FP7 Project 2007- Grant agreement n°: 218575

Project Acronym: **INESS**

Project Title: **INtegrated European Signalling System**

Instrument: Large-scale integrating project
Thematic Priority: Transport

Document Title:

INESS_WSC_Deliverable_C2.1_Report on data model definition, needed adaptations and applicability

Due date of deliverable	March 2012
Actual submission date	30.04.2012

Deliverable ID:	**D.C.2.1**
Deliverable Title:	Report on data model definition, needed adaptations and applicability
WP related:	C1 + C2
Responsible partner:	Bombardier
Task/Deliverable leader Name:	Tom Stein
Contributors:	Javier Vicente Fajardo (ADIF), Eduardo Frigolet Ron (ALSTOM), Bertrand Badot (ALSTOM), Marina Ferrando (ANSALDO), Juraj Majtan (AZD), Michal Pavel (AZD), Martin Kroupa (AZD), Tom Stein (Bombardier), Theo Lange (Bombardier), Roland Spannaus (DB), Matthias Grimm (DLR), Florian Brinkmann (DLR), Christian Linder (DLR), Ikedichi Mbakwe (DLR), José Antonio Quintano (Eliop Seinalia), Philipp Stüer (FIR), Carsten Gerke (Funkwerk), Andrew Stuart (Invensys Rail), Paul Curson (Invensys Rail), Neil Barnatt (Network Rail), Daan van der Meij (ProRail), Francesco Cirillo (RFI), Benjamin Lau (Siemens), Eckhard Rimkus (Thales), Geltmar von Buxhoeveden (TU BS), Emmanuel Buseyne (UIC)

Start date of the project: 01-10-2008 Duration: 42 Months

Project coordinator: George Barbu
Project coordinator organisation: UIC

Dissemination Level[1]: PU

[1] PU: Public, PP: Restricted to other programme participants (including the Commission Services), RE: Restricted to a group specified by the consortium (including the Commission Services), CO: Confidential, only for members of the consortium (including the Commission Services).

Document Information

Document type: Deliverable Report
Document Name: INESS WS C Deliverable C.2.1_Report on data model definition, needed adaptations and applicability_SB_Finalised.docx
Author: T. Stein
Dissemination level: PU

Approvals

	Name	Company	Date	Visa
WP leader	Tom Stein Carsten Gerke	Bombardier Funkwerk IT	06.03.12	ok
WS Leader	Tom Stein	Bombardier	06.03.12	ok
Project Manager	Emmanuel Buseyne	UIC	11.03.12	ok
Steering Board	N.A	N.A	16.03.12	ok

Document history

Revision	Date	Modification	Author
1.0	14.12.2011	Draft based on D.C.1.2 v1.4	P. Stüer
1.1	31.01.2012	Included final list of requirements and coverage by RailML	T. Stein et al.
1.2	21.02.2012	Finalised for review by TAB and WS	T. Stein
1.3	01.03.2012	Integrated review comments by TAB	P. Stüer
1.4	06.03.2012	Integrated review comments by WS members, finalised for submission to SB+EC	T. Stein
TAB+PMC_Reviewed_Ver1	11.03.2012	Adapt metadata	E. Buseyne
SB_Finalised	16.03.2012	Adapt metadata	E. Buseyne

TABLE OF CONTENTS

GLOSSARY AND ABBREVIATIONS

Abbreviation	Meaning
CENELEC	Comité Européen de Normalisation Électrotechnique (European Committee for Electrotechnical Standardisation)
DM	Data Model
DTD	Document Type Definition
ERTMS	European Rail Traffic Management System
ETCS	European Train Control System
EUDRI	European Unified Description of Railway Infrastructures
HAT	Headway Analysis Tool
IDAFF	Interlocking Data File Format
INESS	INtegrated European Signalling System
ISP	Infrastructure plan
LCC	Life Cycle Costs
R&D	Research and Development
SA	Specific Application
SDEF	Signalling Data Exchange Format
SSADS	Signalling Schemes Asset Data Store
SSIM	Signalling Schemes Image Models
SoA	State of the Art
SORAT	Signal Overrun Risk Assessment Trial
STPE	Signalling Telecoms Programme Engineering
WS	Work Stream
XML	eXtensible Markup Language
TRU	Track Recording Unit
UIC	Union Internationale des Chemins de fer (International Union of Railways)

Section 1 – EXECUTIVE SUMMARY

1.1 Task

The main objective of INESS workstream C is to describe a European Unified Data Model for Railway Infrastructures (EUDRI) for the data flow into an interlocking system. Existing complex processes, design tools and interlocking configuration tools, that are specific to railway infrastructure maintainers or suppliers, will not be renewed by INESS. Therefore, this data file format EUDRI has to be a harmonized data model compatible with existing design and interlocking configuration tools. It shall enable data transfer for invitation of tenders, production and implementation of an INESS compliant interlocking, thus supporting the data flows linked with the INESS system architecture. The contained data can also be used to maximise the knowledge base of owned assets within the railway infrastructure.

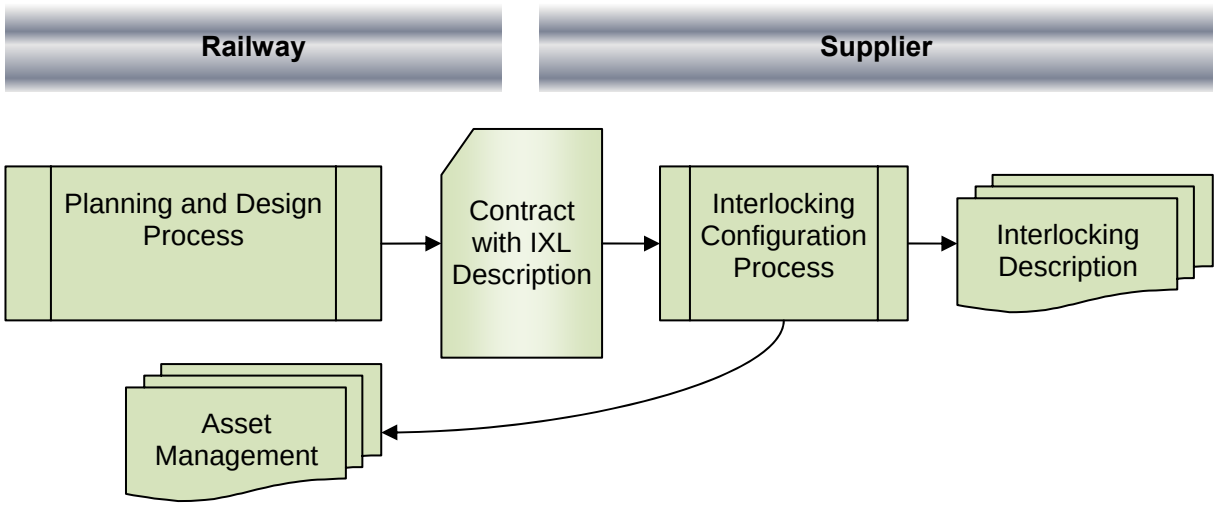

Figure 1. Legacy Interlocking Design Process

1.2 Approach

The INESS project did not tend to re-invent conventional interlockings. Therefore, a system design for a conventional interlocking (or the conventional part of an INESS interlocking) did not seem to be useful. Re-inventing processes and existing tools for the design of an interlocking on the railways' side was not useful or intended. Instead, the design of a data model for exchanging data between existing design tools (especially for exchanging data with interlocking configuration tools) was targeted. Rather than creating a new data model, WS C selected an existing state of art data model.

The deliverable describes six data models for the specification of technical interlocking systems. It investigates also the data models at their current status and version. The selection of the data models took place at several WS meetings. The information about each data model was provided by an expert of the respective data model. The data model documentation can be also found on Myndsphere platform here:

https://www.myndsphere.com/gm/document-1.9.231927/INESS_WSC_Deliverable_DC11_DataModelDescription_100.doc

At the end of each section, the link to the relevant documents is mentioned, so it is possible to access more information about the selected data models, if needed.

Furthermore, this document describes the data model comparison process followed. The goal of the comparison process was to determine, in a non-biased and objective way, a ranking of the proposed data models, to give arguments and simplify the selection of one of the most appropriate European standard data models.

The generic steps to achieve the selection of the data model were:

- **Model Evaluation & Comparison Process:**
  - Description of the proposed data models and complete documentation
  - List of requirements agreed by all the workstream members, including requirements for objects from the extended core requirements of WS D.
  - Development of a checklist, which includes weights for every requirement, and the criticality of some requirements, expressed as “Knock-out requirements”.
  - Comparison of six pre-evaluated state of art data models against the requirements checklist (in groups).
  - Review of criteria followed for requirement scoring (when necessary).
  - Preparation of references of the requirements in the Data Model documentation and a list of high-level gaps.

- **Generation of overall results and results for every detected requirements category (for each data model).**

- **Model Selection Process:**
  - Individual analysis of the results.
  - Voting and justification.

- **Trial:**
  - The list of requirements for the data model was put to trial in order to find a detailed list of gaps and to verify the used requirements

1.3 Results

Because of this methodology, the EUDRI data model contains all objects relevant for an interlocking including ETCS, which includes legacy areas, like signals, points, and tracks, as well as infrastructure areas like length of tracks, gradients, positions. EUDRI has a backbone functionality, that means it links railways to suppliers (like it is used in bids and contracts) and back (informing the railway about the installed assets) and even can link railways across borders by unified concepts. EUDRI uses open standards based on XML, enabling to use (most of) existing databases to export, share and import data in EUDRI format.

As no existing data model had been designed for INESS’ purposes, after selecting the best fitting data model RailML and describing already known (major) gaps, the requirements were put to trial by suppliers and railway companies. The result of this is a very precise list of gaps and a verification of the work done so far.

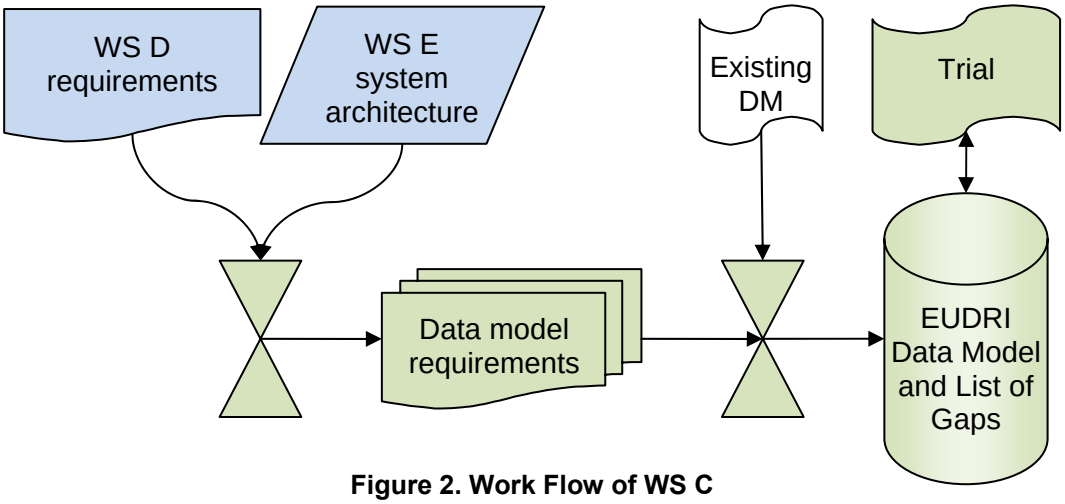

Figure 2. Work Flow of WS C

During this trial a bunch of details have been identified that need to be improved in RailML to be usable for the intended purpose of EUDRI. Therefore, the selected data model will have to be enhanced as described by the requirements found by this workstream. For starting the data model improvement and maintaining the data model in the future, a process and funding have to be provided. As the data currently used to describe interlockings by usage or standards is defined by the railway infrastructure managers, it is up to them to keep their existing interface (whether it is on paper or already a national or railway specific data model) or to use an international or at least European standard. If they decide to use EUDRI, the benefit would be

- A wider scope of tools, even from different markets, that can be used to build the interlocking description,
- faster reaction on tenders,
- faster implementation of interlockings,
- less mistakes in the implementation of interlockings,
- by that reduced costs for tenders and implementation,
- better quality of data describing the used assets, and
- by that reduced costs for maintenance.

All this could be achieved simply because the suppliers do not need to rewrite the interlocking description in order to get it into their tools for calculation and implementation, and railway infrastructure managers do not need to rewrite the description of the assets used by the new interlocking to get those assets into their maintenance tools. So it is up to the railway infrastructure managers to start using EUDRI (and claiming the benefits).

None of the benefits in itself can reduce the overall costs of an interlocking in a significant way. This is one of the important results of workstream B. As an interlocking is a complex system surrounded by hundreds of processes, only the significant optimisation of every process will reduce the overall costs significantly. EUDRI is a method to reduce those costs driven by describing the interlocking, either driven by (existing or future) tools on a railway infrastructure managers side or by reading the description on a suppliers side. As shown by workstream B, it is up to the railway infrastructure managers to drive this change and as a first step add resources to the RailML consortium to fill the gaps that are described at the end of this report. In a near future, this will result in RailML being useful as EUDRI.

1.4 Outlook

One important aspect concerning the future use of the RailML data model is maintenance. As mentioned before, it is important that the data model except of being "standard", also is used and maintained by independent organizations and institutes. RailML has the advantage of being already in industrial use, supported and developed by independent businesses and institutions from various European countries. This provides the highest potential concerning the exploitability of RailML.

Main aspects for further technical development of the RailML data model have been described in Section 6. Further development of the RailML data model in the listed requirements would positively impact exploitability of the model.

SECTION 2 – INTRODUCTION

Purpose of INESS WS C is not re-inventing conventional interlockings. The objective of WS C is to improve the data transfer from infrastructure design systems of the railways infrastructure managers to interlocking design system of the suppliers. Therefore the focus of WS C is on digital descriptions of interlockings by use of data models. Instead of “inventing” a new data model, WS C has selected one of the existing data models already defined and (partly) in use. Therefore existing models have been checked against a list of requirements to find the best fitting state of the art (SoA) data model and to describe still existing weaknesses of this data model which have to be eliminated in future.

WS C started trying to get an overview on SoA data models to get to know how they are used and what their purpose is. Then data model requirements for an EUDRI data model were defined, starting with the use of existing data models and the generic requirements of WS D. The SoA data models have been compared to these data model requirements and evaluated. In addition, the selected data model was put to trial at different suppliers and railways to evaluate it.

The detailed methodology steps used to achieve the trial of the data model were as follows:

1. Preparation of the work, finding of suitable SoA data models and definition of requirements‘ objectives (M25-M31)
2. Specification of the EUDRI data model requirements (M31-33)
3. Description of the needed tooling (M32-33)
4. Setting up a of checklist for the comparison of data models against the EUDRI requirements (M32-34)
5. Definition of a way to select the appropriate data model (M32-34)
6. Collection of descriptions of SoA data models (M31-34)
7. Comparison of the SoA data models against the requirements checklist (M33-35)
8. Preparation and assessment in case more than 1 suitable data model suitable is found (M36)
9. Selection of the data model (Milestone, M36)
10. Description of the next steps for adapting the data model to the INESS information framework used by suppliers and railways. (M35-39)
11. Trial of the data model by suppliers and railways

Section 3 – DESCRIPTION OF DATA MODELS (WP.C.1)

As mentioned above, one of the objectives of this deliverable is to give a description of the data models. The information about each one of them was provided by an expert of the respective data model. The description of the data models is the first and very important step toward the selection process. Furthermore the general aspects for each one of the data models are provided in the form of a table. Each one of the tables contains basic characteristics describing the data model.

The description of the data models was not harmonized in order to give a first overview on the most important parts of the data models from the view of some experts of each data model. This was later used to derive requirements for all data models. For the same reason, there is no example of each data model, as a small example would not help to describe the complexity or maturity of a data model.

For comparing the data models high-level criteria have been defined and a qualitative estimation of how well they are fulfilled is given.

3.1 Euro-Interlocking Data Model

3.1.1 Description of the Euro-Interlocking Data Model

The Euro-Interlocking data model has been developed as part of the Euro interlocking project. It is mentioned by UIC that the Euro-Interlocking project launched in 1999, when the new ETCS with its three levels of application began to become reality and when it was foreseeable that similar efforts for harmonization and joint development would also become necessary in the adjacent signalling environment. From the beginning the reductions of life-cycle costs, improved performance and optimal compliance with ETCS have been the main targets.

The Euro-Interlocking Data Preparation work was created in order to further develop the standardization of data exchange file formats for interlocking applications. The Euro-Interlocking Project has contributed to the development of standardized file formats for interlocking data exchange. These standardized means of exchanging information between different phases of the data preparation process are needed to fulfil the CENELEC (European Committee for Electro technical Standardization) process for safety and reliability. Euro-Interlocking file format standards describe a common European structure and framework for project and configuration data transfer between data preparation tools and for data exchange between railways and suppliers.

