS id="swi01" continueCourse="left" branchCourse="right" type="ordinarySwitch">
   <spotLocation id="swi01_sloc01" netElementRef="ne_a001" applicationDirection="normal" pos="79">
    <linearCoordinate positioningSystemRef="lps01" measure="79" />
    <geometricCoordinate positioningSystemRef="gps01" x="12.4033707" y="47.9983509" />
   </spotLocation>
   <leftBranch netRelationRef="nr_a001a006" branchingSpeed="25" joiningSpeed="40" radius="0" />
   <rightBranch netRelationRef="nr_a001a003" branchingSpeed="5" joiningSpeed="40" radius="100" />
  </switchIS>
  <switchIS id="swi02" continueCourse="left" branchCourse="right" type="ordinarySwitch">
   <spotLocation id="swi02_sloc01" netElementRef="ne_a004" applicationDirection="reverse" pos="0">
    <linearCoordinate positioningSystemRef="lps01" measure="120.4" />
    <geometricCoordinate positioningSystemRef="gps01" x="12.4034647" y="47.9979690" />
   </spotLocation>
   <leftBranch netRelationRef="nr_a002a004" branchingSpeed="5" joiningSpeed="5" radius="0" />
   <rightBranch netRelationRef="nr_a003a004" branchingSpeed="5" joiningSpeed="5" radius="100" />
  </switchIS>
  <switchIS id="swi03" continueCourse="left" branchCourse="right" type="ordinarySwitch">
   <spotLocation id="swi03_sloc01" netElementRef="ne_a005" applicationDirection="normal"
pos="89.2">
    <linearCoordinate positioningSystemRef="lps01" measure="145.6" />
    <geometricCoordinate positioningSystemRef="gps01" x="12.4036889" y="47.9978075" />
```

```xml
    </spotLocation>
    <leftBranch netRelationRef="nr_a005a008" branchingSpeed="5" joiningSpeed="5" radius="0" />
    <rightBranch netRelationRef="nr_a005a007" branchingSpeed="5" joiningSpeed="5" radius="100" />
   </switchIS>
   <switchIS id="swi04" continueCourse="left" branchCourse="right" type="ordinarySwitch">
    <spotLocation id="swi04_sloc01" netElementRef="ne_a013" applicationDirection="reverse"
pos="0.0">
     <linearCoordinate positioningSystemRef="lps01" measure="180.8" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4037419" y="47.9974720" />
    </spotLocation>
    <leftBranch netRelationRef="nr_a006a013" branchingSpeed="25" joiningSpeed="40" radius="0" />
    <rightBranch netRelationRef="nr_a007a013" branchingSpeed="5" joiningSpeed="25" radius="100" />
   </switchIS>
   <switchIS id="swi05" continueCourse="right" branchCourse="left" type="ordinarySwitch">
    <spotLocation id="swi05_sloc01" netElementRef="ne_a012" applicationDirection="reverse"
pos="0.0">
     <linearCoordinate positioningSystemRef="lps01" measure="212.7" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4038586" y="47.9971955" />
    </spotLocation>
    <leftBranch netRelationRef="nr_a011a012" branchingSpeed="5" joiningSpeed="25" radius="100" />
    <rightBranch netRelationRef="nr_a012a013" branchingSpeed="25" joiningSpeed="25" radius="0" />
   </switchIS>
  </switchesIS>
  <tracks>
   <track id="badEndorf-Obing" type="mainTrack">
    <linearLocation id="badEndorf-Obing_lloc01" applicationDirection="both">
     <associatedNetElement netElementRef="ne_a001" keepsOrientation="true" posBegin="0.0"
posEnd="79"></associatedNetElement>
     <associatedNetElement netElementRef="ne_a006" keepsOrientation="true" posBegin="0.0"
posEnd="101.8"></associatedNetElement>
     <associatedNetElement netElementRef="ne_a013" keepsOrientation="true" posBegin="0.0"
posEnd="31.9"></associatedNetElement>
    </linearLocation>
    <trackBegin ref="bus01" />
    <trackEnd ref="swi05" />
    <length value="212.7" type="physical" />
   </track>
   <track id="service_yard1-Obing" type="sidingTrack">
    <linearLocation id="service_yard1-Obing_lloc01" applicationDirection="both">
     <associatedNetElement netElementRef="ne_a005" keepsOrientation="true" posBegin="0.0"
posEnd="89.2"></associatedNetElement>
    </linearLocation>
    <trackBegin ref="bus02" />
    <trackEnd ref="swi03" />
    <length value="89.2" type="physical" />
   </track>
   <track id="service_yard2-Obing" type="sidingTrack">
    <linearLocation id="service_yard2-Obing_lloc01" applicationDirection="both">
     <associatedNetElement netElementRef="ne_a002" keepsOrientation="true" posBegin="0.0"
posEnd="120.4"></associatedNetElement>
     <associatedNetElement netElementRef="ne_a004" keepsOrientation="true" posBegin="0.0"
posEnd="92.3"></associatedNetElement>
    </linearLocation>
    <trackBegin ref="bus03" />
    <trackEnd ref="swi05" />
    <length value="202.7" type="physical" />
   </track>
   <track id="connection1-Obing" type="connectingTrack">
```