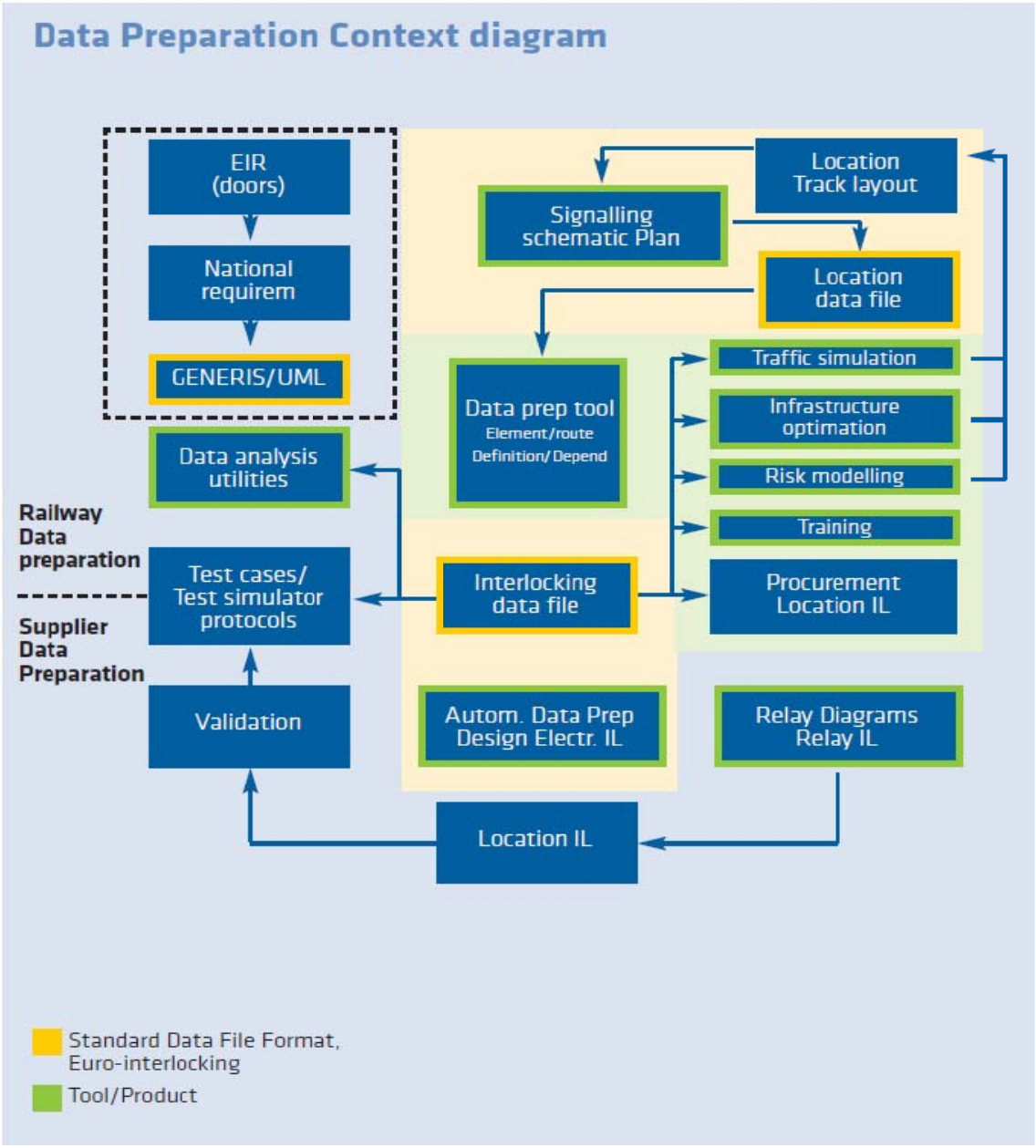

Figure 3. Euro Interlocking Data Preparation Context diagram

The Excel spreadsheet which is called the ''Element Description File'' contains specific elements of the Euro-Interlocking data file format with their attributes and properties.

The IDAFF itself is an XML-format with the syntax defined using the technique of Document Type Definition (DTD). Since XML is a quasi standard in software technologies this allows developing applications for IDAFF files quite easily. Nevertheless the DTD technique is a little outdated and mainly substituted by XML schema definition, but still a lot of frameworks support DTD.

The Interlocking Data File itself contains details of topology, individual system elements, and route dependencies. The file consists of two general levels:

- A project level with:
    - project data and
    - basic header data
- An element level with:
    - geographic data
    - signalling element data
    - route information data

Geographic data are signalling-independent data. This includes topology data such as gradients and bridges as well as operational data such tracks and lines.

Signalling element data comprises all wayside equipment elements, including train detection sections and points. Route information data include interlocking routes (including shunting routes) and shunting areas.

All the elements with all their attributes included in the IDAFF are described in detail in an excel-sheet. The meaning of the columns in the excel sheet is as follows:

- group — information needed for the DTD file
- attribute, zero level — attributes (standard in all folders)
- attribute, first level — element relating to special attributes
- attribute, second level — element relating to special attributes
- attribute, third level — element relating to special attributes
- explanation — comment to explain the content of the attributes
- multiplicity — multiple attributes possible, or single only?
- mandatory — attribute is mandatory, or optional?
- syntax — syntax description
- example data — example for explanation purposes

Due to the limitations of the DTD not all the restrictions from the excel sheet are transferred into the DTD-file itself. Therefore not all information contained in the excel sheet is technically accessible to applications via the DTD.

The data model allows the extension of objects with specific parameters to meet e.g. specific operational requirements or other railway specific information.

The Euro-Interlocking project ended in 2008 and the results were published in a baseline 8.0. In a way Euro-Interlocking is succeeded by the INESS project.

The Euro-Interlocking file formats are currently used by the ÖBB in their interlocking planning process. This proves the applicability of the IDAFF as a data model especially for interlocking systems.

3.1.2 General Aspects of the Euro-Interlocking Data Model

Table 1: Summary of the Euro-Interlocking Data Model

Criteria	Assessment	Comment
Applicability to railway applications	Very High	Data model was specifically developed to meet the requirements of an interlocking system, dedicated data types for the interlocking domain
Technical basis	XML	Data model based on XML, data model and data types described using a DTD, open to extensions for specific needs
Already in industrial use	Yes	Used in Austria by ÖBB and some railway suppliers, data preparation tool BEST
Development Group	project finished	Project finished with Baseline 8.0. In the project phase participation of railways and suppliers as well as scientific players

Criteria	**Assessment**	**Comment**
Railways partners in Euro-Interlocking	*19*	ADIF,Spain Banestyrelsen, Denmark Banverket, Sweden BLS, Switzerland CD, Czech Republic CFR, Romania DB, Germany JBV, Norway MAV, Hungary NS-Pro Rail, The Netherlands PKP, Poland Network Rail, UK REFER, Portugal RFF, France RFI, Italy RHK, Finland SBB/CFF, Switzerland SNCB/NMBS, Belgium SZ, Slovenia
Industry partners in Euro-Interlocking	12	AEA Technology Rail Thales Alstom Transport Ansaldo AZD Praha Bombardier Transportation I-logix Invensys Rail Systems OFFIS/OSC Siemens Transportation Systems Telelogic Vossloh Information Technologies
Development Status	Baseline 8.0	Final version, no further development in progress
License Model		Open to INESS members

Criteria	Assessment	Comment
Expected costs		Depending on license model, probably none
Documentation	Documentation to Baseline 8.0 available on myndsphere	Introduction as PDF-document, detailed description for each object and attribute in excel-sheet
Project maintenance and progress	Project finished	No further development from the Euro-Interlocking project
Modularity	Partly	Objects are conceptually divided in different levels but all included in one DTD
Limitations	Some	e.g. definitions in DTD, no conceptual data model

3.2 RailML

3.2.1 Description of the RailML Data Model

Because of the increasing number of simulation and operating programmes, the development of individual interfaces will become impractical. In order to face that problem RailML has been developed using the XML (eXtensible Markup Language) to simplify data transfer through the use of a common data structure which is a simple and efficient way to transfer data between applications. Transferring data between two particular programmes requires that two different interfaces are developed (one for each direction of transfer). As the number of programmes increases the number of possible data transfers increases significantly.

RailML is a generic language that can be used to describe railway-related data. The language has been divided into sub-formats (or schemes) for particular types of railway data. RailML consortium partners are currently developing data structures for the three main types of data files used in railway simulation: timetable (schedule), infrastructure (with subsets for lines and stations), and rolling stock. At present, the RailML timetable structure is most developed and has already been incorporated into several applications (e.g. Open Track, FBS, Viriato and Open Time Table, information can be found in the internet). A significant amount of work has been completed on both the infrastructure and rolling stock schemes, but these have not yet been finalized. Several additional schemes have been suggested. As a web-based development partnership, each scheme has its own newsgroup open for discussion. The newsgroups' purpose is two-fold: firstly, newsgroup partners exchange information and secondly, knowledge and processing of definitions is discussed and archived. RailML.org also supports a newsgroup for general exchange and discussions that are not scheme-related.

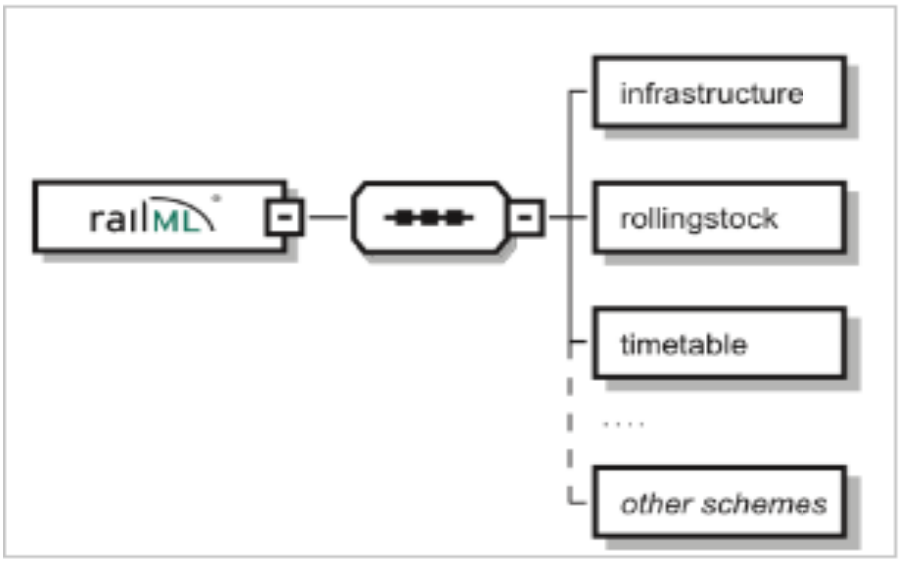

Figure 4. RailML sub-formats

RailML is an XML-based language. XML data files include both data and descriptions of the data that they contain. All XML derived languages use a very simple and flexible ASCII title format for their documents. In all cases the documents are hierarchical, i.e. each document has a clear root element, from which navigation can start using the general document structure (common to all RailML documents) as a guide. The RailML document's root element is called <railml>.The subschemes (infrastructure, rolling stock, and timetable),which contain the relevant data, are derived from this root structure. Using this flexible approach, each individual RailML compatible application determines which particular type of data should be used. The data in XML documents are organised and administered by means of elements and attributes.

```
<railml>
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with OpenTrack (http://www.opentrack.ch) -->
<railmlxmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="timetable.xsd">
<timetable version="0.95" scheduleformat="hh:mm:ss" periodformat="s">
<train trainID="RX 100.2" type="planned" source="opentrack">
<timetableentries>
<entry posID="ZU" departure="06:08:00" type="begin">
</entry>
<entry posID="ZWI" departure="06:10:30" type="pass">
</entry>
<entry posID="ZOER" arrival="06:16:00" departure="06:17:00" minStopTime="9"
type="stop">
</entry>
<entry posID="DUE" departure="06:23:00" type="pass">
</entry>
<entry posID="SCW" departure="06:27:00" type="pass">
</entry>
<entry posID="NAE" departure="06:29:00" type="pass">
</entry>
<entry posID="UST" arrival="06:34:30" type="stop">
</entry>
</timetableentries>
</train>
</timetable>
</railml>
```

Figure 5. RailML data file example

A RailML element begins with a start tag ‘<railml>’, and ends with an ending tag ‘</railml>’. An element can have attributes for more detailed description and may also contain additional elements. Figure 3 illustrates an example RailML document, which contains timetable data. Under the element ‘<timetable>’are listed the individual courses in the element ‘<train>’.The most important attribute of a course, the course number (trainID), is required. Other attributes are optional. Every train running (course) contains as many timetable entries as desired ‘<entry>’, which are arranged under the element of ‘<timetableentries>’.The individual timetable entries contain attributes such as station abbreviation (posID), arrival time (arrival), departure time (departure), or minimum station stopping time (minStopTime). The RailML language is very similar to other XML-based languages and therefore relatively easy to understand and use.

RailML is a simple and efficient way to transfer data between computer programmes used to model different aspects of railway operations. Programmes using the RailML language produce export files with the RailML structure. These files can then be used directly by other programmes. The receiving programme parses the incoming file to obtain only the data it needs, which allows many different programmes to use the same data file. RailML provides two ways for data exchange. The first consists of running applications separately and having them produce output files that are then used as inputs for other applications. The second approach, currently under development, consists of transferring data between two applications directly by means of inter-process communication. Figure 4 illustrates the two forms of data transfer process.

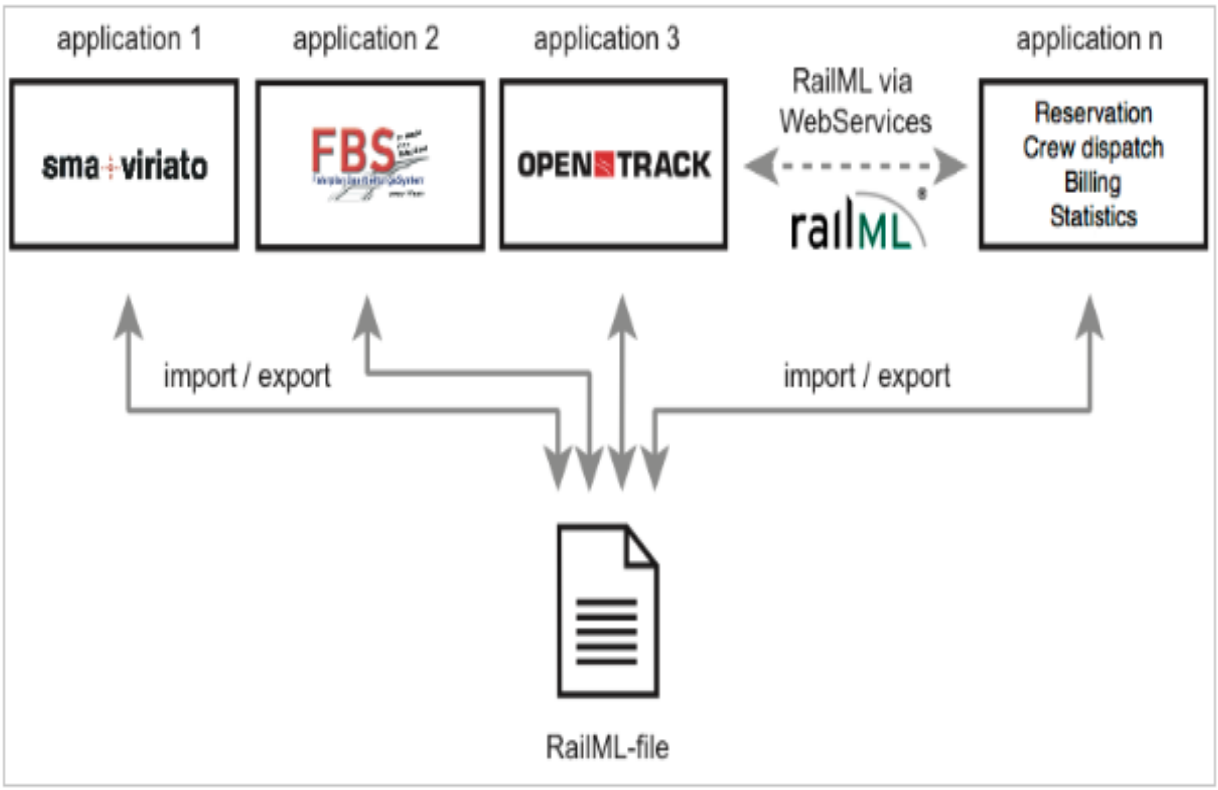

Figure 6. RailML Data Transfer Process

Today, RailML is in daily use although some of the schemes currently not needed/used are not finished yet. This is a main advantage of the flexible RailML format: Schemes can be extended without impact on the data already in use. RailML schemes are in operation for the HSLZuid–the new high speed line from Amsterdam/Den Haag to Bruxelles, for energy calculations in Chinese railway projects, or for time table exchange of private railways in Austria. Also a lot of studies for future railway projects and many research projects are driven by RailML. Last but not least, big railway operators like SBB (Switzerland) demand in their official tenders for new software the availability of a RailML interface.