```
    <linearLocation id="connection1-Obing_lloc01" applicationDirection="both">
     <associatedNetElement netElementRef="ne_a007" keepsOrientation="true" posBegin="0.0"
posEnd="35.2"></associatedNetElement>
    </linearLocation>
    <trackBegin ref="swi03" />
    <trackEnd ref="swi04" />
    <length value="35.2" type="physical" />
   </track>
   <track id="connection2-Obing" type="connectingTrack">
    <linearLocation id="connection2-Obing_lloc01" applicationDirection="both">
     <associatedNetElement netElementRef="ne_a003" keepsOrientation="true" posBegin="0.0"
posEnd="41.4"></associatedNetElement>
    </linearLocation>
    <trackBegin ref="swi01" />
    <trackEnd ref="swi02" />
    <length value="41.4" type="physical" />
   </track>
  </tracks>
</functionalInfrastructure>
```

5.4 Validation of Obing in railVIVID

The complete railML 3.1 code cannot be visualised in railVIVID as stated in 2.8, however, the validation returns zero error with the message “The validation finished without messages. The document is syntactically correct” as shown in Figure 42. To validate a railML file in railVIVID, the “open file” is left-clicked and then the file (which can be saved with the .xml or .railml extension) is selected. Once successfully loaded, a summary of the file content is displayed separately. If this is “ok” as the intended file, the next step is to left click on “validator” and then “validate”. The result of the validation is also displayed in a separate window. If there are (syntactic) errors, this will be displayed with a description and reference to where the error(s) occurred.

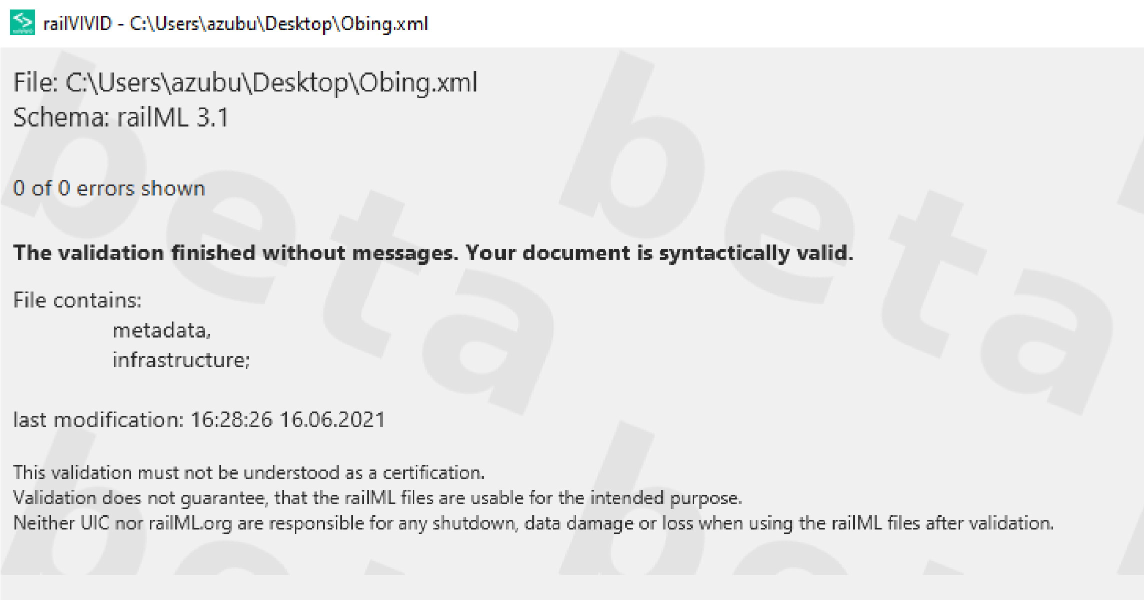


Figure 42: Validation of Obing in railVIVID

5.5 SWOT Analysis of converting and modelling different railway data sources in railML

The content of the analysis in Table 8 and Table 9 reflects the perspective of the author when trying to use the content of the data from the infrastrucuture data sources described in chapter 2.4 to model the railway infrastrucutre of Obing station in railML 3.1, which implements the RTM. It can be seen that OSM represents the best source for obtaining railway data for conversion and modelling in RTM. DB Netz AG is the IM for most of the railway lines in Germany and therefore represents the most reliable source of railway data in Germany, however its open data portal does not provide data in such a way as OSM, where all the relevant (railway) data (such as the tracks, functional infrastructure elements associated to the tracks, location etc) regarding a particular area or route can be obtained in a single file. Rather, several files have to be pieced together which can be highly time consuming.

5.5.1 SWOT Analysis of converting and modelling railway data from OSM to railML 3.1

Table 8: SWOT Analysis of converting OSM data to railML 3.1

STRENGTHS (+)	WEAKNESSES (-)
+ No license fee required. + Possible to download data in different text and image formats. This allows for the flexibility of interpreting the data from any preferred format. + Topology aggregation is possible since information on the most basic level of description is provided. + Possible to model functional infrastructure elements. + The (railway) data related to an area is well arranged and structured.	– Sometimes, some functional elements and attributes of the functional elements are missing or ambiguous and have to be assumed. – Coordinate reference is available in the WGS84. No linear reference method available and therefore has to be done manually
OPPORTUNITIES (+)	**THREATS (-)**
+ Further mapping could improve the quantity/content of the data, making it more extensive. + Can easily be updated (by individuals) to reflect up to data without waiting for an official update from the IM which often takes time.	– OSM data base is managed by a private foundation which cannot be held accountable for any false data. – Since private individuals can add data and no IM manages any data in OSM, the accuracy of the data cannot be assured

5.5.2 SWOT Analysis of converting and modelling railway data from Microsoft Bing and Google maps to railML 3.1

Table 9: SWOT Analysis of converting Bing/Google maps to railML 3.1

STRENGTHS (+)	WEAKNESSES (-)
+ No license fee required. + Possible to download data in image formats. + Using the image data to model the topology is possible.	– Not possible to download data in text format. – Basic level of description of the topology is sometimes missing. – Modelling infrastructure elements is difficult (if possible) due to lack of sufficient information regarding its attributes.
OPPORTUNITIES (+)	**THREATS (-)**
+ Can easily be updated	– Since both sources are also not owned by any IM, the accuracy of the data cannot be assured.

6 Importing Track Infrastrucutre data of Obing to Gazebo

The railway track of the station Obing can be imported as a custom shape in gazebo, using the model editor menu. While all known shapes in gazebo should be in 3D, GIS file formats are usually in 2D. Therefore, an interface is needed to convert the exported data. AutoCAD Map 3D is used for this conversion because it is suited to read 2D maps saved in different formats including shape and text formats. This is advantageous because, data from the overpass-turbo can only be exported using one of the following GIS text formats: GML, KML, GeoJSON and OSM.

The 2D data can be downloaded through the overpass-turbo API using the code in **Appendix 2**, this time in .kml format which is readable by AutoCAD Map 3D as seen in Figure 43. To open this file, one has to left-click on the “insert” menu at the top left, and then click on “Map import” from where the desired .kml file (or any other recognisable format) can be selected from the stored location.

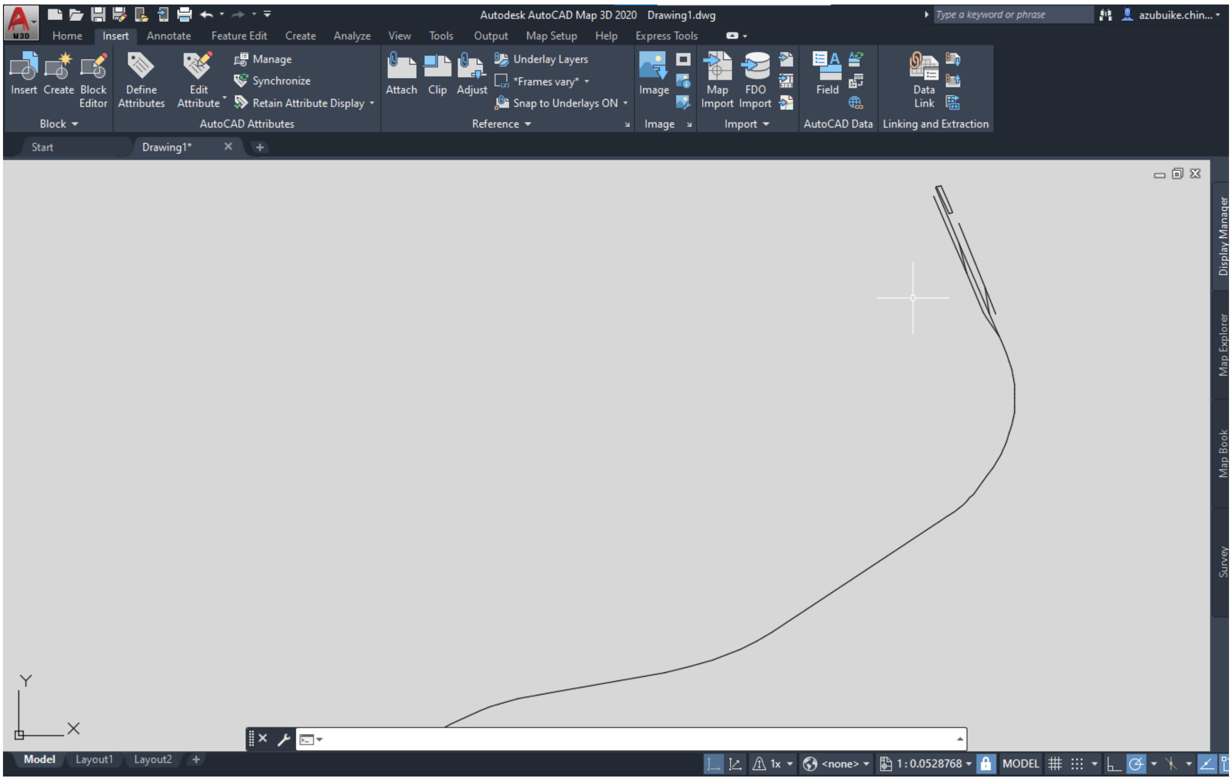


Figure 43: Visualization of Obing in AutoCAD Map 3D

The data can then be converted to several other formats that are readable by CAD programs such as SolidWorks and AutoCAD. Different operations are performed on the 2D visualisation to convert it into a 3D shape (Figure 44 and Figure 45) which has to be saved in .stl. SolidWorks is capable of saving a 3D model in .stl format.

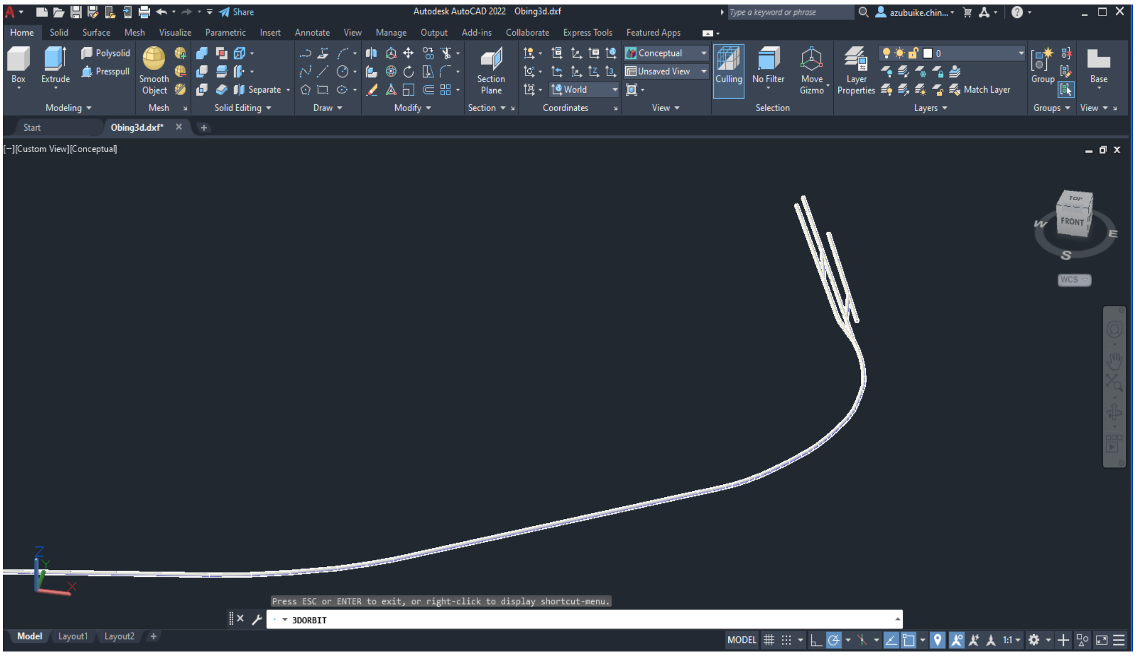


Figure 44: 3D Model visualisation in AutoCAD

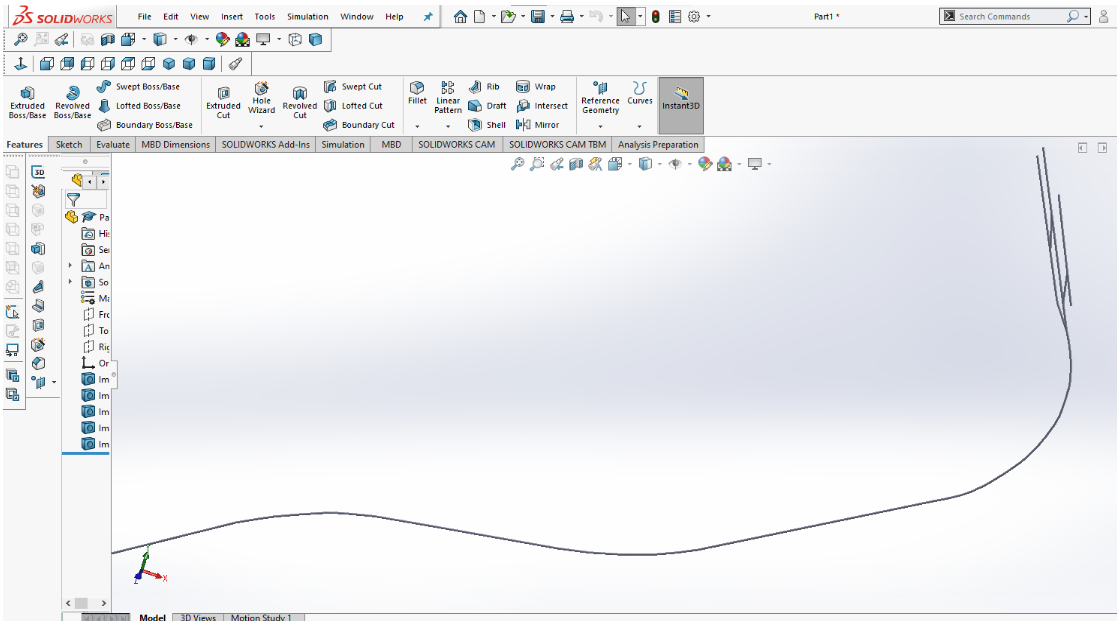


Figure 45: 3D Model visualisation in SolidwWorks

The .stl file can then be opened in gazebo by inserting it as a custom model, in the model editor tab (Figure 46). Sometimes, the file might not be visible in gazebo when imported because it's dimensions are too small. Increasing the scale of the model in AutoCAD before saving in .stl can solve this problem.

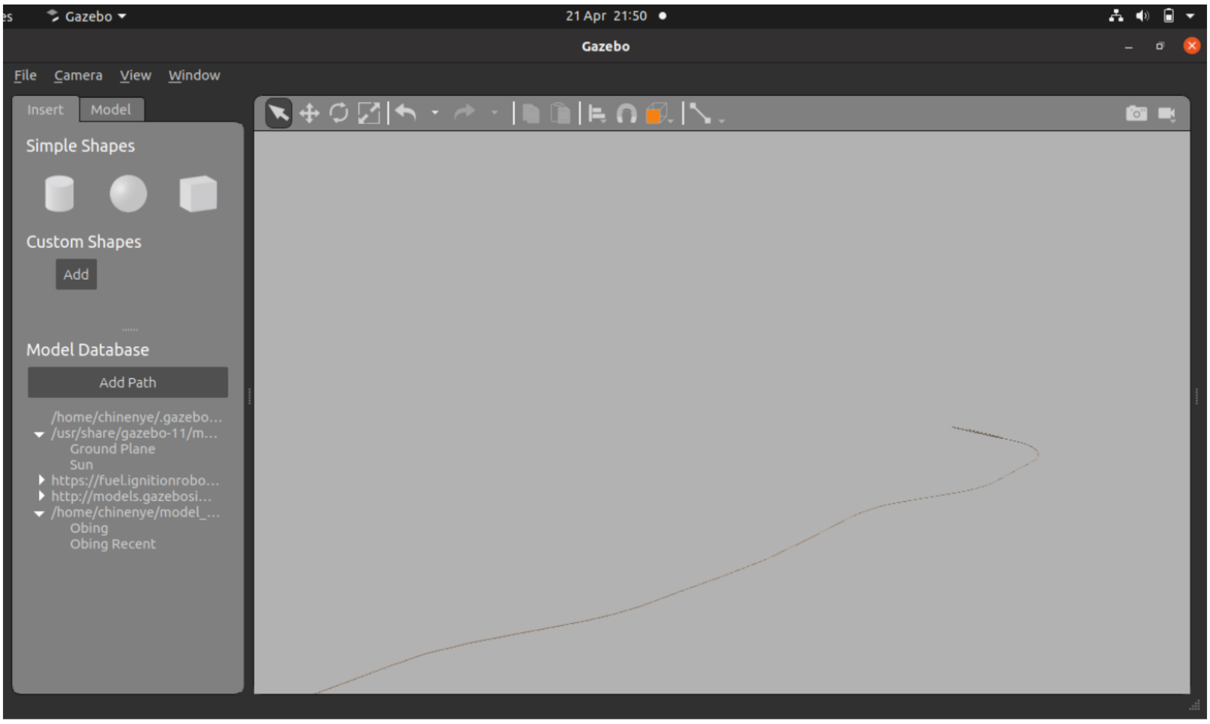


Figure 46: Visualization in gazebo

This simply shows that there is a possibility to import railway data from available map sources in gazebo without having to build from the scratch. However, this does not examine the accuracy of the converted data since no dimensions where examined all through the conversion process.

7 Conclusion and Outlook

This research work shows many available infrastructure data sources which are free of restrictive licenses from which railway track data for a small station – Obing, can be sourced. While many sources are deficient of some important aspects of the railway station, ORM provides enough information for modelling in RTM using the railML 3.1 format. Nevertheless, some minute information for a more accurate modelling were missing. Such missing information includes the radius of switches used, explicit information on the allowed speed at the switch and the default direction of the switch. In addition, a track can sometimes be curved, but the information on the radius of a track is not provided. This limitation is inheritent on the structure of most GIS vector data formats which can be exported from ORM using the overpass-turbo. To examine the results of this approach, a visualisation on railVIVID is needed, which is presently not possible for railML 3.1 format.

For simulation of railway movements, it has also been shown that it is unecessary to build the station from the scratch since it is possible to convert the railway data from available 2D formats to .stl (3D) format which can be read by gazebo.

It is recommended as next steps to examine the possibility of writing a script which can do some or all of the conversion processes. Also, owing to the fact that webot can directly utilise OSM map data, it is recommended to examine if there is an interface in this software that can be used to filter for only the railway data which can then be used for simulations. This solution might imply that webot could also be more advantageous in automated railway simulations when compared to gazebo, in terms of importing accurate railway track data. Operations in AutoCAD or SolidWorks to obtain a more realistic 3D representation of the physical view of the track from the available 2D view can be investigated, even though intersections (nodes) might pose a challenge.

Finally, a more futuristic approach which represents the best case scenario would be for existing and future railway map providers to make the railML 3.1 format as an available export option. This would also require that the IMs and RUs make their railway data more open, so that more (accurate) details can be included in the public sources.

Bibliography

[1] EU BIM Task Group, “Handbook for the introduction of Building Information Modelling by the European Public Sector,” 2017.

[2] Siemens Mobility Global Website, Siemens Mobility, 28 03 2019. [Online]. Available: https://www.mobility.siemens.com/global/en/portfolio/rail/stories/how-digitalization-is-revolutionizing-rail-traffic.html. [Accessed 07 October 2020].

[3] Jana Pieriegud, "Digital Transformation of Railways,", Siemens Sp. z o. o., Poland, 2018.

[4] Roland Berger Strategy Consultants and Bundesverband der Deutschen Industry e.V., "The Digital Transformation of Industry," Berlin, 2015.

[5] Union Internationale des Transports Publics (UITP), "World Report On Metro Automation," 2018.

[6] F. Esser and C. Schindler, "Assisted, Automated and Autonomous Driving ("Tripple A") for Railway Traffic," RAILCON '16, 2016.

[7] L. Fenzi, "Rise of the machines: Rio Tinto breaks new Ground with AutoHaul," 2019.

[8] D. Burroughs, "DB and Siemens to develop fully automated S-Bahn line in Hamburg," 2018.

[9] Community of European Railway and Infrastructure Companies (CER), "30 by 2030 – how rail freight achieves its goals," Brussels, 2020.

[10] DB Systel GmbH, *The future of shunting,* Digital Spirit, 2017.

[11] L. Intermodal, "Lineas Intermodal: News," Lineas Intermodal, 15 July 2020. [Online]. Available: https://lineasintermodal.net/en/news/ato. [Accessed 08 October 2020].

[12] Rail Vision, "Streamlining Shunting Yard Operations through Automation and Deep Learning," Rail Vision, 2019.

[13] L. Flamm and B. Scheier, "An integrated evaluation of train control systems on the path to automated train operations," vol. III, no. 6, 2019.

[14] I. A. Hansen, "Automated shunting of rail container wagons in ports and terminal areas," vol. 27, no. 5, 2004.

[15] C. Gralla, "The brain train - autonomous driving on the rails connects the science region," Aachen, 2019.

[16] European Railway Agency (ERA), Big data in railways, European Railway Agency (ERA), 2016.

[17] International Union of railways (UIC), RailTopoModel - A conceptual model to describe a railway network, v1.0 ed., Paris: UIC, 2015.

[18] B. Ning and X. Li, *A comprehensive information system for railway networks,* WIT Press, 2002.

[19] THE EUROPEAN PARLIAMENT AND THE COUNCIL OF THE EUROPEAN UNION, "DIRECTIVE (EU) 2016/797 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 11 May 2016 on the interoperability of the rail system within the European Union," *Official Journal of the European Union,* vol. 138, pp. 44-101, 2016.

[20] railML.org, "The railML subschemas," [Online]. Available: https://www.railml.org/en/user/subschemes.html. [Accessed 06 June 2021].

[21] railML.org, "Infrastructure," [Online]. Available: https://www.railml.org/en/user/subschemes/infrastructure.html. [Accessed 25 June 2021].

[22] railML 3 wiki, "railML version 3 main page," railML.org, [Online]. Available: https://wiki3.railml.org. [Accessed 23 June 2021].

[23] International Union of Railways (UIC), RailTopoModel - Railway infrastructure topological model, vol. v1.0, Paris: International Union of Railways, 2016.

[24] R. N. D. Kes, Generic Topology model for Rail infrastructure - A foundation for standardized handling of rail infrastructure data, ProRail, 2015.

[25] International Union of Railways (UIC)/ trafIT Solutions GmbH, "Feasibility study - UIC RailTopoModel and data exchange format," International Union of Railways, Zurich/Paris, 2013.

[26] S. Wunsch, M. Lehnert, J. Krimmling and J. Easton, "Data formats, models and concepts for railway operations - selected results of work package 3.4 in the EU project Capacity4Rail," Railway Traffic | Capacity, Dresden, 2016.

[27] G. Caimi, L. Kroon and C. Liebchen, "Models for Railway Timetable Optimization: Applicability and Applications in Practice," *Journal of Railway Transport Planning and Management,* vol. 6, no. 4, pp. 285 - 312, 2017.

[28] DB Vertrieb GmbH, "Fahrplan API: Documentation," 25 February 2016. [Online]. Available: https://data.deutschebahn.com/dataset/api-fahrplan.html. [Accessed 25 June 2021].

[29] Rail Delivery Group - National Rail, Timetable Information: Data Feed Interface Specification, 03-00 ed., London: Rail Settlement Plan Limited, 2018.

[30] A. Radtke and D. Hauptmann, "Automated planning of timetables in large railway networks using a microscopic data basis and railway simulation techniques," *Computers in Railways IX,* vol. 74, 2004.

[31] V. Cacchiani, A. Caprara and P. Toth, "Models and Algorithms for the Train Unit Assignment Problem," *International Symposium on Combinatorial Optimization,* vol. 7422, pp. 24-35, 2012.

[32] A. Merel, X. Gandibleux, S. Demassey and R. Lusby, "An improved Upper Bound for the Railway Infrastructure Capacity Problem on the Pierrefitte-Gonesse Junction," *Actes du dixième congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision,* pp. 62-79, 2009.

[33] J. Pachl, Railway Operation and Control, 4th ed., Mountlake Terrace: VTD Rail Publishing, 2018.

[34] DB Netz AG, "PlanPro: Data modelling for planning of interlocking systems - Activities of DB Netz AG," DB Netz AG, Paris, 2013.

[35] Eclipse Foundation, "Eclipse Signalling Engineering Toolbox," Eclipse Foundation, [Online]. Available: https://projects.eclipse.org/proposals/eclipse-signalling-engineering-toolbox. [Accessed 18 September 2020].

[36] D. B. Eisweiler, "EULYNX - European Initiative Linking Interlocking Systems - 23," in *Innotrans*, Berlin, 2014.

[37] H. Butler, M. Daly, A. Doyle, S. Gillies, T. Schaub and S. Hagen, *The GeoJSON Format,* Internet Engineering Task Force (IETF), 2016.

[38] Environmental Systems Research Institute (ESRI), Inc., "ESRI Shapefile Technical Description - An ESRI White Paper," Environmental Systems Research Institute, Inc., California, 1998.

[39] Open Geospatial Consortium (OGC), OGC KML 2.3, vol. 1, D. Burggraf, Ed., Open Geospatial Consortium, 2015.

[40] Open Geospatial Consortium (OGC), "OGC® Approves KML as Open Standard," 14 April 2008. [Online]. Available: https://www.ogc.org/pressroom/pressreleases/857. [Accessed 13 May 2021].

[41] Google Developers, "KML Reference," 19 August 2020. [Online]. Available: https://developers.google.com/kml/documentation/kmlreference. [Accessed 20 May 2021].

[42] OpenstreetMap, "History of OpenStreetMap," OpenStreetMap Wiki, 17 April 2021. [Online]. Available: https://wiki.openstreetmap.org/w/index.php?title=History_of_OpenStreetMap&oldid=2142980. [Accessed 25 June 2021].

[43] OpenStreetMap Wiki contributors, "OpenRailwayMap," OpenStreetMap Wiki, 14 February 2021. [Online]. Available: https://wiki.openstreetmap.org/wiki/OpenRailwayMap. [Accessed 25 June 2021].

[44] M. Binzen and J. Jüngst, Open Data und Open Source bei der Deutschen Bahn - Aktivitäten und Erfahrungen, DB Open Data & Hackathon, 2019.

[45] OpenRailwayMap blog, "OpenRailwayMap - Deutsche Bahn and Open-Data," OpenRailwayMap, 03 November 2015. [Online]. Available: https://blog.openrailwaymap.org/. [Accessed 12 February 2021].

[46] L. G'ely, G. Dessagne, P. Pesneau and F. Vanderbeck, "A multi scalable model based on a connexity graph representation," *WIT Transactions on the Built Environment,* vol. 114, pp. 193-204, 2010.

[47] A. Hlubucek, "RailTopoModel and railML 3 in overall context," *Acta Polytechnica CTU Proceedings,* vol. 11, pp. 16-21, 2017.

[48] railML.org, "railML.org: Download railVIVD," railML.org, [Online]. Available: https://www.railml.org/en/user/railvivid/downloadV12.html. [Accessed 26 June 2021].

[49] H. Choi, C. Crump, C. Duriez, A. Elmquist, G. Hager, D. Han, F. Hearl, J. Hodgins, A. Jain, F. Leve, C. Li, F. Meier, D. Negrut, L. Righetti, A. Rodriguez, J. Tan and J. Trinkle, On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward, vol. 118, N. Simaan, Ed., Proceedings of the National Academy of Sciences of the United States of America, 2021.

[50] Cyberbotics, "Webots: Open Source Robot Simulator," Cyberbotics, [Online]. Available: https://cyberbotics.com/. [Accessed 17 June 2021].

[51] Gazebosim.org, "Gazebo: Robot Simulation made Easy," [Online]. Available: http://gazebosim.org/. [Accessed 18 June 2021].

[52] Coppelia Robotics, "CoppeliaSim," [Online]. Available: https://www.coppeliarobotics.com/. [Accessed 18 June 2021].

[53] STDR Simulator, "S.T.D.R. Simulator," 2014. [Online]. Available: http://stdr-simulator-ros-pkg.github.io/#prettyPhoto. [Accessed 18 June 2021].

[54] ROS Wiki, "ROS.org: stdr_simulator Package Summary," 25 July 2014. [Online]. Available: http://wiki.ros.org/stdr_simulator. [Accessed 18 June 2021].

[55] T. Quaatrani, Introduction to UML 2.0, IBM Corporation, 2005.

[56] X. He, Z. Ma, W. Shao and G. Li, "A metamodel for the notation of graphical modelling languages," in *31st Annual International Computer Software and Applications Conference(COMPSAC 2007)*, Beijing, 2007.

[57] Visual Paradigm, "What is Unified Modelling Language (UML)," Visual Paradigm, [Online]. Available: https://www.visual-paradigm.com/guide/uml-unified-modelling-language/what-is-uml/. [Accessed 05 April 2021].

[58] Object Management Group (OMG), OMG Unified Modelling Language (OMG UML) - Version 2.5, Object Management Group Inc., 2015.

[59] railML Wiki contributors, "Times," railML.org, [Online]. Available: https://wiki2.railml.org/wiki/TT:times. [Accessed 16 June 2021].

[60] D. L. Anderson, "The Shape of the Earth, Heat Flow and Convection," in *Theory of the Earth*, J. Staples and A. Alden, Eds., Boston, Blackwell Scientific Publications, 1989, pp. 239-258.

[61] S. Lewis, Summary of Best practices in Linear Referencing Systems, Chester County: Philadelphia Regional ESRI GIS Transportation Special Interest Group, 2008.

[62] Alexa Crawls, "The Wayback Machine: Elementary-Geometry Trigonometry," R. I. 'Scibor-Marchochi, 01 September 1997. [Online]. Available: https://web.archive.org/web/19991010004728/http://www.geocities.com:80/ResearchTriangle/2363/trig02.html. [Accessed 31 May 2021].

[63] Movable Type Scripts, "Calculate distance, bearing and more between Latitude/Longitude points," Moveable-Type.co.uk, [Online]. Available: http://www.movable-type.co.uk/scripts/latlong.html. [Accessed 16 May 2021].

[64] A. Hedges, "Finding distances based on Latitude and Longitude," Efficacy.net, [Online]. Available: http://efficacy.net/experiments/haversine/. [Accessed 16 May 2021].

[65] C. Rahmig, railML 3.1 Tutorial: Simple Example Step-by-Step, Part 1: Infrastructure, railML.org, 2018.

[66] railML.org, "railML.org - Introduction," [Online]. Available: https://www.railml.org/en/introduction.html. [Accessed 25 June 2021].

[67] Chiemgauer Lokalbahn, "The Chiemgauer Lokalbahn Betriebsgesellschaft mbH," Chiemgauer Lokalbahn, [Online]. Available: https://chiemgauer-lokalbahn.com/wir-ueber-uns/betreibergesellschaft/. [Accessed 20 june 2021].

[68] Bob Janssen, "EULYNX Data Prep Introduction", version 1.0, EULYNX, 2019.

[69] EULYNX Consortium, "EULYNX Data preparation model," [Online]. Available: https://eulynx.eu/index.php/dataprep. [Accessed 12 August 2021].

[70] C. Carter, "Great Circle Distances - Computing the distance between two points on the surface of the earth," SiRF White Paper, 2002.

[71] J. A. Bondy and U. S. R. Murty, "Graph theory with applications", New York: Elsevier Science Publishing Co.., Inc, 1976.

[72] railML.org, "railML.org - Introduction - Background" [Online]. Available: https://www.railml.org/en/introduction.html. [Accessed 14 August 2021].

[73] ISO, "ISO/TS 12911:2012(en) Framework for building information modelling (BIM) guidance" [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:ts:12911:ed-1:v1:en. [Accessed 14 August 2021].

[74] ACCA Software S.p.a. , "BIM maturity Levels: from stage 0 to stage 3" [Online]. Available: https://biblus.accasoftware.com/en/bim-maturity-levels-from-stage-0-to-stage-3/. [Accessed 14 August 2021].

[75] A. Dakhil , M. Alshawi , and J. Underwood, "BIM Client Maturity: Literature Review," in 12th International Post-Graduate Research Conference 2015 University of Salford, Manchester, 2015.

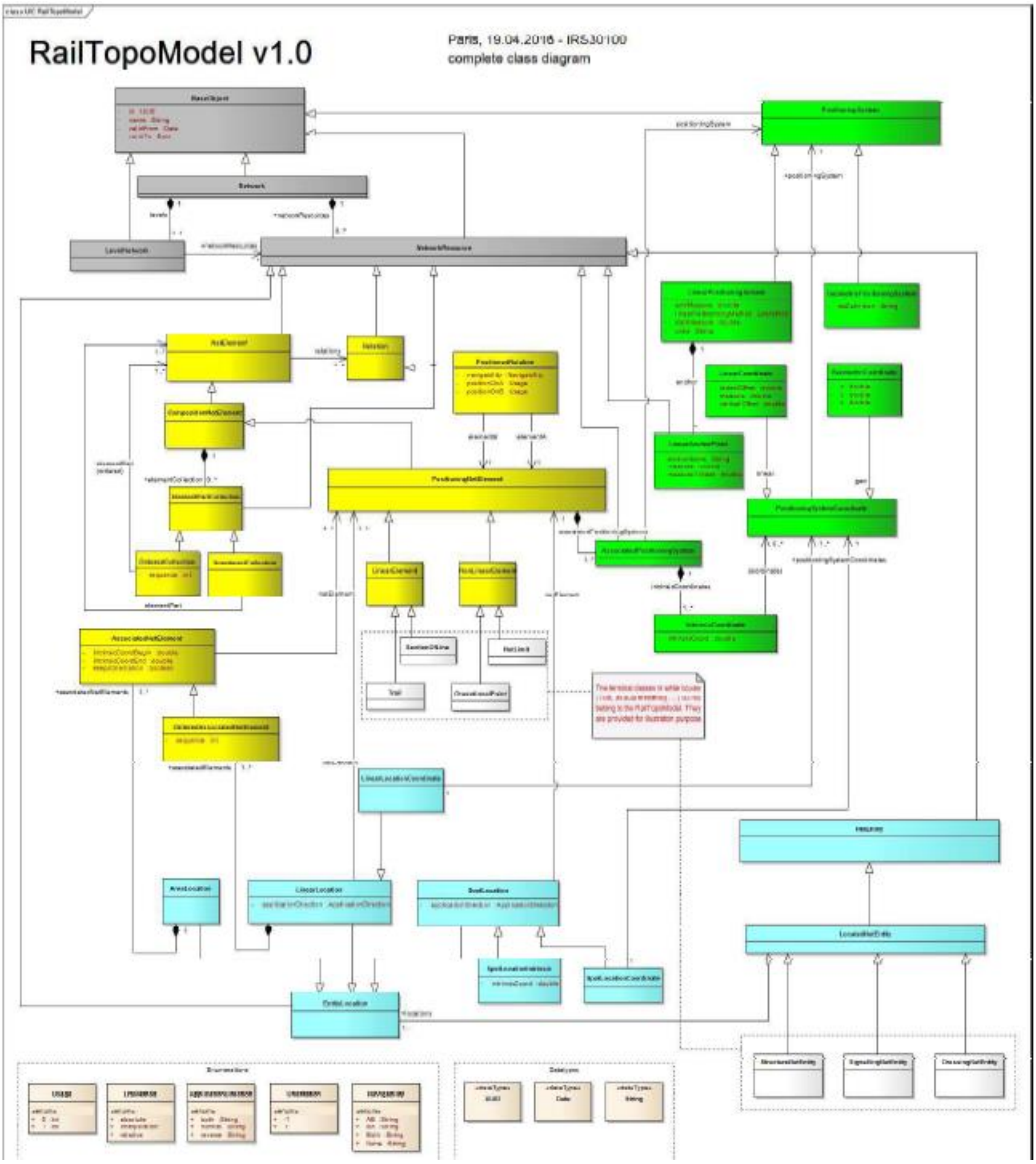
[76] M. Carey, "Human Factors Challenges of Automation in Railway Control," *European Rail Human and Organisational Factors Seminar* - November 2018.

List of figures

List of tables

Appendix

Appendix 1: RailTopoModel complete class diagram



Appendix 2: Overpass query for Obing Station