3.2.2 General Aspects of the RailML Data Model

Table 2: Summary of the RailML Data Model

Criteria	**Assessment**	**Comment**
Applicability to railway applications	High	Data Type was specifically developed to meet railway requirements
Already in industrial use	Yes	OpenTrack, FBS, etc.

Criteria	Assessment	Comment
Technical basis	XML	Data model based on XML, data model, open to extensions for specific needs
Development Group	Very active, High reputation members	Members from rail- and infrastructure companies as well as scientific players
Development Group Size	11	TU Dresden, ETH Zürich, Thales, DLR, etc.
Known User groups	29	DB, ÖBB, SBB, etc.
Known Suppliers	43	DB, Siemens, Thales, Funkwerk, TU Berlin, TU Dresden, TU Wien, etc.
Development Status	Version 2.0	Already reached full-release status. No beta or alpha versions.
License Model	Creative Commons 2.0	Similar to Open Source
Expected costs	No costs	Software is free for download
Documentation	Comprehensive online documentation	Different Documentation styles for different targeted groups (Forums, articles, technical references, wiki)
Project maintenance and progress	Reliable	Next major release (3.0) scheduled for 2014
Modularity	Yes	Organized in substructures
Limitations	Some	Roadbed type, Routes

3.3 DB data model

3.3.1 Description of the DB Data Model

The DB data model is currently being developed by a team of Deutsche Bahn with the participation of manufacturers of interlocking systems. Based on a design created in 2009, the detailed work to complete the data model started early in 2010. The data model shall be completed 2012 as an XML-schema for first prototype applications.

In subsequent steps of the development of the data model, it is planned to model the workflow of planning electronic interlocking and to develop tools for testing and verifying the contents of data files.

Furthermore, the process of storing the data in a database and maintaining the model during use will be discussed.

The model is created in UML using the software tool "Enterprise Architect" by Sparx Systems. The model is publicly available for use as an html-extract from the Enterprise Architect model and as a set of XML-schema file (XSD). A XML-file can be created based on the definitions in the XSD-files as a text file with encoding in UTF-8. The elements and attributes in the XSD-file are currently in German.

The data model will be used to ensure a consistent basis for the transfer and storage of planning data of interlocking systems by the DB AG. The tools for creating the plan have to save and transfer their results in form of a XML-file validated against the given XML-schema.

The planning tools are in principle used as an editor for the XML-file and to represent the data to the user.

The amount of filling of the XML-file may differ in depth depending on the particular planning phase. This data can be completed in any further processing of the planning. For each planning phase (in the context of the Deutsche Bahn), the required data are defined within the XSD-file.

After each major processing step of the planning, the planning results are stored in a database. A transfer of a plan for further processing can take place only via this database; the database thus controls the planning workflow. The data model determines the formal language in which the data are exchanged.

The created planning data are used to simulate the interlocking functions to check if the operational needs are fulfilled. The data will be given to the signalling industry for the construction of their systems. As a result of the implementation of the interlocking system, a correction of the data in the database may be required. In the database, the result of the real implementation of the interlocking will be saved for future use.

Within the data model, the names/Ids of the contributing persons (e.g. planner, validator) who have been working on the data are stored. Each element (object) is secured with a checksum to detect changes in the process.

The model contains a way to represent the logical connection between different elements. These are either defined by references or can be received from the topology. The properties of the elements are supplemented by attributes. This modelling principle opens the model for further developments.

The base of the data model is nodes and edges of the track topology. Almost all other elements are related to the topology either as points on the topology or as regions/lines on the topology. Some additional objects such as logical objects without direct connection to the topology complete the model. The main elements of the model are summarized in table 3.

3.3.2 General Aspects of the DB Data Model

Table 3: Main Elements of the DB Data Model

Group of Elements	Name of Elements (in German)	Description of Elements
elements for the	Basis_Objekt	Data for unified management of the objects

Group of Elements	Name of Elements (in German)	Description of Elements
organization	Punkt_Objekt	positioning of an object to a point on a topological edge
	Bereich_Objekt	line-like object along a topological edge (this can be some edges and part of edges)
	Topographische Objekte	image of the real track layout in the form of nodes (3-dimensional point in the terrain) and edges (length and geometric shape of a track piece)
	Topologische Objekte	abstract representation of the topographic features as topology in a graphical node-edge model
physical elements	Gleis	tracks
	Weichen	points
	Gleissperren	derailers
	Signale	signals
	Einwirkungsstellen	action points of train
	Bauwerke	buildings
logical elements	Fahrstraßen	routes
	Flankenschutz	Flank protection
	Block	blocking
	Schlüsselabhängigkeit	key dependency
	Nahbedienung	shunting areas
	Zugdisposition	Disposition
	Bedienoberfläche	User interface

Table 4: Summary of the DB Data Model

Criteria	Assessment	Comment
Applicability to railway applications	High	Data Type was specifically developed to meet railway requirements
Already in industrial use	No	Under development, release of first version (prototype) planned for 2011
Technical basis	UML	Data model is defined as a UML model created by the software Enterprise Architect
Development Group	Very active, High reputation members	Members from DB and interlocking manufacturers, TU Dresden
Known User groups	1	DB
Known Suppliers		Funkwerk IT, IVV GmbH, TU Dresden
Development Status	Draft	Under development
License Model		
Expected costs		
Documentation	Will be available	Documentation will be available in German
Project maintenance and progress	Reliable	First release planned for 2011, development will be done by Project team
Modularity	Yes	Organized in subschemata
Limitations	Some	Specially developed for the German railway system

3.4 STAMP

3.4.1 Description of the STAMP Data Model

The major component of the broader asset information strategy for the signalling discipline at Network Rail is provided by the Signalling Tools and Methods Programme (STAMP), which is a programme of work to improve the efficiency of the delivery of signalling data. The main target of this work is to align 'tools' to the role of individual users, to automate tasks where practical and to reduce the need to exchange asset data between individuals, processes and systems.

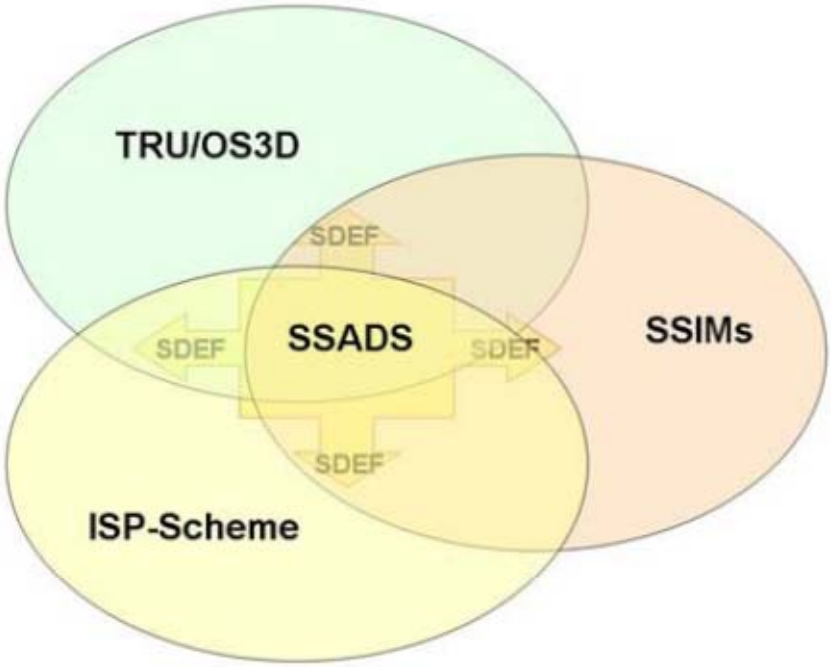

Figure 7. STAMP work streams

As the STAMP development is growing five principal workstreams have been created as the basis to the management of the Signalling asset data within Network Rail

- The use of the Track Recording Unit (TRU) and the development of high definition video kiosks which are surveying the 'as-is' railway and providing asset data to engineers for use within their design tools.
- The deployment of ISP-Scheme. The design tool that takes asset survey data and turns it into the base Scheme Plan.
- The visualization of the designs within Signalling Schemes Image Models (SSIMs), a tool which provides the engineer and stakeholders with a 3D view of the 'to-be' railway before construction even starts.
- The store of the 'to-be' signalling asset data within the Signalling Schemes Asset Data Store (SSADS). A database of the evolving re-signalling schemes from asset condition assessments to eventual handback as the new 'as-is' railway.
- The Signalling Data Exchange Format (SDEF) that allows the efficient movement of signalling asset data between these principal workstreams and also to analysis tools such as those for headway and signal overrun assessments.

Automatic Mapping of the Railway

This will see signalling data in the form of high definition video, route and track centre lines, asset profiling and height data captured for use in the development of signalling schemes, including the building of signal sighting, constructability models, driver simulator models and driver route learning.

With the signalling upgrade of the TRU complete, a process for projects to request the use of the TRU for signalling asset surveys has been trialled on modular signalling and ERTMS projects. With the completion of these trials the use of the TRU for signalling surveys moves away from the development programme into a 'business as usual' environment that will see resignalling projects working directly with asset management.

These trials have also seen the development of a stand-alone high definition video version of Omnicom Engineering Limited's OS3D software. After a period of successful use by signalling designers within Network Rail it is anticipated that an integrated, networked solution will be required to deliver high definition images and OS3D functionality in the future.

Intelligent Scheme Plans (ISP)

Both ISP-Scheme and the second release of ISP-Sketch are now in use in Network Rail and are available for use by signalling designers external to Network Rail. These design tools, in particular ISP-Scheme which makes use of the signalling asset survey data from the TRU, will see the effective and robust incorporation of 'as-is' signalling asset data into resignalling designs.

Phase 2 of ISP-Scheme is currently the subject of an invitation to tender for its development that will deliver a modern design tool for the production of Signalling Scheme Plans and subsequent child documents such as Aspect Sequence Charts, Location Area Plans, Stageworks Plans, etc.

ISP-Sketch is entering the development phase for it's third release in response to change requests from users after use on projects such as CrossRail and Modular Signalling. ISP-Sketch is using a process of Rapid Application Development (RAD) that sees requested functionality delivered to users at a managed but faster rate. ISP-Sketch has also been chosen as the 'engine' to drive the emerging Headway Analysis Tool (HAT) and the Signal Overrun Risk Assessment Tool (SORAT).

Signalling Schemes Image Models (SSIM)

Following on from the successful use of virtual reality and high definition video in models and simulations for the sighting of signals, programme modeling, Network Rail is currently working with suppliers to create an environment where the use of visual 'engineering models' of the 'to-be' railway becomes the norm for resignalling projects. The use of these models and simulations can include:

- Risk Based Signal Sighting
- Constructability Assessments
- 4D (time) Simulations
- Driver Route Learning
- Lessons Learnt Reviews
- Driver Training & Assessment

Signalling Schemes Asset Data Store (SSADS)

The first phase of SSADS is specifically concerned with improving and managing the asset data used for the signal engineering asset renewal condition assessment and renewal planning processes.

Following on from a series of workshops where future users of SSADS where exposed to 'touch and feel' mock-ups to allow the all important user interface to be matched to their needs, the development programme is now entering the critical stage of data cleansing and migration.

Benefits from Phase 1 of SSADS include:

- Greater control of assessment planning and visibility of results
- More detailed asset data
- Improved planning of signalling renewals workbanks, i.e. at more detailed level
- Enhanced data quality (e.g. assets mapped to interlockings via CMN)
- Improved data warehouse reporting tools
- Reduced number of signalling asset databases

Signalling Data Exchange Format (SDEF)

In November 2006 an XML Schema called the Signalling Data Exchange Format was launched to the industry in order to facilitate signalling data exchange tasks.

The development of a SDEF validation tool has recently been added to the STAMP scope, initially for use in validating the use of SDEF by emerging tools. Originating from a workshop held with industry representatives it is planned to move this tool into production as a SDEF Validator for industry wide use. Version 5 of SDEF itself is due for release via its secure web site SDEF User Portal Access (SUPA).

The Signalling Data Exchange Format (SDEF) as part of the Signalling Tools And Methods Programme (STAMP) allows the description of railway assets and signalling controls in such a manner that the information collected in the early stage of a scheme design, when the potential options are being assessed can be passed to analysis tools and CAD packages without loss of information or the potential for errors during the human input stage of the transfer. SDEF is using XML schemata and will be updated permanently by Network Rail.

The SDEF has been developed in order to allow the transfer of signalling layout data between design and analysis tools. This includes traffic flows and logical data so that analysis tools are able to effectively simulate train movements and signalling behaviour.

The SDEF is maintained by the Signalling and Telecomms Programme Engineering (STPE) team. STPE is responsible for keeping the SDEF up to date with new equipment and any alterations necessary to allow its continued use. Procedures associated with STPE are not detailed as part of this specification.

3.4.2 General Aspects of the STAMP Data Model

Table 5: Summary of the STAMP Data Model

Criteria	Assessment	Comment
Applicability to railway applications	High	Data Type was specifically developed to meet railway requirements

Criteria	Assessment	Comment
Already in industrial use	Yes	STAMP and SDEF is in use since 2006, actual in use is version 6 of the data model.
Technical basis	XML	SDEF is using XML schemata
Development Group	Very active, High reputation members	Dedicated team from NR is developing and maintaining the data model
Known User groups	1	NR
Known Suppliers	-	Invensys
Development Status	Version 6	Actual version was rolled out in 2010
License Model	-	-
Expected costs	-	-
Documentation	Available	A full documentation is available in English
Project maintenance and progress	Reliable	Dedicated NR team is maintaining the data model
Modularity	Yes	Organized in subschemata
Limitations	Some	Specially developed for the British railway system

3.5 POE

3.5.1 Description of the PoE

The Position of Elements (PoE) format defines the topology for a railway network based on a node edge model. It provides all topology information needed for ETCS trackside engineering. PoE is designed as an interface format between a measurement company or railway network operator and Siemens AG. The format as Excel file is easily readable by humans, editable and supports a simple machine import.

Basic nodes of PoE are: tips of point and buffer stops or project borders. Additional km-jumps and isolation joints / axle counter can be defined as nodes. Tracks are the edges of the model. Signals, panels, balise groups, locations of gradients or speed changes, platforms, level crossings, tunnels, bridges and every track conditions are attributes of an edge. It is possible to define various types of signals, panels and joints.

Every edge attribute has got a km-value giving the distance to a fix point. Optional a plan km value can be defined for each attribute.

3.5.2 General Aspects of the PoE Data Model

Table 6: Summary of the PoE Data Model

Criteria	Assessment	Comment
Applicability to railway applications	High	Data Type was specifically developed to meet railway requirements
Already in industrial use	Yes	PoE is in use at Siemens for several projects – see documentation in the next section
Technical basis	XLS format	The data model is created as an Excel-Table
Development Group	Very active, High reputation member	Continuous development within Siemens
Known Usergroups	1	Siemens
Known Suppliers	1	Siemens
Development Status	In use	PoE is in use at Siemens for several projects – see documentation in the next section
License Model	Proprietary data model by Siemens	PoE was developed by Siemens for their inhouse use
Expected costs		
Documentation	Full documentation is available	Documentation fully available in English
Project maintenance and progress	Reliable	PoE is in use since 2007, for further development is the Siemens engineering team responsible
Modularity	Yes	States information management depending of the content
Limitations	Some	Developed for ETCS only

3.6 UNISIG Subset 112

3.6.1 Description of the UNISIG Subset 112

According to the “communication from the commission to the European parliament and the council” on the deployment of the European rail signalling system ERTMS/ETCS from 4 July 2005 (COM (2005) 298 final), UNISIG was asked a proposal for Interoperability (IOP) Testing. This description represents the version 1.0.3 (04.05.2010) of the UNISIG Subset 112.

The UNISIG Subset 112 is part of a series of UNISIG subset, which will specify a homogeneous European model for ETCS interoperability testing. This series of subset will possibly be used in the nearer future for the definition of a related Technical Specification of Interoperability (TSI), which is then the basic standard for all further developments.

The UNISIG Subset 112 defines input data definition, input data structure for the configuration of the infrastructure under test. Further documents are the UNISIG Subsets 110 and 111, which contain specifications for the test environment and the test procedures. Furthermore it defines a way to evaluate test results and a template to report test results.

The purpose of the UNISIG Subset 112 is to define the content of the infrastructure scenarios, not the representation of the exact format, and to simplify understanding and evaluation of test scenarios running in different IOP test environments. The purpose of the test evaluation definition is to prove a common format of the evaluation and the report for easier understanding between different parties.

During the IOP test the track definition must be based on real (target) trackside implementation. A fictive trackside can be used during integration and tuning steps. Such a fictive trackside shall support at least the main functions required for interoperability tests. The format of the trackside data must be common. The purpose is to exchange information between different parties. However, in this version of the UNISIG Subset 112, the common format has not yet been developed; only objects and their attributes that are needed for testing are developed. The track layout shall be based on segment elements but shall be easily converted on track element. The conversion will be made easily by using the geographical position of the connection elements; for the test scenarios not the exact geographical position of the tracks are important, but the attributes of the vehicle movements.

The track layout is split into four levels:

- Topology: include all elements necessary to draw a track representation; geographic information is not included; “track” and “switch” were joined by so called “connectors”
- Track: Physical elements which can be added on the representation of the way. The positioning of all elements (physical or functional) makes reference to a segment and an offset on this segment; an offset across to the track is not specified; only the functions of ETCS components were described.
- Routes: non physical elements already represented by a predefined list. Only the routes are described without any further protection options, such as flank protection.
- Track Properties.

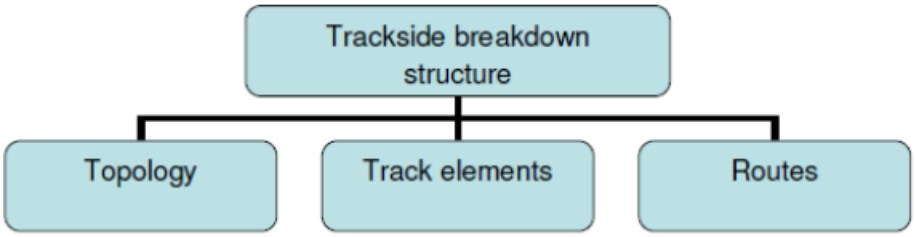

Figure 8. UNISIG Track Layout

The infrastructure modelling data will be extended with the related ETCS project information and stored as XML file. Train data is given for test and simulation in a similar way. The needed data is only listed; no special format for this is given.

The issue of this simulation is, to give evidence of the same behaviour of train and infrastructure by using different supplier for both on-board and trackside equipment. Due to numerous possible combinations testing on real tracks is more or less impossible the use of reference models is needed. The used data model contains the infrastructure functionalities but no sufficient data for an interlocking project.

3.6.2 General Aspects of the Unisig Subset 112 Data Model

Table 7: Summary of the UNISIG Subset 112 Data Model

Criteria	Assessment	Comment
Applicability to railway applications	High	Data Type was specifically developed to meet railway requirements
Already in industrial use	Yes	UNISIG Subset 112 is used by Bombardier for IOP-tests
Technical basis	XML	UNISIG Subset 112 is using XML schemata
Development Group	Very active, High reputation member	Continuous development within the UNISIG group
Known Usergroups	2	UNISIG partners, DB
Known Suppliers	1	UNISIG partners
Development Status	In use in document version 1.0.3 (04-05-2010)	UNISIG Subset 112 is used by Bombardier for IOP-tests
License Model	Proprietary data model by UNISIG	UNISIG Subset 112 was developed and is intended only for UNISIG internal use

Criteria	Assessment	Comment
Expected costs	No costs expected	Data model is open source
Documentation	Full documentation is available	Documentation fully available in English
Project maintenance and progress	Reliable	The subset 112 is maintained by the UNISIG IOP test Working Package
Modularity	Yes	States information management depending of the content
Limitations	Some	Developed for ETCS

Section 4 – DATA MODEL COMPARISON AGAINST A REQUIREMENTS CHECKLIST (WP.C.1)

The goal of the comparison process was to determine, in a non-biased and objective way a ranking of the proposed data models, to give arguments and make the selection of one of the proposed data models transparent. The list of requirements used for the comparison of the data models was assembled from two main sources. The first source was the expertise of workstream C partners being involved in several interlocking design and description processes within their companies. Second main source was the input of workstream D defining and describing a complete set of functional requirements for interlockings of each INESS participating country and extracting the needed objects and attributes. While the first list of requirements is not traceable by definition, linking the derived requirements to the functional requirements of work package D3 was not useful as these extracted objects and attributes were not always as detailed as needed. The main detailed requirements came from outside INESS and are therefore not traceable within INESS.

4.1 List of the Requirements

The main step towards the comparison process of the data models was the creation of a liste of requirements, agreed by all WS C members. Every requirement inside is characterised by a unique ID number. Also a small summary/history for each of the requirements is provided for better understanding. A complete table with all the requirements can be found in the appendix. Some requirements have been split into ´´child requirements´´ which are fulfilling a ´parent´ or basic requirement. The main reason for doing that is because in general some of the basics requirements cannot be easily measured so by dividing them to ´child requirements´, accurate measurability can be achieved.

The general requirements are giving information about the data integrity, free use and authentication of the data model that will be chosen.

The data model structure is described by the requirements which are belonging to the structure group. According to these requirements the data model should be extendable, with clear definitions and with the ability of adding and removing objects. The compatibility with previous versions is necessary as well as the proper documentation in English language. Finally the technical basis should be XML definition.

Very important is the fact that the data model should be compatible with the ERTMS L2 and L1. The European Rail Traffic Management System (ERTMS) is an EU “major European industrial project” to enhance cross-border interoperability and signalling procurement by creating a single Europe-wide standard for railway signalling with the final aim of improving the competitiveness of the rail sector. Main element of the ERTMS is the ETCS (European Train Control System) which includes the control of movement authorities, automatic train protection and the interface to interlockings. Apart from giving a concrete description of the data models the categorization of the requirements provides a better understanding of the strengths and weaknesses of the data models. The five requirements´ categories are listed below:

Table 8: Main Groups of Requirements

Category	**Requirements included** (described by their unique ID number)
STRUCTURE	DM_001 / DM_001-1 / DM_001-2 / DM_001-3 / DM_005 / DM_005-1 / DM_007 / DM_011 / DM_036 / DM_012 / DM_037 / DM_014 / DM_014-1 / DM_025 / DM_030 / DM_035 / DM_035-1
GENERAL_REQ	DM_008 / DM_016 / DM_017 / DM_018 / DM_023 / DM_024 / DM_028 / DM_034
FIELD_ELEMENTS	DM_016-1 / DM_016-1.1 / DM_016-1.2 / DM_016-1.3 / DM_016-1.4 / DM_016-1.5 / DM_016-1.6 / DM_016-1.7 / DM_016-1.8 / DM_016-1.9 / DM_016-1.10 / DM_016-1.11 / DM_016-1.12 / DM_016-1.13 / DM_016-1.14 / DM_016-1.15 / DM_016-1.16 / DM_016-1.17
ERTMS	DM_016-2 / DM_016-2.1 / DM_016-2.2 / DM_016-2.3 / DM_016-2.4 / DM_016-2.5 / DM_016-2.6 / DM_016-2.7 / DM_019
TRACK_LAYOUT	DM_020 / DM_020-1 / DM_026 / DM_026-1 / DM_026-2

The full list of the requirements is given in Appendix A

4.2 Checklist description

The data model comparison against the requirements checklist is also part of the evaluation process. The checklist for data model together with the list of the requirements previously mentioned, are part of the requirements capture excel file.

The checklist for data model includes five requirements´ criteria. The criteria for checking the requirements of data models are:

- **Importance**
  - Importance of the requirement which ranges from 1 to 3
- **Boolean criterion**
  - The requirement is either fulfilled or not
- **Knock out criterion**
  - Critical requirement that has to be fulfilled
- **Fulfilment 0 to 4**
  - Degree of the fulfilment of the requirement which ranges from 0 to 4
- **Description & Comments**

The partners involved in the evaluation of a data model had to learn their features, analyse and decide which of the requirements in the checklist were fulfilled and which their fulfilment degree is.

After that, all the members of each workgroup had to gather their opinions and build an agreed version of fulfilment scores. The fulfilment decisions are accompanied by references to the Data Model documentation and comments when necessary. After some iteration, the final results and documentation references for each data model were prepared.

The complete excel file with the comparison of the data models against the requirement checklist can be found in the Appendix B.

4.3 Formula used for the generation of the data models result

The following formula was agreed to generate an overall data model result/mark.

Data Model Mark **X:** Regardless the effect of knock out requirements

X= $y_k \cdot z_k$ **(1)**

X_{max} = $\sum_1^n y_k \cdot z_{kmax}$ **(2)**

The Data Model Mark, expressed as a percentage:

$$P = \frac{X}{Xmax} \times 100 \quad \textbf{(3)}$$

Where:

y_k= **W**eight (Importance) of the requirement “k”.

Range= [1, 3].

z_k= **F**ulfilment **D**egree of the requirement “k”

in the Data Model

Range = [0, 4]

n = number of requirements

z_{kmax}=maximum fulfilment of the requirement “k”, which is 4 by definition

X_{max} = 408. [The calculation of the **X_{max}** was done by applying the data of the table 9 in the formula (2)].

In the appendix D the overall results, as they have been obtained after the completion of the marking procedure, are given. In order to generate those results the partners have used the formula described above.

Section 5 – RESULTS OF THE DATA MODEL SELECTION (WP.C.1)

5.1 Results

A part of the overall results are shown in the tables that follow. The complete overall results can be found in the appendix D. The following results were generated by using the formula which was described in the previous section.

Table 9: Overall Results

Results for checking	RailML	EurolXL	Stamp	DB Model	PoE Siemens	UNISIG 112
Overall data model mark	**346**	**263**	**349**	**321**	**270**	**290**
Overall data model mark [(X/ X_{max})*100]	**85%**	**64%**	**86%**	**79%**	**66%**	**71%**
Category Results						
STRUCTURE	141	111	158	126	111	119
GENERAL_REQ	28	24	25	28	24	21
FIELD_ELEMENTS	85	94	95	103	57	74
ERTMS	46	2	31	16	38	36
TRACK_LAYOUT	46	32	40	48	40	40
KNOCKOUT REQ. FULFILED	96	93	105	84	87	105

Table 10: Data for Xmax Calculation

Data for calculating X_{max}	Degree of Importance		
	1	**2**	**3**
Number of Requirements	10	16	20

The complete excel file with the overall results can be found in the INESS_WSC_DM_Comparison_Results_v1.3 (Checklist_v1.1) document in the following myndsphere folder:

https://www.myndsphere.com/gm/folder-1.11.123505

5.2 Weaknesses of the Data Models

Apart from the measurable results, the workgroups were also asked to prepare a ´list of gaps´ for each one of the six data models. The weaknesses that the workgroups have discovered, during the selection process for each one of the data model separately, are listed below.

RailML:

- The glossary within the RailML wiki webpage should be completed.
- Lockable devices object should be improved as part of enumeration object for lockable devices.
- The data model should be updated to cover signal aspect and related information within routes and automatic route setting.
- Object “track topology” should be updated to allow the definition of ETCS level 1 or 2 areas.
- For the linking information the parameter Q_LINK is missing.

Euro-Interlocking:

- Documentation only introduction + Excel-file describing the attributes
- No information how to use / fill the data model
- No information about maintenance of the data model
- No example XML-file
- No ERTMS (not in scope of Euro-Interlocking)
- No experience with use of the data model (only used by ÖBB which is not part of WS C)

STAMP

- There is currently no support for level 1 and level 2 ETCS area definition
- The model does not currently support the M_Version attribute
- The start-up delay timer is not currently supported
- Not all lockable devices are supported at present

DB Data Model

- the DB data model is still in development
- the first step of development is using for electronical interlocking without ETCS
- the versioning of XSD-file is planned, not realized
- the DB data model will use ADDONS for special objects and functions
- documentation is not in English, but only in German
- names of objects and attributes are not in English - only in German
- for object "level crossing" only the interface to an interlocking is implemented
- incompatible routes can be found only by calculation

PoE

- no XML file available

- almost no track elements used by interlockings

UNISIG Subset 112

The workgroup did not deliver a list of gaps for this data model.

5.3 Ranking

Apart from the results that have been conducted and after the evaluation of the data models, it was decided that each railway and supplier should judge on how important the different requirements are according to his personal expert opinion. Only those WS C experts with practical experience with the data models have been asked to create a ranking according to their point of view. In addition, reasons for the personal ranking of each partner have be given. The results of the ranking are shown in the table that follows.

Table 11:Ranking

	ADIF	Alstom	Ansaldo	AZD	Bombardier	DB	Invensys	Network Rail	Pro Rail	RFI	Siemens	Thales	Funkwerk	Eliop	*Average*	***Ranking***	*Expected #1*
Stamp (349 Points)	2	2	4	1	2	3	1	1	2	2	3	2	3	2	*2,14*	***2***	*3*
RailML (346 Points)	1	1	1	3	3	2	2	2	1	4	1	4	2	1	*2,00*	***1***	*6*
DB Model (321 Points)	4	3	5	2	1	1	3	3	3	1	2	1	1	3	*2,36*	***3***	*5*
Unisig 112 (290 Points)	3	4	3	4	4	5	4	4	5	5	5	4	4	4	*4,14*	***4***	*0*
PoE Siemens (270 Points)	5	5	6	6	6	6	5	5	6	6	4	6	6	6	*5,57*	***6***	*0*
EurolXL (263 Points)	6	6	2	5	5	4	6	6	4	3	6	5	5	5	*4,86*	***5***	*0*

5.4 Justification for the partners ranking

As it was mentioned before, each one of the partners has given his reasons for justifying the personal ranking.

Reasons given by ADIF

The ranking is based on

- the overall fulfilment of requirements
- the completeness of the data model resulting in less development effort needed for adapting it to fully comply with WS C requirements
- ERTMS requirements fulfilled
- maturity and use of the data model

Reasons given by Alstom

The ranking is based on

- the results of the requirements evaluation that has been performed by the working group
- the results of the ERTMS requirements evaluation that has been performed by the working group
- the industrial use of the Model internationally by several railways
- the maintainability of the model, supported by independent businesses and institutions and existing documentation

Based on this RailML is the preferred data model.

Reasons given by Ansaldo

Maintainability has to be a requirement for the use of the data model in the future. It is important that the data models were "standard" (no proprietary, in English and XML), as the data model should be used and maintained by INESS community and that the data model should cover INESS IXL requirements.

Reasons given by AZD

The base criterion for a structure of the data model is XSD file format. For that reason, we omitted PoE, Euro IXL and UNISIG 112. The remaining models have a structure very similar (based on XSD), so they were considered according to data content.

Stamp reached the highest number of points, has the highest number of points for knock out requirements, contains a large number of objects required for the modelling and description of a schema, and already is in industrial use. However, the documentation is not clear, it is impossible to compare the attributes and elements of each objects, links to websites mentioned in the documentation are not available, and is developed only for the British railway system.

The DB Model received the most number of points in three categories (general req., field elements, track layout), is most compatible with data objects and attributes deduced from WP D3, besides ETCS, and has a large quantity of documentation. However, it does not fulfil two knock out requirements, the documentation and data model are not in English, it is developed only for the German railway system, and the data model is under development.

RailML already is in industrial use, and is supported and developed by independent businesses and institutions from various European countries. However its primary use is not with focus on interlocking, and it has a large quantity of redundant information (timetable, rolling stock)

Reasons given by Bombardier

The data model shall use modern modelling techniques, shall already contain most parts of the complex conventional interlocking data, and shall be useful not only for interfacing a railway to a supplier, but also support a railways process to set up these data. The DB Model has been developed with this in mind together with partners from the industry (Siemens, Thales, Bombardier and others) and contains all the data needed in the complexity that is needed, not only to define the interlocking but to define the cabling, tracks etc., too. That is what is needed at least in German interlocking projects. We see most of this is true for Stamp, too. Stamp is proven in use and contains all the data needed in UK (we missed some complexity of data needed in Germany which would make it not useable until Improvement), DB Model is the most sophisticated one. RailML and PoE lack some important elements (especially wayside). Those are the reasons for sometimes overriding the overall points.

Reasons given by DB

The DB Model fulfils many requirements of the checklist completely. Although the description is in German and it uses German terms, a translation into English will be available if needed and without changing the technical functionality. Currently large parts of the documentation (but not the data model itself) are already translated into English. Additional objects (e.g. ETCS) can be integrated. The model is currently still in development. The versioning is planned but not yet fully implemented.

RailML has originally been developed for the roadmap. It contains the typical objects of the railway. The accuracy of the representation and the properties of the objects is limited. Technical details are not shown in the depth required. The model has evolved over 10 years internationally, and is in use by several railways.

There are the typical applications for Stamp implemented by Network Rail. It has been used 5 years nationally. Logical functions (road, flank protection) are not shown.

Reasons given by Eliop Seinalia

The main criteria followed for the ranking selection have been:

- Good scores in all the categories.
- Fulfilment of all knock-out requirements.

Only RailML and STAMP meet the criteria, but Eliop Seinalia partners consider that RailML has one important additional advantage: RailML is supported by an independent organization, not by any railway company or supplier and such neutrality can ease the introduction of the model in every company. The rest of models have been chosen according to their scores in the comparison.

Reasons given by Invensys

Having considered the ranking provided by the INESS data model analysis, IRG are broadly in agreement with these results, and IRG are therefore happy to adopt the INESS ranking. IRG does not think there is much to choose between RailML and Stamp (SDEF), since both have some significant strengths, and are readily adaptable for future use. However, whilst DB Model also has some significant strengths, IRG would have serious concerns over the lack of support for English with this model.