```
}}
{{OverpassTurboExample|loc=47.99827;12.40351;20|query=
/*
This has been generated by the overpass-turbo wizard.
The original search was:
"railway=* in obing"
*/
[out:json][timeout:25];
// fetch area "obing" to search in
{{(()}}geocodeArea:obing{())}}->.searchArea;
// gather results
(
  // query part for: "railway=*"
  node["railway"](area.searchArea);
  way["railway"](area.searchArea);
  relation["railway"](area.searchArea);
);
// print results
out body;
>;
out skel qt;
}}
```

Appendix 3: GeoJSON data for Obing Station


```
{
  "type": "FeatureCollection",
  "generator": "overpass-ide",
  "copyright": "The data included in this document is from www.openstreetmap.org.
The data is made available under ODbL.",
  "timestamp": "2021-06-26T19:14:28Z",
  "features": [
    {
      "type": "Feature",
      "properties": {
        "@id": "way/904442939",
        "area": "yes",
        "departures_board": "timetable",
        "lit": "yes",
        "public_transport": "platform",
        "railway": "platform",
        "surface": "unpaved",
        "train": "yes"
      },
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [
            [
              12.4030964,
              47.9990395
            ],
            [
              12.4032404,
              47.9987063
            ],
            [
              12.4032605,
              47.9987109
            ],
            [
              12.4032865,
```