Reasons given by Network Rail

Stamp has the highest score on the evaluation and there is money and a team available to extend the model in this year's budget to overcome shortcomings. There is a maintenance contract in place with industry and a contract has been set to produce a validator/convertor to allow older versions to be converted. This will be finished by March 2012. Stamp is available to industry and future development is guided by a standing steering committee.

Reasons given by Pro Rail

RailML is preferred because it is being used and continuously developed by an independent organisation. Stamp has a good documentation, though the maintenance of Stamp is not very clear to ProRail. The DB Model seems to be good, but the documentation is not very detailed. EuroIXL is specially made for European interlockings, but unfortunately not maintained, and there is no information on how to use and maintain this data model. Unisig 112 has not much documentation and a low score. PoE is not XML and has a low score.

Reasons given by RFI

From the point of view of RFI is very important that the data model can describe all the necessary objects and functions. DB Model, Stamp and EuroIXL include many of the objects/functions required and have the opportunity to include additional information not initially provided. Nevertheless, the DB Model does not currently have a documentation in English, Stamp is specifically developed for the British railways and EuroIXL does not support the XSD file format.

RailML, Unisig112 and PoE are not able to add information to that already provided in the construction of a route. PoE does not even support the XSD file format.

Reasons given by Siemens

The most important criterion for the decision was the ranking in the categories "ERTMS" and "TRACK_LAYOUT". A second criterion was that the three best models in the overall ranking should also be the three best models in Siemens ranking.

In the evaluation RailML has the highest score in the category "ERTMS", gives good possibilities to build up the track layout (comes in second in that category behind DB Model), it is easy to add new field elements (so the average number of scores in that category are not essential).

The DB Model has the best possibility to build up the Track Layout, but the missing ERTMS information has to be added.

Stamp has the highest ranging in the overall fulfilment and in the category "Structure", but in the categories "ERTMS" and "TRACK_LAYOUT" it is not one of the best.

PoE comes in second in the category "ERTMS" and is already in use by Siemens.

UNISIG 112 has more points in the category "ERTMS" than DB Model or Stamp, but it is not under the first 3 data models in the overall ranking.

EuroIXL is the data model with the lowest number of points in both categories "ERTMS" and "TRACK_LAYOUT".

Reasons given by Thales

Since the DB Model has been developed in cooperation between railway and suppliers, its design fits the needs of all partners involved in an interlocking project. It fulfils a high range of requirements in the categories “GENERAL_REQ”, “FIELD_ELEMENTS” and “TRACK_LAYOUT”. Therefore, an English clone of this model may be used with only small adaptations and the partners may have a benefit out of the model in short time. Moreover, many aspects of the project lifecycle (versioning, handling of releases, putting data from supplier to the railway) are already implemented, which is an advantage to Stamp.

EuroIXL is a little bit old-fashioned (DTD instead of XSL) but has been developed for the same purpose as the INESS-Data-Model. It is a UIC-Model, so the ownership and the maintenance are clear.

RailML still doesn’t have an interlocking scheme. Important elements like routes are missing. So there will be lots of work to do to gain some benefit from this model.

Reasons given by Funkwerk

The DB Model has the best separation of topography, topology, and functional aspects. Thus, it is the most flexible and adaptable model. But there are two main disadvantages of this model. On one hand, the German language in the naming of objects/attributes and in the documentation (which can be solved with comparably small effort, maybe even giving the chance to set up a multilingual dictionary for all objects and attributes while translating the data model and thus clearly defining all terms within the INESS context). On the other hand, ERTMS is currently missing in the DB Model. The model is considered flexible enough to add these data and the DB is currently working on this topic. Furthermore, the DB is currently actively developing the data model and is discussing it with several suppliers (which are also part of INESS).

The reason to prefer RailML to Stamp is the mainly because of the greater community active in RailML, but the two models are almost on the same level.

Section 6 - VALIDATION OF THE SELECTED DATA MODEL (WP.C.2)

One result of the WP C.1 was a list of objects and attributes needed in the EUDRI data model in order to fulfil the requirements from WP D.3. In order to put the chosen data model onto trial, in a first step all partners in the WS C were asked to check this list against real interlockings and real environments. The partners were asked to analyse this list of objects and their attributes in detail by engineers to see what is missing in order to build an interlocking.

From the partners in WS C the analysis was done by Invensys, Bombardier, Ansaldo, AZD, Siemens, Deutsche Bahn, and ProRail. As a result, the list of requirements was completed from the point of view of these partners adding objects and attributes that have been missing and deleting information that are not needed by the partner. The results were gathered by the WP leader Funkwerk IT and integrated into one list. In this step, the analysis of Funkwerk IT was integrated into the list.

Some of the answers were duplicated and it was not always clear what was intended by the attributes. Thus, it was necessary to consolidate the list of objects and attributes among all partners on two WS meetings. Following general principles have been applied in this consolidation process

- objects or attributes shall be deleted if no partner justifies the need for it,
- objects or attributes are added to the list if the partners agreed on the necessity of it, and
- the list concentrates on functional aspects of the interlocking, constructional details are not included

According to these principles no objects or attributes from the original list have been deleted, but several objects and attributes have been added to the list. Most of these come from a more detailed analysis of the requirements from an engineering point of view. But from the discussion of the added attributes it was clear that they arouse from an additional functionality of real interlockings. This means that there is a requirement that has not been identified by WP D.3, a feedback to this WP has been initiated.

The consolidated list of objects and their attributes, that have to be covered by the EUDRI data model, can be found in appendix E.

The second step in the trial of the chosen data model RailML was to check if the objects including their attributes from the list described above are included in RailML. This comparison has been done by the experts for the RailML data model from WP C.1, the DLR, together with the WP leader Funkwerk IT. The result is given in appendix E. If the information is included in RailML it is in detail given where it can be found. Objects or attributes that are up to now not covered in RailML can also be seen in the result. This list of gaps defines what has to be added to the RailML data model in order to cover the needs for a EUDRI data model.

As a result of the selection process, RailML model has been selected as the most appropriate data model. However, there are still some gaps that have to be described and filled by extending the selected data model. In order to achieve greater applicability of this data model a list of requirements should be added. The main groups of the requirements are the following:

- Route
- Powered moveable element
- Lockable and detection device
- Signal
- Local shunting area
- Level crossings
- Train detection section
- Aspect
- Auto device
- Automatic warning system
- Direction
- Interlocking
- Lamp

➢ Engineering configuration requirements

- System configuration
- Route configuration
- Local shunting area configuration
- Powered point configuration
- Signal configuration
- TVP section configuration
- Level crossings configuration
- Catenary group line block
- Power supply
- Group of points addresses
- Main central
- ARS
- Internal block
- Track segment

➢ General configuration

The groups contain different objects and attributes that the data model should fulfil. The complete list is presented in the Appendix E.

Section 7 - SAFETY

It must be mentioned that the data model is not safety relevant. Although it must include all necessary data for defining a safe interlocking, as each user will only use parts of the data model even the completeness of the generic data model is not relevant to safety. The safety of the data used must be provided by the surrounding processes and tools. This safety includes:

- Completeness: All data relevant to all safety relevant functions of an interlocking must be included.
- Integrity: Redundant information must be free of contradictions.
- Reliability: It must be assured that the data is not altered after creation and assessment.

There is an ongoing discussion on if a digital signature can ensure that data is not changed within the life cycle of an interlocking. Therefore, safety was not in the focus of workstream C. Just as with data written on paper, the safety of the data must be provided by the surrounding processes.

Section 8 - CONCLUSIONS

The completion of the evaluation process led to the selection of the RailML data model as the most suitable one. During the selection process some really important general characteristics, that the data model should contain, were discovered. These features are:

- Maturity and previous industrial use of the data model
- Completeness of the data model concerning its fulfilment towards the WS C requirements
- Industrial use internationally
- Documentation in English
- XML format
- No proprietary software

RailML is a simple and efficient way to transfer data between computer programmes used to model different aspects of railway operations. It has high railway applicability and it is already in industrial use. In addition, RailML has one important additional advantage: it is supported by an independent organization, not by any railway company or supplier and such neutrality can ease the introduction of the model in every company.

One important aspect concerning the future use of the data model is the maintenance. As mentioned before, it is really important that the data model, except of being "standard" (no proprietary, in English and XML), also has to be used and maintained by independent organizations and institutes. RailML has the advantage of being already in industrial use, supported and developed by independent businesses and institutions from various European countries. Main aspects for further technical development of the RailML data model have been described in Section 6. Further development of the RailML data model in the listed requirements would positively impact exploitability of the model.

For the

Section 8 – BIBLIOGRAPHY

Ref.	Date	Version	Description
[1]	01/06/2011	1.0	INESS_WSC_Deliverable_D.C.1.1_DataModelDescription
[2]	23/05/2011	1.1	INESS_WSC_Data Model Checklist
[3]	07/07/2011	1.3	INESS_WSC_DM_Comparison_Results_v1.3(Checklist_v1.1)
[4]	07/07/2011	0.3	INESS_WSC_Deliverable_DataModelsEvaluation
[5]	08/08/2011	0.2	INESS _WSC_ Data Model Ranking
[6]	23/07/2010	6	STAMP-3093 Signalling Data Exchange Format Version 6, Network Rail
[7]	2008		European Standards for Railways Interlocking Systems (UIC) http://www.uic.org/download.php/publication/505_05E.pdf
[8]	26/10/2005		RailML – a standard interface for railway data Vasco Paul Kolmorgen, Coordinator of the RailML group, Dresden, Germany; Dr. Daniel Huerlimann, Coordinator of the RailML group, Zurich, http://www.railml.org/
[9]	16/06/2011	2.1	RailML – a standard interface for railway data Vasco Paul Kolmorgen, Coordinator of the RailML group, Dresden, Germany; Dr. Daniel Huerlimann, Coordinator of the RailML group, Zurich, http://www.railml.org/

Section 9 - LIST OF FIGURES

Section 10 - LIST OF TABLES

Section 11 – APPENDICES

Appendix A: List of requirement

Req. Unique ID	Req. Description
DM_001	The Data Model (DM) shall be extendable with easy evolution for introducing (removing) new objects to the model and new attributes to existing objects
DM_001-1	The DM structure shall allow the definition/removal of objects and their attributes.
DM_001-2	The DM structure shall allow the definition/removal of new/existing objects without impacting the existing (already defined) objects.
DM_001-3	The DM structure shall allow the definition/removal of new/existing attributes within an existing object without impacting the existing (already defined) attributes for that object
DM_005	The DM shall ensure the retro-compatibility with previous (old) versions
DM_005-1	The DM shall include a dedicated object for version management for the complete DM
DM_007	The version management shall handle with one version for the data structure and one version for the data itself
DM_008	The DM shall be free and available within the INESS partners This topic will be discussed later on (in a second step) for new comers in the market.
DM_011	The DM shall be documented.
DM_036	The DM documentation shall be available in English
DM_012	The objects, attributes, etc of the DM shall be named in English.
DM_037	The DM documentation shall include a glossary for object and attributes naming
DM_014	The DM shall be based on the XML format file
DM_014-1	The DM shall be described with an XML scheme definition XSD
DM_016	The DM shall be compatible with data objects and attributes deduced from WP D3, including ETCS level 2 and level 1 features
DM_016-1	The DM shall be compatible with data objects and attributes deduced from WP D3
DM_016-1.1	The DM shall include the object **'Point'**
DM_016-1.2	The DM shall include the object **'Derailer'**
DM_016-1.3	The DM shall include the object **'Movable switch diamond crossing'**
DM_016-1.4	The DM shall include the object **'Detectors'**
DM_016-1.5	The DM shall include the object **'Lockable devices'**
DM_016-1.6	The DM shall include the object **'Signal'**
DM_016-1.7	The DM shall include the object **'Local Shunting Area'**
DM_016-1.8	The DM shall include the object **'Level crossing'**
DM_016-1.9	The DM shall include the object **'TVP section'**
DM_016-1.10	The DM shall include at least the minimum information to define routes (mandatory information).
DM_016-1.11	If additional information is available, the DM shall be able to cover this additional information to define routes (e.g. Flank protections, incompatible routes).
DM_016-1.12	The DM shall include signal aspect sequence for each route.

Req. Unique ID	Req. Description
DM_016-1.13	The DM shall include a 'start up' delay timer
DM_016-1.14	The DM shall allow the definition of Automatic Route Setting area
DM_016-1.15	The DM shall allow the definition of L1 area
DM_016-1.16	The DM shall allow the definition of L2 area
DM_016-1.17	The DM shall include configuration data for the objects listed in DM_016-1.1 to 1.9
DM_016-2	The DM **should** be compatible with data objects and attributes deduced from ERTMS L2 and L1 according to UNISIG SUBSET 026
DM_016-2.1	The DM shall include the objects **'Balise'** and/or **'Balise group' and/or 'Loop'**
DM_016-2.2	The DM shall include **'Direction'** information with respect to the track plan, at least for the objects requiring this information (objects and/or data used by ERTMS process)
DM_016-2.3	The DM shall be able to to handle location / distance information to a resolution of 10 cm
DM_016-2.4	The DM shall include the '**M_VERSION**' information as a global parameter
DM_016-2.5	The DM shall include **'Linking'** information
DM_016-2.6	The DM shall include **'Gradient Profile'** information
DM_016-2.7	The DM shall include **'International Static Speed Profile'** information
DM_017	The safety level of the DM shall be defined by the process/architecture surrounding the DM and not by the DM itself
DM_018	The integrity of data exchanged between Railways and Supplier(s) shall be ensured by the process surrounding the DM and not by the DM itself
DM_019	The DM shall be compatible with the architecture defined in WP E3
DM_020	The DM shall be able to deal with geographical data for the objects
DM_020-1	The position reference shall be declared inside the DM.
DM_023	The DM shall include free text labelling inside the objects
DM_024	The integrity of data of the DM shall be ensured by the process surrounding the DM and not by the DM itself
DM_025	The DM shall be readable by human and machines.
DM_026	The DM shall include the means for track layout representation
DM_026-1	The DM shall allow the identification of a track element that follows a previous one.
DM_026-2	The DM shall allow the calculation of the travelled distance between two geographical elements along a continous set of routes
DM_028	Object instance within the DM shall be unique.
DM_030	The DM shall support the definition of valid ranges and valid enumerations for attributes of objects.
DM_034	The DM should support authentification of model data by the process/architecture surrounding the DM
DM_035	The DM shall use universal units for attributes compatible with the INESS Railways community.
DM_035-1	The units used shall be declared inside the DM.

Appendix B: Checklist

Req. Unique ID	Req. Description	Comments / History	Basic req.	Type	Child req. from	Top Mark	Measur-able	Bool-ean req.	Knock-out req.	Impor-tance from 1 to 3
DM_001	The Data Model (DM) shall be extendable with easy evolution for introducing (removing) new objects to the model and new attributes to existing objects	From the user's point of view, a tool should be defined in order to make the use of the DM easy for the user >> TOOL DEVELOPMENT. This point should be kept for the moment, but not as a requirement.	X	STRUCTURE		0				
DM_001-1	The DM structure shall allow the definition/removal of objects and their attributes.			STRUCTURE	DM_001	12	X		X	3
DM_001-2	The DM structure shall allow the definition/removal of new/existing objects without impacting the existing (already defined) objects.			STRUCTURE	DM_001	12	X		X	3
DM_001-3	The DM structure shall allow the definition/removal of new/existing attributes within an existing object without impacting the existing (already defined) attributes for that object			STRUCTURE	DM_001	12	X		X	3
DM_005	The DM shall ensure the retro-compatibility with previous (old) versions	The retro-compatibility shall also be part of the TOOL requirement.	X	STRUCTURE		0				
DM_005-1	The DM shall include a dedicated object for version management for the complete DM			STRUCTURE	DM_005	8	X	X		2

Req. Unique ID	Req. Description	Comments / History	Basic req.	Type	Child req. from	Top Mark	Measur-able	Bool-ean req.	Knock-out req.	Impor-tance from 1 to 3
DM_007	The version management shall handle with one version for the data structure and one version for the data itself		X	STRUCTURE		8	X			2
DM_008	The DM shall be free and available within the INESS partners This topic will be discussed later on (in a second step) for new comers in the market.	Includes Tom's comment (TS 27.4.11: "Open Source" is wrong and shall be replaced by a well known and correct wording)	X	GENERAL_-REQ		12	X	X	X	3
DM_011	The DM shall be documented.		X	STRUCTURE		12	X		X	3
DM_036	The DM documentation shall be available in English	Includes Tom's comment (TS 27.4.11: "shall be available in English")	X	STRUCTURE		12	X	X	X	3
DM_012	The objects, attributes, etc of the DM shall be named in English.		X	STRUCTURE		12	X	X	X	3
DM_037	The DM documentation shall include a glossary for object and attributes naming			STRUCTURE		12	X			3
DM_014	The DM shall be based on the XML format file	State of the art of exchange of files.	X	STRUCTURE		12	X	X	X	3
DM_014-1	The DM shall be described with an XML scheme definition XSD	The DM shall be constructed as a class model (preferably in UML) from which the XML definitions (XSD) are derived.		STRUCTURE	DM_014	12	X	X		3
DM_016	The DM shall be compatible with data objects and attributes deduced from WP D3, including ETCS level 2 and level 1 features		X	GENERAL_-REQ		0				

Req. Unique ID	Req. Description	Comments / History	Basic req.	Type	Child req. from	Top Mark	Measur-able	Bool-ean req.	Knock-out req.	Impor-tance from 1 to 3
DM_016-1	The DM shall be compatible with data objects and attributes deduced from WP D3	Action: List D3 ouput objects (PIC = Eduardo) >> Refer to sheet 'input from WP D3'	X	FIELD_-ELEMENTS	DM_016	0				
DM_016-1.1	The DM shall include the object **'Point'**			FIELD_-ELEMENTS	DM_016-1	12	X			3
DM_016-1.2	The DM shall include the object **'Derailer'**			FIELD_-ELEMENTS	DM_016-1	4	X			1
DM_016-1.3	The DM shall include the object **'Movable switch diamond crossing'**	Including diamond crossings and switch diamond crossing, single and double slip switches		FIELD_-ELEMENTS	DM_016-1	12	X			3
DM_016-1.4	The DM shall include the object **'Detectors'**	See list of detectors in sheet 'Input from WP D3'		FIELD_-ELEMENTS	DM_016-1	4	X			1
DM_016-1.5	The DM shall include the object **'Lockable devices'**	See list of lockable devices in sheet 'Input from WP D3'		FIELD_-ELEMENTS	DM_016-1	4	X			1
DM_016-1.6	The DM shall include the object **'Signal'**			FIELD_-ELEMENTS	DM_016-1	12	X			3
DM_016-1.7	The DM shall include the object **'Local Shunting Area'**			FIELD_-ELEMENTS	DM_016-1	8	X			2
DM_016-1.8	The DM shall include the object **'Level crossing'**			FIELD_-ELEMENTS	DM_016-1	12	X			3
DM_016-1.9	The DM shall include the object **'TVP section'**	TVP = Track Vacancy Proven = Track train detection system		FIELD_-ELEMENTS	DM_016-1	12	X			3
DM_016-1.10	The DM shall include at least the minimum information to define routes (mandatory information).	Usefull for route setting/revoking configuration		FIELD_-ELEMENTS	DM_016-1	12	X			3
DM_016-1.11	If additional information is available, the DM shall be able to cover this additional information to define routes (e.g. Flank protections, incompatible routes).	Usefull for route setting/revoking configuration		FIELD_-ELEMENTS	DM_016-1	8	X			2

Req. Unique ID	Req. Description	Comments / History	Basic req.	Type	Child req. from	Top Mark	Measur-able	Bool-ean req.	Knock-out req.	Impor-tance from 1 to 3
DM_016-1.12	The DM shall include signal aspect sequence for each route.	Usefull for IXL logic		FIELD_-ELEMENTS	DM_016-1	8	X			2
DM_016-1.13	The DM shall include a 'start up' delay timer	Usefull in degraded situation after restart of the IXL		FIELD_-ELEMENTS	DM_016-1	4	X	X		1
DM_016-1.14	The DM shall allow the definition of Automatic Route Setting area	An ARS area shall be configured to define an area where routes set by the automatic route setting system will be rejected in degraded situations		FIELD_-ELEMENTS	DM_016-1	4	X			1
DM_016-1.15	The DM shall allow the definition of L1 area	A L1 area shall be configured to define an area for trains operating in L1.		FIELD_-ELEMENTS	DM_016-1	4	X			1
DM_016-1.16	The DM shall allow the definition of L2 area	A L2 area shall be configured to define an area for trains operating in L2		FIELD_-ELEMENTS	DM_016-1	4	X			1
DM_016-1.17	The DM shall include configuration data for the objects listed in DM_016-1.1 to 1.9	Included as attributes in DM_016-1.1 to 1.9		FIELD_-ELEMENTS	DM_016-1	0				
DM_016-2	The DM **should** be compatible with data objects and attributes deduced from ERTMS L2 and L1 according to UNISIG SUBSET 026	Refer to Excel sheet 'ERTMS parameters'	X	ERTMS	DM_016	0				
DM_016-2.1	The DM shall include the objects **'Balise'** and/or **'Balise group' and/or 'Loop'**	As per packet #5 of UNISIG, see 'ERTMS parameters'		ERTMS	DM_016-2	8	X			2

Req. Unique ID	Req. Description	Comments / History	Basic req.	Type	Child req. from	Top Mark	Measur-able	Bool-ean req.	Knock-out req.	Impor-tance from 1 to 3
DM_016-2.2	The DM shall include **'Direction'** information with respect to the track plan, at least for the objects requiring this information (objects and/or data used by ERTMS process)	This 'direction' information (e.g. left-to-right / right-to-left) will be necessayr for ERTMS. It is also extendable to balise groups.		ERTMS	DM_016-2	4	X			1
DM_016-2.3	The DM shall be able to to handle location / distance information to a resolution of 10 cm			ERTMS	DM_016-2	8	X	X		2
DM_016-2.4	The DM shall include the '**M_VERSION**' information as a global parameter	This will be useful when baseline 3 of UNISIG/ERA will be used.		ERTMS	DM_016-2	4	X	X		1
DM_016-2.5	The DM shall include **'Linking'** information	As per packet #5 of UNISIG, see 'ERTMS parameters'		ERTMS	DM_016-2	8	X			2
DM_016-2.6	The DM shall include **'Gradient Profile'** information	As per packet #21of UNISIG, see 'ERTMS parameters'		ERTMS	DM_016-2	8	X			2
DM_016-2.7	The DM shall include **'International Static Speed Profile'** information	As per packet #27of UNISIG, see 'ERTMS parameters'		ERTMS	DM_016-2	8	X			2
DM_017	The safety level of the DM shall be defined by the process/architecture surrounding the DM and not by the DM itself	**T**he DM will not have to match any SAFETY level by itself	X	GENERAL_-REQ		0				
DM_018	The integrity of data exchanged between Railways and Supplier(s) shall be ensured by the process surrounding the DM and not by the DM itself	Includes Tom's comment (TS 27.4.11: The difference to DM_018 shall be expressed more offensive)	X	GENERAL_-REQ		0				

Req. Unique ID	Req. Description	Comments / History	Basic req.	Type	Child req. from	Top Mark	Measur-able	Bool-ean req.	Knock-out req.	Impor-tance from 1 to 3
DM_019	The DM shall be compatible with the architecture defined in WP E3	The coverage of D3 is already traced by DM_016. Includes Tom's comment (TS 27.4.11: These are two requirements. As WP D3 might extend its focus the last requirements might be less important than the first) This requirement is redundant with DM_016 and s	X	ERTMS		0				
DM_020	The DM shall be able to deal with geographical data for the objects	Geographical data will not be used by the IXL itself but may be usefull for other subsystems AND for validation of installation (see req. DM_21)	X	TRACK_-LAYOUT		8	X			2
DM_020-1	The position reference shall be declared inside the DM.	Special case		TRACK_-LAYOUT	DM_020	8	X	X		2
DM_023	The DM shall include free text labelling inside the objects	For example a dedicated attribute of the object. For information use only.	X	GENERAL_-REQ		4	X			1
DM_024	The integrity of data of the DM shall be ensured by the process surrounding the DM and not by the DM itself		X	GENERAL_-REQ		0				
DM_025	The DM shall be readable by human and machines.	For machines, see DM_014. Probably difficult to measure for human beings	X	STRUCTURE		12	X			3

Req. Unique ID	Req. Description	Comments / History	Basic req.	Type	Child req. from	Top Mark	Measur-able	Bool-ean req.	Knock-out req.	Impor-tance from 1 to 3
DM_026	The DM shall include the means for track layout representation	As long as this requirement will be met by all the existing data models, a child requirement may be added here in order to support the decision during the checklist process.	X	TRACK_-LAYOUT		8	X			2
DM_026-1	The DM shall allow the identification of a track element that follows a previous one.			TRACK_-LAYOUT	DM_026	12	X		X	3
DM_026-2	The DM shall allow the calculation of the travelled distance between two geographical elements along a continuous set of routes	Includes Tom's comment (TS 27.4.11: If the distance is not calculatable, it may be included in the data (which makes it calculatable in another way))		TRACK_-LAYOUT	DM_026	12	X			3
DM_028	Object instance within the DM shall be unique.		X	GENERAL_-REQ		12	X	X		3
DM_030	The DM shall support the definition of valid ranges and valid enumerations for attributes of objects.	It may be helpfull to set boundaries for valid attributes.	X	STRUCTURE		8	X	X		2
DM_034	The DM should support authentification of model data by the process/architecture surrounding the DM	See also DM_006	X	GENERAL_-REQ		0				
DM_035	The DM shall use universal units for attributes compatible with the INESS Railways community.	Complementary to DM_016	X	STRUCTURE		8	X			2
DM_035-1	The units used shall be declared inside the DM.			STRUCTURE	DM_035	8	X	X		2

Appendix C: Results

Req. Unique ID	Req. Description	RailML DM	Euro-Interlocking DM	Stamp NetworkRail	DB Model	PoE Siemens	UNISIG 112
		Fulfilment from 0 to 4	Fulfilment from 0 to 4	Fulfilment from 0 to 4	Fulfilment from 0 to 4	Fulfilment from 0 to 4	Fulfilment from 0 to 4
DM_001	The Data Model (DM) shall be extendable with easy evolution for introducing (removing) new objects to the model and new attributes to existing objects	0	0	4	0	0	0
DM_001-1	The DM structure shall allow the definition/removal of objects and their attributes.	4	3	4	4	4	4
DM_001-2	The DM structure shall allow the definition/removal of new/existing objects without impacting the existing (already defined) objects.	2	3	4	4	4	4
DM_001-3	The DM structure shall allow the definition/removal of new/existing attributes within an existing object without impacting the existing (already defined) attributes for that object	2	3	4	4	1	4
DM_005	The DM shall ensure the retro-compatibility with previous (old) versions	3	0	4	4	4	0
DM_005-1	The DM shall include a dedicated object for version management for the complete DM	4	0	4	0	4	0
DM_007	The version management shall handle with one version for the data structure and one version for the data itself	2	4	4	4	0	0
DM_008	The DM shall be free and available within the INESS partners This topic will be discussed later on (in a second step) for new comers in the market.	4	4	4	4	4	4
DM_011	The DM shall be documented.	4	2	4	4	4	3
DM_036	The DM documentation shall be available in English	4	4	4	0	4	4
DM_012	The objects, attributes, etc of the DM shall be named in English.	4	4	4	0	4	4
DM_037	The DM documentation shall include a glossary for object and attributes naming	3	2	4	4	4	4
DM_014	The DM shall be based on the XML format file	4	4	4	4	0	4

Req. Unique ID	Req. Description	RailML DM	Euro-Interlocking DM	Stamp NetworkRail	DB Model	PoE Siemens	UNISIG 112
		Fulfilment from 0 to 4	Fulfilment from 0 to 4	Fulfilment from 0 to 4	Fulfilment from 0 to 4	Fulfilment from 0 to 4	Fulfilment from 0 to 4
DM_014-1	The DM shall be described with an XML scheme definition XSD	4	0	4	4	0	2
DM_016	The DM shall be compatible with data objects and attributes deduced from WP D3, including ETCS level 2 and level 1 features	0	0	0	0	0	0
DM_016-1	The DM shall be compatible with data objects and attributes deduced from WP D3	0	0	2	0	0	0
DM_016-1.1	The DM shall include the object 'Point'	4	4	4	4	4	4
DM_016-1.2	The DM shall include the object 'Derailer'	0	4	4	4	0	2
DM_016-1.3	The DM shall include the object **'Movable switch diamond crossing**'	4	4	4	4	4	0
DM_016-1.4	The DM shall include the object **'Detectors**'	3	0	2	0	0	0
DM_016-1.5	The DM shall include the object **'Lockable devices**'	1	0	2	4	0	0
DM_016-1.6	The DM shall include the object **'Signal**'	4	4	4	4	4	4
DM_016-1.7	The DM shall include the object **'Local Shunting Area**'	4	4	3	4	0	4
DM_016-1.8	The DM shall include the object **'Level crossing**'	4	4	4	3	4	0
DM_016-1.9	The DM shall include the object **'TVP section**'	3	4	4	4	3	4
DM_016-1.10	The DM shall include at least the minimum information to define routes (mandatory information).	4	4	4	4	0	4
DM_016-1.11	If additional information is available, the DM shall be able to cover this additional information to define routes (e.g. Flank protections, incompatible routes).	0	4	2	3	0	0
DM_016-1.12	The DM shall include signal aspect sequence for each route.	0	1	2	4	0	4
DM_016-1.13	The DM shall include a 'start up' delay timer	4	0	0	0	0	0
DM_016-1.14	The DM shall allow the definition of Automatic Route Setting area	0	0	1	4	0	0
DM_016-1.15	The DM shall allow the definition of L1 area	0	0	0	0	0	4
DM_016-1.16	The DM shall allow the definition of L2 area	0	0	0	0	0	4
DM_016-1.17	The DM shall include configuration data for the objects listed in DM_016-1.1 to 1.9	0	0	0	0	0	0

Req. Unique ID	Req. Description	RailML DM	Euro-Interlocking DM	Stamp NetworkRail	DB Model	PoE Siemens	UNISIG 112
		Fulfilment from 0 to 4	Fulfilment from 0 to 4	Fulfilment from 0 to 4	Fulfilment from 0 to 4	Fulfilment from 0 to 4	Fulfilment from 0 to 4
DM_016-2	The DM **should** be compatible with data objects and attributes deduced from ERTMS L2 and L1 according to UNISIG SUBSET 026	0	0	0	0	0	0
DM_016-2.1	The DM shall include the objects **'Balise'** and/or **'Balise group' and/or 'Loop'**	4	0	4	0	4	4
DM_016-2.2	The DM shall include **'Direction'** information with respect to the track plan, at least for the objects requiring this information (objects and/or data used by ERTMS process)	4	2	3	0	4	4
DM_016-2.3	The DM shall be able to to handle location / distance information to a resolution of 10 cm	4	0	4	4	4	2
DM_016-2.4	The DM shall include the '**M_VERSION**' information as a global parameter	4	0	0	0	0	4
DM_016-2.5	The DM shall include **'Linking'** information	3	0	1	0	2	0
DM_016-2.6	The DM shall include **'Gradient Profile'** information	4	0	3	4	4	4
DM_016-2.7	The DM shall include **'International Static Speed Profile'** information	4	0	2	0	3	4
DM_017	The safety level of the DM shall be defined by the process/architecture surrounding the DM and not by the DM itself	0	0	0	0	0	0
DM_018	The integrity of data exchanged between Railways and Supplier(s) shall be ensured by the process surrounding the DM and not by the DM itself	0	0	0	0	0	0
DM_019	The DM shall be compatible with the architecture defined in WP E3	0	0	0	0	0	0
DM_020	The DM shall be able to deal with geographical data for the objects	4	0	4	4	0	0
DM_020-1	The position reference shall be declared inside the DM.	4	0	4	4	4	4
DM_023	The DM shall include free text labelling inside the objects	4	0	4	4	0	0
DM_024	The integrity of data of the DM shall be ensured by the process surrounding the DM and not by the DM itself	0	0	0	0	0	0
DM_025	The DM shall be readable by human and machines.	4	4	4	4	4	4

Req. Unique ID	Req. Description	RailML DM	Euro-Interlocking DM	Stamp NetworkRail	DB Model	PoE Siemens	UNISIG 112
		Fulfilment from 0 to 4	Fulfilment from 0 to 4	Fulfilment from 0 to 4	Fulfilment from 0 to 4	Fulfilment from 0 to 4	Fulfilment from 0 to 4
DM_026	The DM shall include the means for track layout representation	3	4	3	4	4	4
DM_026-1	The DM shall allow the identification of a track element that follows a previous one.	4	4	3	4	4	4
DM_026-2	The DM shall allow the calculation of the travelled distance between two geographical elements along a continous set of routes	4	4	3	4	4	4
DM_028	Object instance within the DM shall be unique.	4	4	3	4	4	3
DM_030	The DM shall support the definition of valid ranges and valid enumerations for attributes of objects.	4	0	3	4	0	0
DM_034	The DM should support authentification of model data by the process/architecture surrounding the DM	4	0	0	0	0	0
DM_035	The DM shall use universal units for attributes compatible with the INESS Railways community.	4	4	4	3	4	3
DM_035-1	The units used shall be declared inside the DM.	4	4	4	4	4	1