```
          47.9987156
        ],
        [
          12.4032305,
          47.9988534
        ],
        [
          12.4031433,
          47.9990486
        ],
        [
          12.4031277,
          47.9990456
        ],
        [
          12.4030964,
          47.9990395
        ]
      ]
    ]
  },
  "id": "way/904442939"
},
{
  "type": "Feature",
  "properties": {
    "@id": "way/62172908",
    "electrified": "no",
    "gauge": "1435",
    "maxspeed": "5",
    "railway": "rail",
    "service": "yard"
  },
  "geometry": {
    "type": "LineString",
    "coordinates": [
      [
        12.4038244,
        47.9974724
      ],
      [
        12.4036889,
        47.9978075
      ],
      [
        12.4033713,
        47.9985926
      ]
    ]
  },
  "id": "way/62172908"
},
{
  "type": "Feature",
  "properties": {
    "@id": "way/160905719",
    "electrified": "no",
    "gauge": "1435",
    "maxspeed": "5",
    "railway": "rail",
    "service": "crossover"
  },
  "geometry": {
    "type": "LineString",
    "coordinates": [
      [
        12.4033707,
```

```
          47.9983509
        ],
        [
          12.4034647,
          47.997969
        ]
      ]
    },
    "id": "way/160905719"
  },
  {
    "type": "Feature",
    "properties": {
      "@id": "way/160905721",
      "electrified": "no",
      "gauge": "1435",
      "maxspeed": "5",
      "railway": "rail",
      "service": "yard"
    },
    "geometry": {
      "type": "LineString",
      "coordinates": [
        [
          12.4030543,
          47.9989226
        ],
        [
          12.4034647,
          47.997969
        ],
        [
          12.4036691,
          47.9974938
        ]
      ]
    },
    "id": "way/160905721"
  },
  {
    "type": "Feature",
    "properties": {
      "@id": "way/184572449",
      "electrified": "no",
      "gauge": "1435",
      "maxspeed": "50",
      "name": "Bad Endorf-Obing",
      "operator": "Chiemgauer Lokalbahn CLB",
      "railway": "rail",
      "railway:preserved": "yes",
      "source": "Bing",
      "usage": "branch",
      "wikidata": "Q801879",
      "wikipedia": "de:Bahnstrecke Bad Endorf-Obing"
    },
    "geometry": {
      "type": "LineString",
      "coordinates": [
        [
          12.3768487,
          47.9867784
        ],
        [
          12.382348,
          47.9870248
        ],
        [
```

```
        12.3841809,
        47.987107
      ],
      [
        12.3848452,
        47.9871343
      ],
      [
        12.3852072,
        47.9871548
      ],
      [
        12.3853825,
        47.9871698
      ],
      [
        12.3855551,
        47.9871866
      ],
      [
        12.3856654,
        47.9871974
      ],
      [
        12.3859438,
        47.9872403
      ],
      [
        12.3860773,
        47.9872702
      ],
      [
        12.3861997,
        47.9873094
      ],
      [
        12.3866953,
        47.9875061
      ],
      [
        12.3871266,
        47.9877496
      ],
      [
        12.3874358,
        47.9879281
      ],
      [
        12.3876969,
        47.9880788
      ],
      [
        12.3881536,
        47.9883138
      ],
      [
        12.3887687,
        47.9884831
      ],
      [
        12.3894108,
        47.9885886
      ],
      [
        12.3905652,
        47.9887617
      ],
```

```
[
  12.3910218,
  47.9888762
],
[
  12.3914965,
  47.9890502
],
[
  12.3917155,
  47.9891617
],
[
  12.3922985,
  47.9894585
],
[
  12.3931954,
  47.989898
],
[
  12.3941562,
  47.9904157
],
[
  12.3943236,
  47.9905291
],
[
  12.3952458,
  47.9911507
],
[
  12.3966742,
  47.9921842
],
[
  12.3968419,
  47.992285
],
[
  12.3970096,
  47.9923815
],
[
  12.3971365,
  47.992444
],
[
  12.3972507,
  47.9925001
],
[
  12.3974938,
  47.9926138
],
[
  12.3976062,
  47.9926532
],
[
  12.3977185,
  47.9926863
],
[
  12.3979493,
  47.9927485
```

```
],
[
  12.3984352,
  47.992838
],
[
  12.3997576,
  47.9930742
],
[
  12.4000639,
  47.9931445
],
[
  12.4003445,
  47.993232
],
[
  12.4006829,
  47.9933613
],
[
  12.4008876,
  47.9934629
],
[
  12.4010694,
  47.9935645
],
[
  12.4031295,
  47.9949292
],
[
  12.4032244,
  47.9949897
],
[
  12.4033139,
  47.9950514
],
[
  12.4033624,
  47.9950861
],
[
  12.4034111,
  47.9951279
],
[
  12.4034713,
  47.9951817
],
[
  12.4035032,
  47.9952194
],
[
  12.403542,
  47.995257
],
[
  12.4036023,
  47.9953323
],
[
  12.4036583,
```

```
          47.9954076
        ],
        [
          12.4037145,
          47.9954829
        ],
        [
          12.4037856,
          47.9955898
        ],
        [
          12.4038219,
          47.9956446
        ],
        [
          12.4038523,
          47.9956994
        ],
        [
          12.4038807,
          47.9957517
        ],
        [
          12.4039032,
          47.995804
        ],
        [
          12.4039266,
          47.9958563
        ],
        [
          12.4039472,
          47.9959086
        ],
        [
          12.4039854,
          47.9960154
        ],
        [
          12.4040159,
          47.9961222
        ],
        [
          12.4040304,
          47.9961714
        ],
        [
          12.4040391,
          47.9962218
        ],
        [
          12.4040488,
          47.996272
        ],
        [
          12.4040525,
          47.9963222
        ]
      ]
    },
    "id": "way/184572449"
  },
  {
    "type": "Feature",
    "properties": {
      "@id": "way/904208962",
      "electrified": "no",
```

```
      "gauge": "1435",
      "maxspeed:backward": "40",
      "maxspeed:forward": "25",
      "name": "Bad Endorf-Obing",
      "operator": "Chiemgauer Lokalbahn CLB",
      "railway": "rail",
      "railway:preserved": "yes",
      "source": "Bing",
      "usage": "branch",
      "wikidata": "Q801879",
      "wikipedia": "de:Bahnstrecke Bad Endorf-Obing"
    },
    "geometry": {
      "type": "LineString",
      "coordinates": [
        [
          12.4038586,
          47.9971955
        ],
        [
          12.4037419,
          47.997472
        ],
        [
          12.4033707,
          47.9983509
        ],
        [
          12.4031738,
          47.9988171
        ],
        [
          12.4030826,
          47.999033
        ]
      ]
    },
    "id": "way/904208962"
  },
  {
    "type": "Feature",
    "properties": {
      "@id": "way/904442941",
      "electrified": "no",
      "gauge": "1435",
      "maxspeed": "5",
      "railway": "rail",
      "service": "crossover"
    },
    "geometry": {
      "type": "LineString",
      "coordinates": [
        [
          12.4037419,
          47.997472
        ],
        [
          12.4036889,
          47.9978075
        ]
      ]
    },
    "id": "way/904442941"
  },
  {
    "type": "Feature",
    "properties": {
```

```
        "@id": "way/904564456",
        "electrified": "no",
        "gauge": "1435",
        "maxspeed": "5",
        "railway": "rail",
        "service": "yard"
      },
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [
            12.4036691,
            47.9974938
          ],
          [
            12.4036954,
            47.9974409
          ],
          [
            12.4038586,
            47.9971955
          ]
        ]
      },
      "id": "way/904564456"
    },
    {
      "type": "Feature",
      "properties": {
        "@id": "way/904564457",
        "electrified": "no",
        "gauge": "1435",
        "maxspeed:backward": "40",
        "maxspeed:forward": "25",
        "name": "Bad Endorf-Obing",
        "operator": "Chiemgauer Lokalbahn CLB",
        "railway": "rail",
        "railway:preserved": "yes",
        "source": "Bing",
        "usage": "branch",
        "wikidata": "Q801879",
        "wikipedia": "de:Bahnstrecke Bad Endorf-Obing"
      },
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [
            12.4040525,
            47.9963222
          ],
          [
            12.4040549,
            47.9964348
          ],
          [
            12.4040561,
            47.9964905
          ],
          [
            12.4040534,
            47.9965462
          ],
          [
            12.4040533,
            47.9966066
          ],
          [
```

```
          12.4040454,
          47.996667
        ],
        [
          12.4040332,
          47.9967281
        ],
        [
          12.4040191,
          47.9967878
        ],
        [
          12.4039897,
          47.9968914
        ],
        [
          12.4039525,
          47.9969933
        ],
        [
          12.4039081,
          47.9970962
        ],
        [
          12.4038586,
          47.9971955
        ]
      ]
    },
    "id": "way/904564457"
  },
  {
    "type": "Feature",
    "properties": {
      "@id": "node/775618569",
      "railway": "buffer_stop"
    },
    "geometry": {
      "type": "Point",
      "coordinates": [
        12.4033713,
        47.9985926
      ]
    },
    "id": "node/775618569"
  },
  {
    "type": "Feature",
    "properties": {
      "@id": "node/1640183905",
      "internet_access": "no",
      "name": "Obing",
      "operator": "Chiemgauer Lokalbahn",
      "public_transport": "station",
      "railway": "station",
      "toilets:wheelchair": "no",
      "train": "yes",
      "usage": "tourism",
      "wheelchair": "no"
    },
    "geometry": {
      "type": "Point",
      "coordinates": [
        12.4031738,
        47.9988171
      ]
    },
```

```
    "id": "node/1640183905"
  },
  {
    "type": "Feature",
    "properties": {
      "@id": "node/1640183908",
      "railway": "buffer_stop;signal",
      "railway:signal:direction": "forward",
      "railway:signal:minor": "DE-ESO:sh2",
      "railway:signal:minor:form": "sign",
      "railway:signal:minor:height": "normal",
      "railway:signal:position": "in_track"
    },
    "geometry": {
      "type": "Point",
      "coordinates": [
        12.4030826,
        47.999033
      ]
    },
    "id": "node/1640183908"
  },
  {
    "type": "Feature",
    "properties": {
      "@id": "node/1728793636",
      "railway": "switch"
    },
    "geometry": {
      "type": "Point",
      "coordinates": [
        12.4038586,
        47.9971955
      ]
    },
    "id": "node/1728793636"
  },
  {
    "type": "Feature",
    "properties": {
      "@id": "node/1728793642",
      "railway": "buffer_stop"
    },
    "geometry": {
      "type": "Point",
      "coordinates": [
        12.4030543,
        47.9989226
      ]
    },
    "id": "node/1728793642"
  },
  {
    "type": "Feature",
    "properties": {
      "@id": "node/8399675375",
      "railway": "switch"
    },
    "geometry": {
      "type": "Point",
      "coordinates": [
        12.4033707,
        47.9983509
      ]
    },
    "id": "node/8399675375"
  },
```

```
    {
      "type": "Feature",
      "properties": {
        "@id": "node/8399675376",
        "railway": "switch"
      },
      "geometry": {
        "type": "Point",
        "coordinates": [
          12.4034647,
          47.997969
        ]
      },
      "id": "node/8399675376"
    },
    {
      "type": "Feature",
      "properties": {
        "@id": "node/8399675377",
        "railway": "switch"
      },
      "geometry": {
        "type": "Point",
        "coordinates": [
          12.4037419,
          47.997472
        ]
      },
      "id": "node/8399675377"
    },
    {
      "type": "Feature",
      "properties": {
        "@id": "node/8399675378",
        "railway": "switch"
      },
      "geometry": {
        "type": "Point",
        "coordinates": [
          12.4036889,
          47.9978075
        ]
      },
      "id": "node/8399675378"
    }
  ]
}
```

Appendix 4: Complete Obing data in railML 3.1

```
<?xml version="1.0" encoding="utf-8"?>
<railML xmlns="https://www.railml.org/schemas/3.1" xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:gid="http://www.gpsinfradat.de/schemas/gid" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="https://www.railml.org/schemas/3.1
https://www.railml.org/schemas/railml-3.1/railml3.xsd" version="3.1">
 <metadata>
  <dc:format>3.1</dc:format>
  <dc:identifier>2</dc:identifier>
  <dc:source>The data behind this model is from www.openstreemap.org and generated by overpass-
turbo.eu</dc:source>
  <dc:title xml:lang="en">Obing Station in railML 3.1</dc:title>
  <dc:language>en</dc:language>
  <dc:date>2021-05-10T14:35:01Z</dc:date>
  <dc:creator xml:lang="ig-NG">Chinenye Azubuike</dc:creator>
```

```xml
  <dc:description>This file has been coded manually and therefore may not be free of errors. Several assumptions are also made in this file when necessary</dc:description>
  <dc:rights>Copyright (c) Chinenye Azubuike, Institute of Rail Vehicles and Transport Systems (ifs) RWTH Aachen, Germany. All Rights Reserved</dc:rights>
 </metadata>
 <common id="co_01">
  <organizationalUnits>
   <infrastructureManager id="im_01" code="RSE" />
  </organizationalUnits>
  <positioning>
   <geometricPositioningSystems>
    <geometricPositioningSystem id="gps01" crsDefinition="epsg:4326">
     <name name="WGS84" language="en" />
     <isValid from="2021-01-01" to="2021-08-18" />
    </geometricPositioningSystem>
   </geometricPositioningSystems>
   <linearPositioningSystems>
    <linearPositioningSystem id="lps01" units="m" startMeasure="0.0" endMeasure="306.6" linearReferencingMethod="relative">
     <name name="railway obing station" language="en" />
     <isValid from="2021-01-01" to="2021-12-31" />
    </linearPositioningSystem>
   </linearPositioningSystems>
  </positioning>
 </common>
 <infrastructure id="is_01">
  <topology>
   <netElements>
    <netElement id="ne_a001" length="79">
     <relation ref="nr_a001a006" />
     <relation ref="nr_a001a003" />
     <associatedPositioningSystem id="ne_a001_aps01">
      <intrinsicCoordinate id="ne_a001_aps01_ic01" intrinsicCoord="0">
       <linearCoordinate positioningSystemRef="lps01" measure="0.0" />
       <geometricCoordinate positioningSystemRef="gps01" x="12.4030826" y="47.9990330" />
      </intrinsicCoordinate>
      <intrinsicCoordinate id="ne_a001_aps01_ic02" intrinsicCoord="1">
       <linearCoordinate positioningSystemRef="lps01" measure="79" />
       <geometricCoordinate positioningSystemRef="gps01" x="12.4033707" y="47.9983509" />
      </intrinsicCoordinate>
     </associatedPositioningSystem>
    </netElement>
    <netElement id="ne_a002" length="110.4">
     <relation ref="nr_a002a004" />
     <relation ref="nr_a002a003" />
     <associatedPositioningSystem id="ne_a002_aps01">
      <intrinsicCoordinate id="ne_a002_aps01_ic01" intrinsicCoord="0">
       <linearCoordinate positioningSystemRef="lps01" measure="10" />
       <geometricCoordinate positioningSystemRef="gps01" x="12.4030543" y="47.9989226" />
      </intrinsicCoordinate>
      <intrinsicCoordinate id="ne_a002_aps01_ic02" intrinsicCoord="1">
       <linearCoordinate positioningSystemRef="lps01" measure="120.4" />
       <geometricCoordinate positioningSystemRef="gps01" x="12.4034647" y="47.9979690" />
      </intrinsicCoordinate>
     </associatedPositioningSystem>
    </netElement>
    <netElement id="ne_a003" length="41.4">
     <relation ref="nr_a001a003" />
     <relation ref="nr_a002a003" />
     <relation ref="nr_a003a004" />
     <relation ref="nr_a003a006" />
     <associatedPositioningSystem id="ne_a003_aps01">
      <intrinsicCoordinate id="ne_a003_aps01_ic01" intrinsicCoord="0">
       <linearCoordinate positioningSystemRef="lps01" measure="79" />
       <geometricCoordinate positioningSystemRef="gps01" x="12.4033707"
```