Appendix D: Overall results

Statistics on the requirements	
Overall number of requirements	57
Number of measurable requirements	46
Number of Boolean requirements	13
Number of Knock-out requirements	9
Number of requirements importance 3	20
Number of requirements importance 2	16
Number of requirements importance 1	10
Top Mark	408

STRUCTURE	GENERAL_REQ	FIELD_ELEMENTS	ERTMS	TRACK_LAYOUT	TOTAL
160	28	124	48	48	408
39%	7%	30%	12%	12%	100%

Results for checking	RailML DM	EurolXL	Stamp	DB Model	PoE Siemens	UNISIG 112	
Overall fulfilment	**346**	**263**	**349**	**321**	**270**	**290**	Highest Mark per category
Overall fulfilment Formula 1	**85%**	**64%**	**86%**	**79%**	**66%**	**71%**	
STRUCTURE	141	111	158	126	111	119	160
GENERAL_REQ	28	24	25	28	24	21	28
FIELD_ELEMENTS	85	94	95	103	57	74	124
ERTMS	46	2	31	16	38	36	48
TRACK_LAYOUT	46	32	40	48	40	40	48
KNOCKOUT REQ. FULFILED	96	93	105	84	87	105	108
Number req. fulfilled degree 4	32	21	29	32	25	26	
Number req. fulfilled degree 3	6	3	8	3	2	3	
Number req. fulfilled degree 2	3	3	6	0	1	3	
Number req. fulfilled degree 1	1	1	2	0	1	1	
Number req. not fulfilled	6	18	4	12	18	13	

Results for checking	RailML DM	EurolXL	Stamp	DB Model	PoE Siemens	UNISIG 112
Knock-out requirements						
Fulfilment	**96**	**93**	**105**	**84**	**87**	**105**
Number req. fulfilled degree 4	7	5	8	7	7	8
Number req. fulfilled degree 3	0	3	1	0	0	1
Number req. fulfilled degree 2	2	1	0	0	0	0
Number req. fulfilled degree 1	0	0	0	0	1	0
Number req. not fulfilled	**0**	**0**	**0**	**2**	**1**	**0**
Requirements of importance 3						
Fulfilment	**222**	**207**	**231**	**213**	**192**	**204**
Number req. fulfilled degree 4	16	14	17	17	15	15
Number req. fulfilled degree 3	2	3	3	1	1	2
Number req. fulfilled degree 2	2	2	0	0	0	1
Number req. fulfilled degree 1	0	0	0	0	1	0
Number req. not fulfilled	0	1	0	2	3	2
Requirements of importance 2						
Fulfilment	**104**	**50**	**102**	**92**	**74**	**68**
Number req. fulfilled degree 4	11	6	8	10	8	7
Number req. fulfilled degree 3	2	0	4	2	1	1
Number req. fulfilled degree 2	1	0	3	0	1	1
Number req. fulfilled degree 1	0	1	1	0	0	1
Number req. not fulfilled	2	9	0	4	6	6
Requirements of importance 1						
Fulfilment	**20**	**6**	**16**	**16**	**4**	**18**
Number req. fulfilled degree 4	4	1	2	4	1	4
Number req. fulfilled degree 3	1	0	1	0	0	0
Number req. fulfilled degree 2	0	1	2	0	0	1
Number req. fulfilled degree 1	1	0	1	0	0	0
Number req. not fulfilled	4	8	4	6	9	5

Appendix E: Final list of requirements and coverage by RailML

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
1	**Route**	**Route**					The route is not part of RailML since the subschema interlocking is up to now not included in RailML. Most of the objects referenced by the route are part of RailML (see objects in the list).
			1.1	ID			not covered
			1.2	route name		The route name (including overlap) as string	not covered
			1.3	route type	values defined in ETCS, can be extented for national needs		not covered
			1.4	route signalling type	ETCS L2; L1; conventional	Further attribute to distinguish between different routes for L2, L1 or conventional operation	not covered
			1.5	route body		Parent attribute for other attributes in the list	not covered
			1.6	flank protection for route body		Parent attribute for other attributes in the list	not covered
			1.7	overlap		Parent attribute for other attributes in the list	not covered
			1.8	flank protection for overlap		Parent attribute for other attributes in the list	not covered
			1.9	route entry signal			not covered
			1.10	route exit signal			not covered
			1.11	sub-route signal			not covered
			1.12	signals for flank protection to the route body			not covered

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			1.13	TVP sections in route body		TVP sections must be free by default, exceptions can be put in here	not covered
			1.14	TVP sections in overlap		TVP sections must be free by default, exeptions can be put in here	not covered
			1.15	TVP sections in flank protection		TVP sections must be free by default, exeptions can be put in here	not covered
			1.16	moveable elements in route body			not covered
			1.17	moveable elements in overlap			not covered
			1.18	moveable elements for flank protection	associated elements and their positions		not covered
			1.19	moveable elements in rear of the route entry signal			not covered
			1.20	moveable elements for opposing movement protection			not covered
			1.21	level crossings in route body			not covered
			1.22	level crossings in overlap			not covered
			1.23	lockable devices in route body			not covered
			1.24	lockable devices in overlap			not covered
			1.25	lockable devices for flank protection			not covered
			1.26	detection devices required by the route			not covered
			1.27	approach zone			not covered
			1.28	overlap speed		allowed speed of the overlap	not covered

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			1.29	maximum speed		maximum allowed speed of the route, especially if can not derived from the speed of the elements and overlap	not covered
			1.30	ARS allowed	yes / no	route may be locked / set by an ARS	not covered
			1.31	catenary warning	yes / no	Information whether the route has to be set with a special command (in german "F-Bedienung"). The special command is necessary if you set a route that starts in an area with catenary and advances in a section without catenary.	not covered
			1.32	stopping train / passing train		Necessary for level crossings behind the exit signal	not covered
			1.33	passenger train / freight train train category / suitability		Necessary e.g. for setting more than one route for freight trains and showing special signal aspects on the entry signal.	not covered
			1.34	route allowed by technician	yes / no	Default: no	not covered
			1.35	the opposing movement protection			not covered
			1.36	incompatible routes			not covered
			1.37	signal aspect sequence			not covered
			1.38	shunting route depending on line block permission	yes / no		not covered
			1.39	overlap depending on line block permission	yes / no		not covered
			1.40	information whether the 'main' route exit signal may be 'used' as a 'shunting' route entry signal		This determines whether the main route is followed by a shunt route	not covered

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			1.41	information whether the 'shunting' route entry signal may be 'used' as a 'main' route exit signal		This determines if a shunt route is preceded by a main route or is a standalone shunt signal	not covered
			1.42	signals located at the same TVP boundary as a route exit signal in opposite direction that may be used as route exit signal		Sometimes it is allowed to set opposing routes to those signals.	not covered
			1.43	information about connected route in a neighbouring interlocking		Necessary if a route is divided into two parts in two neighbouring interlockings. (Originally: the information whether the acknowledgement from the neighbouring signaller is required for the route)	not covered
				attributes for route releasing			not covered
			1.44	TVP sections assigned as destination track		This indicates that the TVP is the last one in the route for route releasing and monitoring purposes	not covered
			1.45	TVP section that activates the timed release of a destination track		The timer is only needed if there is locking to be released once the train reaches the destination track	not covered
			1.46	release time for destination track releasing			not covered
			1.47	TVP sections that activate the timed release of an overlap			not covered
			1.48	overlap release delay time			not covered
			1.49	destination track delay time			not covered

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			1.50	release delay time if the cancellation of a route will always require running a delay timer			not covered
			1.51	RBC acknowledgement time		It has to checked if this is a route based or a global parameter	not covered
			1.52	release delay time for the release of route locking following cancellation of a route with an approach zone occupied			not covered
			1.53	residual release delay time for residual route releasing			not covered
			1.54	delay time for staff responsible route releasing		Staff Responsible routes are used in failure conditions when normally locking and monitoring is not possible. Timers are used to improve safety.	not covered
			1.55	TVP sections behind the entry signal that will not set the entry signal to stop		It has to be checked if the method of setting the entry signal to stop has to be defined here by an attribute. (e.g. the requirement that the TVP in front of the signal has to be free, the one behind the signal has to be occupied, used in Switzerland)	not covered
2	**Powered Moveable Element**	**Point**					complex type rail:eSwitch
			2.1	ID			attribute id (type rail:tGenericID)

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			2.2	point name			attribute name (type rail:tGenericName) a second attribute code (type rail:tGenericName) may be used additionally
			2.3	position on track		absolute position of the tip of the point on the track (the travelled distance on the track)	attribute pos (type rail:tLengthM) additionally absPos (type rail:tLengthM) may be used
			2.4	neighbouring elements	elements on a-, b- and c-side	neighbouring elements on a-, b- and c-side a-side is the base of the point	defined by element rail:eSwitch/connection (one for each side of the point)
			2.5	associated TVP			not covered
			2.6	deviated position	left / right	Gives information which of the positions (left or right) is the deviated / diverging position.	attribute trackContinueCourse (type rail:tCourse) further information can be retrieved from the element rail:eSwitch/connection/course
			2.7	preferred position	no / left / right		attribute normalPosition (type rail:tCourse)
			2.8	number of point machines			not covered
			2.9	maximum speed in deviated position		The speed is only needed for the deviated position. The speed on the other leg is modelled through the speed profile of the track.	attribute rail:eSwitch/connection/maxSpeed of the connection element
			2.10	maximum time for point to turn over			not covered
			2.11	is key-locked	yes / no		not covered
			2.12	fouling isolated joints left	yes / no		not covered
			2.13	fouling isolated joints right	yes / no		not covered
			2.14	local control	yes / no		not covered
			2.15	end position sensing unit left leg	yes / no		not covered

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			2.16	end position sensing unit right leg	yes / no		not covered
			2.17	coupled point	corresponding point		not covered
			2.18	type of coupling	physical / logical		not covered
			2.19	arrangement of coupling	no / left-left / left-right		not covered
			2.20	fouling transportation	list of points including the relevant leg	In some cases (points within fouling distance of each other) it is necessary to transport fouling information over one point to another one. A list of points from which fouling information is received including the leg is necessary. New requirement for D3.	not covered
			2.21	information whether the point functions as a selective protection point			not covered
			2.22	default position if the point is a selective protection point			not covered
			2.23	alternative protection elements for each point position if the point is a selective protection point			not covered
3	**Powered Moveable Element**	**Derailer**					not covered
			3.1	ID			not covered
			3.2	derailer name			not covered
			3.3	position on track		absolute position on the track (the travelled distance on the track)	not covered

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			3.4	neighbouring elements	elements on a- and b-side	neighbouring elements on a- and b-side a-side is the starting side of the derailer	not covered
			3.5	associated TVP			not covered
			3.6	preferred position	no / on_rail / off_rail		not covered
			3.7	ejection direction	left / right		not covered
			3.8	maximum moving time			not covered
			3.9	local control	yes / no		not covered
			3.10	coupled point	corresponding point		not covered
			3.11	type of coupling	physical / logical		not covered
			3.12	arrangement of coupling	no / on_rail-left / on_rail-right		not covered
4	**Powered Moveable Element**	**Moveable switch diamond crossing**					complex type rail:eCrossing
			4.1	ID			attribute id (type rail:tGenericID)
			4.2	crossing name			attribute name (type rail:tGenericName) a second attribute code (type rail:tGenericName) may be used additionally
			4.3	position on track		absolute position on the track (the travelled distance on the track)	attribute pos (type rail:tLengthM) additionally absPos (type rail:tLengthM) may be used
			4.4	type of crossing	crossing, diamond crossing, single slip switch, double slip switch		attribute type (type rail:tCrossingType)

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			4.5	neighbouring elements	elements on a-, b-, c- and d-side	neighbouring elements on a-, b-, c- and d-side sides are named a to d counter clockwise, with a/b on one side and c/d on the other side of the crossing is the base of the point	defined by element rail:eCrossing/connection (one for each side of the crossing)
			4.6	associated TVP			not covered
			4.7	number of point machines			not covered
			4.8	maximum speed in direction a-d		The speed is only needed for the deviated positions (see attribute 2.9 of the point).	attribute rail:eCrossing/connection/maxSpeed of the connection element
			4.9	maximum speed in direction b-c		see attribute 4.8	attribute rail:eCrossing/connection/maxSpeed of the connection element
			4.10	maximum moving time			not covered
			4.11	fouling isolated joints leg a			not covered
			4.12	fouling isolated joints leg b			not covered
			4.13	fouling isolated joints leg c			not covered
			4.15	fouling isolated joints leg d			not covered
			4.16	local control	yes / no		not covered
			4.17	end position sensing unit leg a	yes / no		not covered
			4.18	end position sensing unit leg b	yes / no		not covered
			4.19	end position sensing unit leg c	yes / no		not covered
			4.20	end position sensing unit leg d	yes / no		not covered
			4.21	coupled point ab-side	corresponding point		not covered
			4.22	type of coupling ab-side	physical / logical		not covered
			4.23	arrangement of coupling ab-side	no / left-left / left-right		not covered
			4.24	coupled point cd-side	corresponding point		not covered

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			4.25	type of coupling cd-side	physical / logical		not covered
			4.26	arrangement of coupling cd-side	no / left-left / left-right		not covered
			4.27	fouling transportation	list of points including the relevant leg	In special cases (points within fouling distance of each other) it is necessary to transport fouling information over one point to another one. A list of points from which fouling information is received including the leg is necessary. New requirement for D3.	not covered
5	**Lockable and detection device**	**Detector**					not covered
			5.1	ID			not covered
			5.2	detector name			not covered
			5.3	type of detector	derailment embankment slip falling object flat wheel hot box light sensor pantograph carbon layer pedal sand snow or rock slide train profile trip wire wheel load scale wind	List of possible types of detectors	not covered
			5.4	position on track		absolute position on the track (the travelled distance on the track)	not covered

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			5.5	neighbouring elements	elements on a- and b-side	Neighbouring elements on a- and b-side If the detector has a directional function, the a-side is the starting side of the detector.	not covered
			5.6	effected area		Area that the detector as influences on. This can be one track (like the pedal detector) or a whole area (like a sand detector)	not covered
6	**Lockable and detection device**	**Lockable device**					not covered
			6.1	ID			not covered
			6.2	device name			not covered
			6.3	type of lockable device	adjusting ramp awanst border barrier gate gauge changer ground lever frame hand-operated level crossing hand-operated moveable element hand-operated moveable point moveable bridge travelling platform	List of possible types of lockable devices The German "Ausweichanschlussstelle" (short "Awanst") is a key-locked point on an open line. Routes can use the ordinary line block or can head to the unknown zone behind the key-locked point. New requirement for D3.	not covered
			6.4	position on track		absolute position on the track (the travelled distance on the track)	not covered
			6.5	neighbouring elements	elements on all sides	Neighbouring elements on all sides. The number of sides may differ for different devices.	not covered

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			6.6	timer for releasing the hand-operated point		Parameter only valid for the special lockable device hand-operated point	not covered
7	**Signal**	**Signal**					complex type rail:tSignal
			7.1	ID		-	attribute id (type rail:tGenericID)
			7.2	signal name			attribute name (type rail:tGenericName) a second attribute code (type rail:tGenericName) may be used additionally
			7.3	signal type	auto, controlled, distant, shunt, Home, Departure, Block, virtual (may be continued)		attribute type (type rail:tSignalType) The list has to be extended.
			7.4	position on track		absolute position on the track (the travelled distance on the track)	attribute pos (type rail:tLengthM) additionally absPos (type rail:tLengthM) may be used
			7.5	neighbouring elements	elements on a- and b-side	neighbouring elements on a- and b-side a-side is the "visible" side of the signal	have to be calculated / derived from the track and the position on the track The direction of the signal is given in the attribute dir (type rail:tLaxDirection)
			7.6	the type of signal replacement to be used		A signal may be replaced to stop by other than the first wheel passing the signal. This is used when the driver is not at the front of the train e.g. propelling.	not covered
			7.7	the permitted speed restrictions required if speed degradation is active		-	not covered

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			7.8	information whether a signal is associated with a departure indicator (for staff)			not covered
			7.9	information whether this signal is the first signal associated with the departure indicator			not covered
			7.10	information whether this signal is the intermediate signal associated with the departure indicator			not covered
			7.11	information whether this signal is the last signal associated with the departure indicator			not covered
			7.12	information whether a signal is associated with a close doors indicator (from staff to driver)			not covered
			7.13	information whether a signal is associated with a right away indicator (from staff to driver)			not covered
			7.14	information whether a signal is required to display 'proceed' aspects for a 'G' indication		This would normally be the case if the ATP is not continuous, but on some systems, the “G” can mean simply “Close Doors”.	not covered
			7.15	information whether a signal is associated with a route indicator for a configured route			not covered
			7.16	information whether a signal is associated with a proceed with caution indicator for the configured route			not covered

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			7.17	the information whether a shunting signal is used as a modified shunting signal		Shunting signals where the aspect is modified to proceed without the signal being set.	not covered
			7.18	the information whether an approach delay is configured on the signal			not covered
			7.19	the TVP sections required to be 'occupied' for an approach delay			not covered
			7.20	the required time that each TVP section shall be 'occupied' for an approach delay		-	not covered
			7.21	Hand side track location	left , right	This information is sometimes relevant for the Interlocking itself	not covered
			7.22	Signal aspects		Possibility to store signal aspects including parameters if necessary (e.g. the duration for a proceed with caution signal)	not covered
			7.23	signal lamp dimming yes/no			not covered
			7.24	associated dimming control area			not covered
				the associated co-acting signal if the signal is associated		A second signal head is provided where there are difficult sighting conditions, e.g. tunnels or station canopies.	
				type of lamps	bulb, LED, ...	The inclusion of these parameters has to be decided depending on the architecture of the INESS Interlocking (what is in the responsibility of the interlocking). If the information has to be included an object lamp has to be included.	not covered

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
				current of lamps		The inclusion of these parameters has to be decided depending on the architecture of the INESS Interlocking (what is in the responsibility of the interlocking). If the information has to be included an object lamp has to included.	not covered
8	**Local Shunting Area**	**Local Shunting Area**					complex type rail:eLocallyControlledArea can be used as basis
			8.1	ID			attribute id (type rail:tGenericID)
			8.2	LSA name			attribute name (type rail:tGenericName) a second attribute code (type rail:tGenericName) may be used additionally
			8.3	LSA type	e.g. with control box / local shunting		not covered
			8.4	moveable elements in LSA			may be derived from the tracks belonging to the LSA
			8.5	lockable elements in LSA			not covered
			8.6	moveable elements required locked by LSA		including locked position	not covered
			8.7	level crossing required closed by LSA			not covered
			8.8	lockable elements required locked by LSA		including locked position	not covered
			8.9	signals in LSA			may be derived from the tracks belonging to the LSA
			8.10	track sections in LSA			may be derived from the tracks belonging to the LSA
			8.11	border signals of the LSA			not covered
			8.12	shunting routes in LSA			not covered

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			8.13	the timer for local shunting area withdrawal			not covered
9	**Working Area**	**Working Area**					Since the working area is similar to a LSA, it may be modelled similar. Up to now it is not part of RailML.
			9.1	ID			not covered
			9.2	working area name			not covered
			9.3	working area type			not covered
			9.4	moveable elements in working area			not covered
			9.5	lockable elements in working area			not covered
			9.6	moveable elements required locked by working area		including locked position	not covered
			9.7	level crossing required closed by working area			not covered
			9.8	lockable elements required locked by working area		including locked position	not covered
			9.9	signals in working area			not covered
			9.10	TVP sections in working area			not covered
			9.11	border signals of the working area			not covered
			9.12	shunting routes in working area			not covered
10	**Temporary Speed Restriction**	**Temporary Speed Restriction**				If temporary speed restrictions have to be included in the data model because of national rules	not covered
			10.1	ID			not covered
			10.2	name of temporary speed restriction			not covered
			10.3	type of temporary speed restriction			not covered

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			10.4	moveable elements in temporary speed restriction			not covered
			10.5	lockable elements in temporary speed restriction			not covered
			10.6	signals in temporary speed restriction			not covered
			10.7	TVP sections in temporary speed restriction			not covered
11	**Level Crossings**	**Level Crossing**				For level crossings each track of the level crossing has to be defined with the necessary information (see attributes)	complex type rail:tLevelCrossing One element has to be defined for each track of the level crossing. The connection between the tracks is not yet modelled by RailML.
			11.1	ID			attribute id (type rail:tGenericID)
			11.2	level crossing name			attribute name (type rail:tGenericName) a second attribute code (type rail:tGenericName) may be used additionally
			11.3	position on track for each LX track		The exact beginning and end of the Level Crossing has to be given, the position of the Level Crossing has to be defined (middle of the LX or one side)	attribute pos (type rail:tLengthM) additionally absPos (type rail:tLengthM) may be used
			11.4	neighbouring elements of each track			have to be calculated / derived from the track and the position on the track

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			11.5	type of controlling for each track in each direction	e.g. Hp (signal controlled), Fü (remotely monitored), Fü_mit_Stoerhalt (remotely monitored with emergency stop), Üs (with level crossing signal)	Type defines whether the LX is protected by a signal or not.	attribute protections (type xs:string) The type of controlling is only defined by a string, not an enumeration.
			11.6	security type	e.g. full, full with detector, half, without barrier	Based on barrier type different rules may apply	not covered
			11.7	all activation zones for each track, direction and route (including timers)			not covered
			11.8	time the Level Crossing needs to be secured (closing barriers etc.)		needed by the ARS	not covered
			11.9	approach time			not covered
			11.10	minimum opening time		At least needed in Germany	not covered
			11.11	level crossing TVP sections		List of TVP sections including a possible timer and a list of routes for each TVP section (for shortening the approach)	not covered
			11.12	all applicable activation requests	route request, vehicle detection, manual request, level crossing occupancy request		not covered
			11.13	all applicable deactivation requests	partial route release, vehicle detection, manual request		not covered

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			11.14	a signal delay timer compensating for the shorter approach distance for the protecting signal in an activation zone		for each track and direction	not covered
			11.15	the monitoring conditions applied for the protecting signals	list of conditions from req. list		not covered
			11.16	an activation delay timer for an activation zone, enabling the use of longer activation zones for lower speeds		for each track and direction and train type	not covered
			11.17	a timer compensating for the stopping train delay		for each track and direction	not covered
			11.18	conditions for positive car signal		Used e.g. by AZD, the signal is a white flashing light	not covered
12	**TVP Section**	**TVP Section**					complex types rail:tTrainDetector (the detector itself) and rail:tTrackCircuitBorder are part or RailML but not the TVP section itself
			12.1	ID			not covered
			12.2	TVP section name			not covered
			12.3	length of TVP section			can be derived from the detectors or track circuit borders, resp.
			12.4	TVP section type	axle counter electronic relay singel rail double rail jointless ….		can partly be derived from the detectors or track circuit borders, resp.

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			12.5	has virtual track circuit		IR: The train detection section has a virtual (duplicate) train detection section.	not covered
			12.6	has occupancy timer		The train detection section has an occupancy timer.	not covered
			12.7	has clearance timer		The train detection section has a clearance timer.	not covered
			12.8	is requested to be coded by transmission system			not covered
				is junction train detection section		IR: The train detection section is on a junction. **To be checked by IR**	
				is diamond train detection section		IR: The train detection section is on a diamond crossing. **To be checked by IR**	
13	**Track segment**	**Track segment**					complex type rail:eTrack
			13.1	ID			attribute id (type rail:tGenericID)
			13.2	Name			attribute name (type rail:tGenericName) a second attribute code (type rail:tGenericName) may be used additionally
			13.3	The associated TVP sections			not covered
			13.4	Length of the Track segment			from the elements rail:eTrackTopology/trackBegin and rail:eTrackTopology/trackEnd as part of the track
			13.5	Connected track segments (begin, end)			from the elements rail:eTrackTopology/connections

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			13.6	Associated Elements			The elements in RailML are arranged hierarchally beneath the track elements.
			13.7	gradient profile		e.g. through gradient change points	Handled by the elements rail:eTrackElements/gradient Changes of the track.
			13.8	speed profile		e.g. through speed change points	Handled by the elements rail:eTrackElements/speedC hanges of the track.
			13.9	track conditions	e.g. tunnels, bridges etc		RailML covers the following track conditions: radiusChanges tunnels bridges ownerChanges operationModeChanges trainProtectionChanges electrificationChanges powerTransmissionChanges axleWeightChanges trackConditions The trackConditions enable a list of various other track conditions.
14	**Platform**	**Platform**					not covered
			14.1	ID			not covered
			14.2	platform name			not covered
			14.3	position (begin, end)		From these positions the length has to be defined	not covered
			14.4	side of track	left / right		not covered
			14.5	stopping positions of the train			not covered
			14.6	height of platform			not covered

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
15	**Catenary Group**	**Catenary Group**					RailML only knows about rail:eElectrificationChanges/ electrificationChange The catenary group from the interlocking point of view is not covered.
			15.1	ID			not covered
			15.2	Name			not covered
			15.3	track segments (fully or partially) that belong to the catenary group			not covered
16	**Line Block**	**Line Block**					not covered
			16.1	ID			not covered
			16.2	Name			not covered
			16.3	track segment			not covered
			16.4	automatically ask for permission when a train wants to use the block			not covered
			16.5	automatically give away the permission after a train has left the line block			not covered
			16.6	memorise to give away the permission after the block is clear			not covered
			16.7	announcing a train departing by a bell			not covered
			16.8	announcing a train has arrived by a bell			not covered
			16.9	TVP section of the neighbour interlocking to be shown			not covered
			16.10	information that the train has completely left the block			not covered

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
17	**Group of points**	**Group of points**				Defines a mechanism that limits the number of simultaneously switching points to limit amperage	not covered
			17.1	ID			not covered
			17.2	Name			not covered
			17.3	points that belong to this group			not covered
18	**Group of signals**	**Group of signals**					complex type rail:eSignalGroup
			18.1	ID			attribute id (type rail:tGenericID)
			18.2	Name			attribute name (type rail:tGenericName) a second attribute code (type rail:tGenericName) may be used additionally
			18.3	signals that belong to this group			by elements rail:eSignalGroup/signalRef
19	**emergency local panel**	**emergency local panel**					not covered
			19.1	ID			not covered
			19.2	Name			not covered
			19.3	signals that belong to this panel			not covered
			19.4	points that belong to this panel			not covered
			19.5	level crossings that belong to this panel			not covered

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
20	**Balise group**	**Balise group**					complex type rail:eBaliseGroup The balise group in RailML is only a collection of balises without any own attributes or elements.
			20.1	ID			attribute id (type rail:tGenericID) of the balise itself
			20.2	Balise group name			attribute name (type rail:tGenericName) of the balise itself a second attribute code (type rail:tGenericName) may be used additionally
			20.3	Identity number of the country or region		NID_C; SUBSET026 cap. 7.5.1.86	attribute countryID (type rail:tBaliseCountryID)
			20.4	Identity number of the balise group		NID_BG; SUBSET026 cap. 7.5.1.85	attribute groupID (type rail:tBaliseGroupID)
			20.5	neighbouring elements	elements on a- and b-side		through track topology
			20.6	element position			attribute pos (type rail:tLengthM) additionally absPos (type rail:tLengthM) may be used
			20.7	Coordinates			element rail:tPlacedElement/geoCoord
			20.8	Orientation	nominal / reverse		attribute dir (type rail:tLaxDirection)
			20.9	Number of balise(s) in the group		N_TOTAL; SUBSET026 cap. 7.5.1.82	can be derived from the element within the eBaliseGroup
			20.10	distance of the balises in the group			can be derived from positions of the elements within the eBaliseGroup

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			20.11	Accuracy of the balise location		Q_LOCACC; SUBSET026 cap. 7.5.1.115	attribute linkingAccuracy (type rail:tBaliseLinkingAcc)
			20.12	Link Qualifier	Linked / Unlinked	Q_LINK; SUBSET026 cap. 7.5.1.114	not covered
			20.13	Link reaktion Level 1 nominal	no reaktion / service brake / train trip	Q_LINKREACTION; SUBSET026 cap. 7.5.1.117	attribute linkReactionAscending (type rail:tBaliseLinkReaction) Only one link reaction is included, not distinguished between Level 1 and Level 2
			20.14	Link reaktion Level 1 reverse	no reaktion / service brake / train trip	Q_LINKREACTION; SUBSET026 cap. 7.5.1.117	attribute linkReactionDescending (type rail:tBaliseLinkReaction) Only one link reaction is included, not distinguished between Level 1 and Level 2
			20.15	Link reaktion Level 2 nominal	no reaktion / service brake / train trip	Q_LINKREACTION; SUBSET026 cap. 7.5.1.117	see 20.13
			20.16	Link reaktion Level 2 reverse	no reaktion / service brake / train trip	Q_LINKREACTION; SUBSET026 cap. 7.5.1.117	see 20.14
			20.17	associated Signal			not covered
21	**Loop**	**Loop**					not covered
			21.1	ID			not covered
			21.2	Loop name			not covered
			21.3	Identity number of the country or region		NID_C; SUBSET026 cap. 7.5.1.86	not covered
			21.4	Identity number of the loop		NID_LOOP; SUBSET026 cap. 7.5.1.89	not covered
			21.5	neighbouring elements	elements on a- and b-side		not covered
			21.6	element position			not covered
			21.7	Coordinates			not covered

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			21.8	length of Loop		L_LOOP; SUBSET026 cap. 7.5.1.46	not covered
			21.9	Spread Spectrum Code for euroloop		Q_SSCODE; SUBSET026 cap. 7.5.1.133	not covered
			21.10	associated Signal			not covered
22	**Addresses**	**Addresses**				This information has to be refined after other interfaces are defined. Information about other systems connected to the interlocking, e.g. RBC, is needed.	not covered
			22.1	IP-Address	network and netmask		not covered
			22.2	location id			not covered
			22.3	ARS bus address			not covered
23	**Interlocking**	**Interlocking**				An interlocking defines an extent of the railway that is controlled by a physical interlocking.	complexType rail:tController The elements within RailML are linked to the controller they belong to.
			23.1	ID			attribute id (type rail:tGenericID)
			23.2	Name			attribute name (type rail:tGenericName) a second attribute code (type rail:tGenericName) may be used additionally
			23.3	list of adjacent interlockings		including information where other interlockings are connected (to which tracks)	not covered Can maybe be derived from the topology and the elements on it.
			23.4	min. requested time for uninterrupted power supply			not covered
			23.5	startup delay timer		for preventing commands to the interlocking system	not covered

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
			23.6	ARS area		An ARS area shall be configured to define an area where routes set by the automatic route setting system will be rejected in degraded situations	not covered
24	**RBC**	**RBC**				Further requirements for the RBC have to be defined after interfaces are defined.	not covered
			24.1	ID			not covered
			24.2	Name			not covered
25	**General configu-ration**	**General system configu-ration**					
			25.1	L1 area		A L1 area shall be configured to define an area for trains operating in L1.	can be defined by track conditions
			25.2	L2 area		A L2 area shall be configured to define an area for trains operating in L2 RBC borders are defined by using two L2 areas	can be defined by track conditions
26	**Auto Device**	**Auto Device**				An Auto Device is located on a controlled signal and is used to put the signal into ‘automatic’ mode.	not covered
			26.1	ID			
			26.2	is in auto			
			26.3	has auto reminder			

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
Following objects are not included because requirements have to be defined first							
27	**Automatic Warning System**	**Automatic Warning System**				An Automatic Warning System is located on the track a set distance before the signal and is used to inform the train driver of the aspect colour being shown on the signal. New requirement for D3: Requirements for national ATP systems have to be defined.	
			27.1	ID			
			27.2	has suppression output			
28	**Aspect**	**Aspect**				The handling of the national signalling systems within the INESS scope has to be defined. The definition of the requirements for a data model follows afterwards. The attributes given are only examples.	
			28.1	ID			
			28.2	is red		The aspect is a red light to stop	
			28.3	is green		The aspect is a green light to proceed	
			28.4	is yellow		The aspect is a yellow light to proceed with caution	
			28.5	is proceed		The aspect is a proceed	
			28.6	is stop		The aspect is a stop	
			28.7	is on		The aspect is of type on	
			28.8	is off		The aspect is of type off	
			28.9	is subsidiary		The aspect is used in association with a main signal stop.	

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
29	**ARS**	**Automatic Route Setting**				New requirement for D3: Requirements for the ARS systems in the INESS scope have to be defined. The attributes given are only examples (from Germany).	
			29.1	ARS coupling data		Needed to couple ARS in central operating centre (In Germany “ZLV-Anbindung”)	
			29.2	signals of the neighbour interlockings			
			29.3	track names for ARS end points		(In Germany “Streckenübergänge”)	
			29.4	ARS data preview time period			
			29.5	ARS data historical time period			
			29.6	ARS data announcements			
			29.7	ARS data announcement initiates route setting			
			29.8	List of ARS busses			
			29.9	signal groups for ARS		normally all entry and exit signals of a station per direction	
			29.10	range for erroneous train numbers			
			29.11	ARS announcement devices			
			29.12	ARS displays mirroring neighbour's ARS data			

#	Group	Object	Attri-bute No.	Attribute(s)	Value(s)	Comments	Covered in RailML
30	**Lamp**	**Lamp**				The inclusion of this object depends on the architecture of the INESS Interlocking concerning signals (what is within the responsibility of the interlocking, what is part of the signal control). The attributes given are only examples.	
			30.1	ID			
			30.2	is red		The lamp is red to stop	
			30.3	is green		The lamp is green to proceed	
			30.4	is yellow		The lamp is a yellow to proceed with caution	
			30.5	is subsidiary		The aspect is used in association with a main signal proceed	
			30.6	is on		The lamp is an on type	
			30.7	is off		The lamp is an off type	
			30.8	is route indicator		The lamp is a route indicator	