```xml
y="47.9983509"></geometricCoordinate>
      </intrinsicCoordinate>
      <intrinsicCoordinate id="ne_a003_aps01_ic02" intrinsicCoord="1">
       <linearCoordinate positioningSystemRef="lps01" measure="120.4" />
       <geometricCoordinate positioningSystemRef="gps01" x="12.4034647" y="47.9979690" />
      </intrinsicCoordinate>
     </associatedPositioningSystem>
    </netElement>
    <netElement id="ne_a004" length="55.3">
     <relation ref="nr_a002a004" />
     <relation ref="nr_a003a004" />
     <relation ref="nr_a004a009" />
     <associatedPositioningSystem id="ne_a004_aps01">
      <intrinsicCoordinate id="ne_a004_aps01_ic01" intrinsicCoord="0">
       <linearCoordinate positioningSystemRef="lps01" measure="120.4" />
       <geometricCoordinate positioningSystemRef="gps01" x="12.4034647" y="47.9979690" />
      </intrinsicCoordinate>
      <intrinsicCoordinate id="ne_a004_aps01_ic02" intrinsicCoord="1">
       <linearCoordinate positioningSystemRef="lps01" measure="175.7" />
       <geometricCoordinate positioningSystemRef="gps01" x="12.4036691" y="47.9974938" />
      </intrinsicCoordinate>
     </associatedPositioningSystem>
    </netElement>
    <netElement id="ne_a005" length="89.2">
     <relation ref="nr_a005a007" />
     <relation ref="nr_a005a008" />
     <associatedPositioningSystem id="ne_a005_aps01">
      <intrinsicCoordinate id="ne_a005_aps01_ic01" intrinsicCoord="0">
       <linearCoordinate positioningSystemRef="lps01" measure="56.4" />
       <geometricCoordinate positioningSystemRef="gps01" x="12.4033713" y="47.9985926" />
      </intrinsicCoordinate>
      <intrinsicCoordinate id="ne_a005_aps01_ic02" intrinsicCoord="1">
       <linearCoordinate positioningSystemRef="lps01" measure="145.6" />
       <geometricCoordinate positioningSystemRef="gps01" x="12.4036889" y="47.9978075" />
      </intrinsicCoordinate>
     </associatedPositioningSystem>
    </netElement>
    <netElement id="ne_a006" length="101.8">
     <relation ref="nr_a001a006" />
     <relation ref="nr_a006a013" />
     <relation ref="nr_a006a007" />
     <associatedPositioningSystem id="ne_a006_aps01">
      <intrinsicCoordinate id="ne_a006_aps01_ic01" intrinsicCoord="0">
       <linearCoordinate positioningSystemRef="lps01" measure="79" />
       <geometricCoordinate positioningSystemRef="gps01" x="12.4033707" y="47.9983509" />
      </intrinsicCoordinate>
      <intrinsicCoordinate id="ne_a006_aps01_ic02" intrinsicCoord="1">
       <linearCoordinate positioningSystemRef="lps01" measure="180.8" />
       <geometricCoordinate positioningSystemRef="gps01" x="12.4037419" y="47.9974720" />
      </intrinsicCoordinate>
     </associatedPositioningSystem>
    </netElement>
    <netElement id="ne_a007" length="35.2">
     <relation ref="nr_a005a007" />
     <relation ref="nr_a006a007" />
     <relation ref="nr_a007a008" />
     <relation ref="nr_a007a013" />
     <associatedPositioningSystem id="ne_a007_aps01">
      <intrinsicCoordinate id="ne_a007_aps01_ic01" intrinsicCoord="0">
       <linearCoordinate positioningSystemRef="lps01" measure="145.6" />
       <geometricCoordinate positioningSystemRef="gps01" x="12.4036889" y="47.9978075" />
      </intrinsicCoordinate>
      <intrinsicCoordinate id="ne_a007_aps01_ic02" intrinsicCoord="1">
       <linearCoordinate positioningSystemRef="lps01" measure="180.8" />
       <geometricCoordinate positioningSystemRef="gps01" x="12.4037419" y="47.9974720" />
```

```xml
    </intrinsicCoordinate>
   </associatedPositioningSystem>
  </netElement>
  <netElement id="ne_a008" length="38.7">
   <relation ref="nr_a005a008" />
   <relation ref="nr_a007a008" />
   <associatedPositioningSystem id="ne_a008_aps01">
    <intrinsicCoordinate id="ne_a008_aps01_ic01" intrinsicCoord="0">
     <linearCoordinate positioningSystemRef="lps01" measure="145.6" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4036889" y="47.9978075" />
    </intrinsicCoordinate>
    <intrinsicCoordinate id="ne_a008_aps01_ic02" intrinsicCoord="1">
     <linearCoordinate positioningSystemRef="lps01" measure="184.3" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4038244" y="47.9974724" />
    </intrinsicCoordinate>
   </associatedPositioningSystem>
  </netElement>
  <netElement id="ne_a009" length="6.4">
   <relation ref="nr_a004a009" />
   <relation ref="nr_a009a011" />
   <associatedPositioningSystem id="ne_a009_aps01">
    <intrinsicCoordinate id="ne_a009_aps01_ic01" intrinsicCoord="0">
     <linearCoordinate positioningSystemRef="lps01" measure="175.7" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4036691" y="47.9974938" />
    </intrinsicCoordinate>
    <intrinsicCoordinate id="ne_a009_aps01_ic02" intrinsicCoord="1">
     <linearCoordinate positioningSystemRef="lps01" measure="182.1" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4036954" y="47.9974409" />
    </intrinsicCoordinate>
   </associatedPositioningSystem>
  </netElement>
  <netElement id="ne_a011" length="30.6">
   <relation ref="nr_a009a011" />
   <relation ref="nr_a011a012" />
   <relation ref="nr_a011a013" />
   <associatedPositioningSystem id="ne_a011_aps01">
    <intrinsicCoordinate id="ne_a011_aps01_ic01" intrinsicCoord="0">
     <linearCoordinate positioningSystemRef="lps01" measure="182.1" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4036954" y="47.9974409" />
    </intrinsicCoordinate>
    <intrinsicCoordinate id="ne_a011_aps01_ic02" intrinsicCoord="1">
     <linearCoordinate positioningSystemRef="lps01" measure="212.7" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4038586" y="47.9971955" />
    </intrinsicCoordinate>
   </associatedPositioningSystem>
  </netElement>
  <netElement id="ne_a012" length="93.9">
   <relation ref="nr_a011a012" />
   <relation ref="nr_a012a013" />
   <associatedPositioningSystem id="ne_a012_aps01">
    <intrinsicCoordinate id="ne_a012_aps01_ic01" intrinsicCoord="0">
     <linearCoordinate positioningSystemRef="lps01" measure="212.7" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4038586" y="47.9971955" />
    </intrinsicCoordinate>
    <intrinsicCoordinate id="ne_a012_aps01_ic02" intrinsicCoord="1">
     <linearCoordinate positioningSystemRef="lps01" measure="306.6" />
     <geometricCoordinate positioningSystemRef="gps01" x="12.4040525" y="47.9963222" />
    </intrinsicCoordinate>
   </associatedPositioningSystem>
  </netElement>
  <netElement id="ne_a013" length="31.9">
   <relation ref="nr_a006a013" />
   <relation ref="nr_a007a013" />
   <relation ref="nr_a011a013" />
   <relation ref="nr_a012a013" />
```

```xml
    <associatedPositioningSystem id="ne_a013_aps01">
     <intrinsicCoordinate id="ne_a013_aps01_ic01" intrinsicCoord="0">
      <linearCoordinate positioningSystemRef="lps01" measure="180.8" />
      <geometricCoordinate positioningSystemRef="gps01" x="12.4037419" y="47.9974720" />
     </intrinsicCoordinate>
     <intrinsicCoordinate id="ne_a013_aps01_ic02" intrinsicCoord="1">
      <linearCoordinate positioningSystemRef="lps01" measure="212.7" />
      <geometricCoordinate positioningSystemRef="gps01" x="12.4038586" y="47.9971955" />
     </intrinsicCoordinate>
    </associatedPositioningSystem>
   </netElement>
  </netElements>
  <netRelations>
   <netRelation id="nr_a001a006" positionOnA="1" positionOnB="0" navigability="Both">
    <elementA ref="ne_a001" />
    <elementB ref="ne_a006" />
   </netRelation>
   <netRelation id="nr_a001a003" positionOnA="1" positionOnB="0" navigability="Both">
    <elementA ref="ne_a001" />
    <elementB ref="ne_a003" />
   </netRelation>
   <netRelation id="nr_a002a004" positionOnA="1" positionOnB="0" navigability="Both">
    <elementA ref="ne_a002" />
    <elementB ref="ne_a004" />
   </netRelation>
   <netRelation id="nr_a002a003" positionOnA="1" positionOnB="1" navigability="None">
    <elementA ref="ne_a002" />
    <elementB ref="ne_a003" />
   </netRelation>
   <netRelation id="nr_a003a004" positionOnA="1" positionOnB="0" navigability="Both">
    <elementA ref="ne_a003" />
    <elementB ref="ne_a004" />
   </netRelation>
   <netRelation id="nr_a003a006" positionOnA="0" positionOnB="0" navigability="None">
    <elementA ref="ne_a003" />
    <elementB ref="ne_a006" />
   </netRelation>
   <netRelation id="nr_a004a009" positionOnA="1" positionOnB="0" navigability="Both">
    <elementA ref="ne_a004" />
    <elementB ref="ne_a009" />
   </netRelation>
   <netRelation id="nr_a005a007" positionOnA="1" positionOnB="0" navigability="Both">
    <elementA ref="ne_a005" />
    <elementB ref="ne_a007" />
   </netRelation>
   <netRelation id="nr_a005a008" positionOnA="1" positionOnB="0" navigability="Both">
    <elementA ref="ne_a005" />
    <elementB ref="ne_a008" />
   </netRelation>
   <netRelation id="nr_a006a013" positionOnA="1" positionOnB="0" navigability="Both">
    <elementA ref="ne_a006" />
    <elementB ref="ne_a013" />
   </netRelation>
   <netRelation id="nr_a006a007" positionOnA="1" positionOnB="1" navigability="None">
    <elementA ref="ne_a006" />
    <elementB ref="ne_a007" />
   </netRelation>
   <netRelation id="nr_a007a013" positionOnA="1" positionOnB="0" navigability="Both">
    <elementA ref="ne_a007" />
    <elementB ref="ne_a013" />
   </netRelation>
   <netRelation id="nr_a007a008" positionOnA="0" positionOnB="0" navigability="None">
    <elementA ref="ne_a007" />
    <elementB ref="ne_a008" />
   </netRelation>
```

```xml
    <netRelation id="nr_a009a011" positionOnA="1" positionOnB="0" navigability="Both">
     <elementA ref="ne_a009" />
     <elementB ref="ne_a011" />
    </netRelation>
    <netRelation id="nr_a011a012" positionOnA="1" positionOnB="0" navigability="Both">
     <elementA ref="ne_a011" />
     <elementB ref="ne_a012" />
    </netRelation>
    <netRelation id="nr_a011a013" positionOnA="1" positionOnB="1" navigability="None">
     <elementA ref="ne_a011" />
     <elementB ref="ne_a013" />
    </netRelation>
    <netRelation id="nr_a012a013" positionOnA="0" positionOnB="1" navigability="Both">
     <elementA ref="ne_a012" />
     <elementB ref="ne_a013" />
    </netRelation>
   </netRelations>
   <networks>
    <network id="nw01">
     <level id="Lv0" descriptionLevel="Micro">
      <networkResource ref="ne_a001" />
      <networkResource ref="ne_a002" />
      <networkResource ref="ne_a003" />
      <networkResource ref="ne_a004" />
      <networkResource ref="ne_a005" />
      <networkResource ref="ne_a006" />
      <networkResource ref="ne_a007" />
      <networkResource ref="ne_a008" />
      <networkResource ref="ne_a009" />
      <networkResource ref="ne_a011" />
      <networkResource ref="ne_a012" />
      <networkResource ref="ne_a013" />
      <networkResource ref="nr_a001a006" />
      <networkResource ref="nr_a001a003" />
      <networkResource ref="nr_a002a004" />
      <networkResource ref="nr_a002a003" />
      <networkResource ref="nr_a003a004" />
      <networkResource ref="nr_a003a006" />
      <networkResource ref="nr_a004a009" />
      <networkResource ref="nr_a005a007" />
      <networkResource ref="nr_a005a008" />
      <networkResource ref="nr_a006a013" />
      <networkResource ref="nr_a006a007" />
      <networkResource ref="nr_a007a008" />
      <networkResource ref="nr_a007a013" />
      <networkResource ref="nr_a009a011" />
      <networkResource ref="nr_a011a012" />
      <networkResource ref="nr_a011a013" />
      <networkResource ref="nr_a012a013" />
     </level>
    </network>
   </networks>
  </topology>
  <functionalInfrastructure>
   <bufferStops>
    <bufferStop id="bus01" type="fixedBufferStop">
     <spotLocation id="bus01_sloc01" netElementRef="ne_a001" applicationDirection="reverse" pos="0.0">
      <linearCoordinate positioningSystemRef="lps01" measure="0.0" />
      <geometricCoordinate positioningSystemRef="gps01" x="12.4030826" y="47.9990330" />
     </spotLocation>
    </bufferStop>
    <bufferStop id="bus02" type="fixedBufferStop">
     <spotLocation id="bus02_sloc01" netElementRef="ne_a005" applicationDirection="reverse" pos="0.0">
      <linearCoordinate positioningSystemRef="lps01" measure="56.4" />
      <geometricCoordinate positioningSystemRef="gps01" x="12.4033713" y="47.9985926" />
```

```
      </spotLocation>
     </bufferStop>
     <bufferStop id="bus03" type="fixedBufferStop">
      <spotLocation id="bus03_sloc01" netElementRef="ne_a002" applicationDirection="reverse" pos="0.0">
       <linearCoordinate positioningSystemRef="lps01" measure="10" />
       <geometricCoordinate positioningSystemRef="gps01" x="12.4030543" y="47.9989226" />
      </spotLocation>
     </bufferStop>
    </bufferStops>
    <switchesIS>
     <switchIS id="swi01" continueCourse="left" branchCourse="right" type="ordinarySwitch">
      <spotLocation id="swi01_sloc01" netElementRef="ne_a001" applicationDirection="normal" pos="79">
       <linearCoordinate positioningSystemRef="lps01" measure="79" />
       <geometricCoordinate positioningSystemRef="gps01" x="12.4033707" y="47.9983509" />
      </spotLocation>
      <leftBranch netRelationRef="nr_a001a006" branchingSpeed="25" joiningSpeed="40" radius="0" />
      <rightBranch netRelationRef="nr_a001a003" branchingSpeed="5" joiningSpeed="40" radius="100" />
     </switchIS>
     <switchIS id="swi02" continueCourse="left" branchCourse="right" type="ordinarySwitch">
      <spotLocation id="swi02_sloc01" netElementRef="ne_a004" applicationDirection="reverse" pos="0">
       <linearCoordinate positioningSystemRef="lps01" measure="120.4" />
       <geometricCoordinate positioningSystemRef="gps01" x="12.4034647" y="47.9979690" />
      </spotLocation>
      <leftBranch netRelationRef="nr_a002a004" branchingSpeed="5" joiningSpeed="5" radius="0" />
      <rightBranch netRelationRef="nr_a003a004" branchingSpeed="5" joiningSpeed="5" radius="500" />
     </switchIS>
     <switchIS id="swi03" continueCourse="left" branchCourse="right" type="ordinarySwitch">
      <spotLocation id="swi03_sloc01" netElementRef="ne_a005" applicationDirection="normal" pos="89.2">
       <linearCoordinate positioningSystemRef="lps01" measure="145.6" />
       <geometricCoordinate positioningSystemRef="gps01" x="12.4036889" y="47.9978075" />
      </spotLocation>
      <leftBranch netRelationRef="nr_a005a008" branchingSpeed="5" joiningSpeed="5" radius="0" />
      <rightBranch netRelationRef="nr_a005a007" branchingSpeed="5" joiningSpeed="5" radius="100" />
     </switchIS>
     <switchIS id="swi04" continueCourse="left" branchCourse="right" type="ordinarySwitch">
      <spotLocation id="swi04_sloc01" netElementRef="ne_a013" applicationDirection="reverse" pos="0.0">
       <linearCoordinate positioningSystemRef="lps01" measure="180.8" />
       <geometricCoordinate positioningSystemRef="gps01" x="12.4037419" y="47.9974720" />
      </spotLocation>
      <leftBranch netRelationRef="nr_a006a013" branchingSpeed="25" joiningSpeed="40" radius="0" />
      <rightBranch netRelationRef="nr_a007a013" branchingSpeed="5" joiningSpeed="25" radius="100" />
     </switchIS>
     <switchIS id="swi05" continueCourse="right" branchCourse="left" type="ordinarySwitch">
      <spotLocation id="swi05_sloc01" netElementRef="ne_a012" applicationDirection="reverse" pos="0.0">
       <linearCoordinate positioningSystemRef="lps01" measure="212.7" />
       <geometricCoordinate positioningSystemRef="gps01" x="12.4038586" y="47.9971955" />
      </spotLocation>
      <leftBranch netRelationRef="nr_a011a012" branchingSpeed="5" joiningSpeed="25" radius="100" />
      <rightBranch netRelationRef="nr_a012a013" branchingSpeed="25" joiningSpeed="25" radius="0" />
     </switchIS>
    </switchesIS>
    <tracks>
     <track id="badEndorf-Obing" type="mainTrack">
      <linearLocation id="badEndorf-Obing_lloc01" applicationDirection="both">
       <associatedNetElement netElementRef="ne_a001" keepsOrientation="true" posBegin="0.0"
posEnd="79"></associatedNetElement>
       <associatedNetElement netElementRef="ne_a006" keepsOrientation="true" posBegin="0.0"
posEnd="101.8"></associatedNetElement>
       <associatedNetElement netElementRef="ne_a013" keepsOrientation="true" posBegin="0.0"
posEnd="31.9"></associatedNetElement>
      </linearLocation>
      <trackBegin ref="bus01" />
      <trackEnd ref="swi05" />
      <length value="212.7" type="physical" />
     </track>
```

```xml
    <track id="service_yard1-Obing" type="sidingTrack">
     <linearLocation id="service_yard1-Obing_lloc01" applicationDirection="both">
      <associatedNetElement netElementRef="ne_a005" keepsOrientation="true" posBegin="0.0" posEnd="89.2"></associatedNetElement>
     </linearLocation>
     <trackBegin ref="bus02" />
     <trackEnd ref="swi03" />
     <length value="89.2" type="physical" />
    </track>
    <track id="service_yard2-Obing" type="sidingTrack">
     <linearLocation id="service_yard2-Obing_lloc01" applicationDirection="both">
      <associatedNetElement netElementRef="ne_a002" keepsOrientation="true" posBegin="0.0" posEnd="120.4"></associatedNetElement>
      <associatedNetElement netElementRef="ne_a004" keepsOrientation="true" posBegin="0.0" posEnd="92.3"></associatedNetElement>
     </linearLocation>
     <trackBegin ref="bus03" />
     <trackEnd ref="swi05" />
     <length value="202.7" type="physical" />
    </track>
    <track id="connection1-Obing" type="connectingTrack">
     <linearLocation id="connection1-Obing_lloc01" applicationDirection="both">
      <associatedNetElement netElementRef="ne_a007" keepsOrientation="true" posBegin="0.0" posEnd="35.2"></associatedNetElement>
     </linearLocation>
     <trackBegin ref="swi03" />
     <trackEnd ref="swi04" />
     <length value="35.2" type="physical" />
    </track>
    <track id="connection2-Obing" type="connectingTrack">
     <linearLocation id="connection2-Obing_lloc01" applicationDirection="both">
      <associatedNetElement netElementRef="ne_a003" keepsOrientation="true" posBegin="0.0" posEnd="41.4"></associatedNetElement>
     </linearLocation>
     <trackBegin ref="swi01" />
     <trackEnd ref="swi02" />
     <length value="41.4" type="physical" />
    </track>
   </tracks>
  </functionalInfrastructure>
 </infrastructure>
</railML>
```