D 2.3 A strategy for putting methods in to practice and a formal evaluation of demonstrators

Grant Agreement N°: FP7 - SCP0 – GA – 2011 - 265647

Project Acronym: **ON-TIME**

Project Title: **O**ptimal **N**etworks for **T**rain Integration **M**anagement across **E**urope

Funding scheme: Collaborative Project

Project start: 1 November 2011

Project duration: 3 Years

Work package no.: Work package 2

Deliverable no.: Deliverable 2.3

Status/date of document: Final

Due date of document: 31/10/2014

Actual submission date: 31/10/2014

Lead contractor for this document: Trafikverket
Borlänge, Sweden

Project website: www.ON-TIME-project.eu

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination Level		
PU	Public	**X**
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)	

Revision control / involved partners

Following table gives an overview on elaboration and processed changes of the document:

Revision	Date	Name / Company short name	Changes
1	19/05/2014	Magnus Wahlborg/Trv	First issue
2	22/08/2014	Bengt Sandblad/UU	Structure, responsibilities
3	30/10/2014	Magnus Wahlborg/Trv	Final issue before publication

Following project partners have been involved in the elaboration of this document:

Partner No.	Company short name	Involved experts
	TV	Magnus Wahlborg
	UU	Bengt Sandblad
	UoB	Clive Roberts, Gemma Nicholson
	NR	Meena Dasigi
	DB	Roland Kauffman, Sebastian Fey
	IFSTTAR	Joaquin Rodriguez
	TU Delft	Rob Goverde
	TU Dresden	Thomas Albrecht
	RFI	Vito Achille Sante
	NTT Data	Anelli Matteo
	Ansaldo	Miano Antonio
	UoN	David Golightly

Executive Summary

The overall aim of the ON-TIME project is to improve railway customer satisfaction through increased capacity and decreased delays for both passengers and freight. This is achieved through new and enhanced methods, processes and algorithms.

This document is one of the final deliverables of the ON-TIME project, and is produced as an output of Work Package 2: Examination of existing approaches and specification of innovations. The aim of the document is to report on 'How to implement developed methods into practice' (Task 2.4) and 'To collect results and evaluate demonstrators' (Task 2.5).

Chapter 2 details the objectives and expected results in the project. The Technology Readiness Levels (TRLs) of the project innovations before the project start are described. The four demonstration locations, namely the East Coast Main Line, Iron Ore Line, Bologna Node and Netherlands network are briefly described in terms of their traffic types and levels and infrastructure.

Chapter 3 provides an overview of the HERMES simulation platform that has been used throughout the project. The evaluation tool which has been developed to undertake quantitative evaluation of the performed simulations is also explained, together with the measures and processes used to provide a quantitative comparator between solutions.

Chapter 4 summarises the innovations developed in the project for methods and algorithms, tooling and system integration. These were specified in the original project proposal, and form the key technical outputs of the project. Each innovation is described in terms of its: (i) objectives; (ii) research activities; (iii) developed algorithms and systems; (iv) tests and demonstrations; and (v) evaluations and results.

Chapter 5 explains the demonstration systems, simulations and demonstrations which have been undertaken in the project. Four key demonstrations were selected during the first phase of the project. The specific demonstrators were selected to allow the developed innovations to be tested on a range of scenarios from across Europe.

Chapter 6 discusses how the results of the project can be put into practice, while Chapter 7 provides a summary of the research undertaken, the achieved TRLs and future tasks.

Chapter 7 summarises the six innovations developed in the project and the demonstration on the Iron Ore Line, Sweden. Each innovation and the demonstration is described in terms of its: (i) state-of-the-art; (ii) research outputs; (iii) deliverables and proceedings; (iv) future tasks.

Table of contents

List of Tables

List of Figures

1 BACKGROUND, PURPOSE AND DOCUMENT CONTENT

This document is one of the final deliverables of the ON-TIME project, and is produced as an output of Work Package 2: Examination of existing approaches and specification of innovations. The aim of the document is to report on 'How to implement developed methods into practice' (Task 2.4) and 'To collect results and evaluate demonstrators' (Task 2.5). The document therefore serves to verify the innovations of the project (as shown in Figure 2) and show how the project results can be integrated and taken forward into practice, as well as how they can be evaluated.

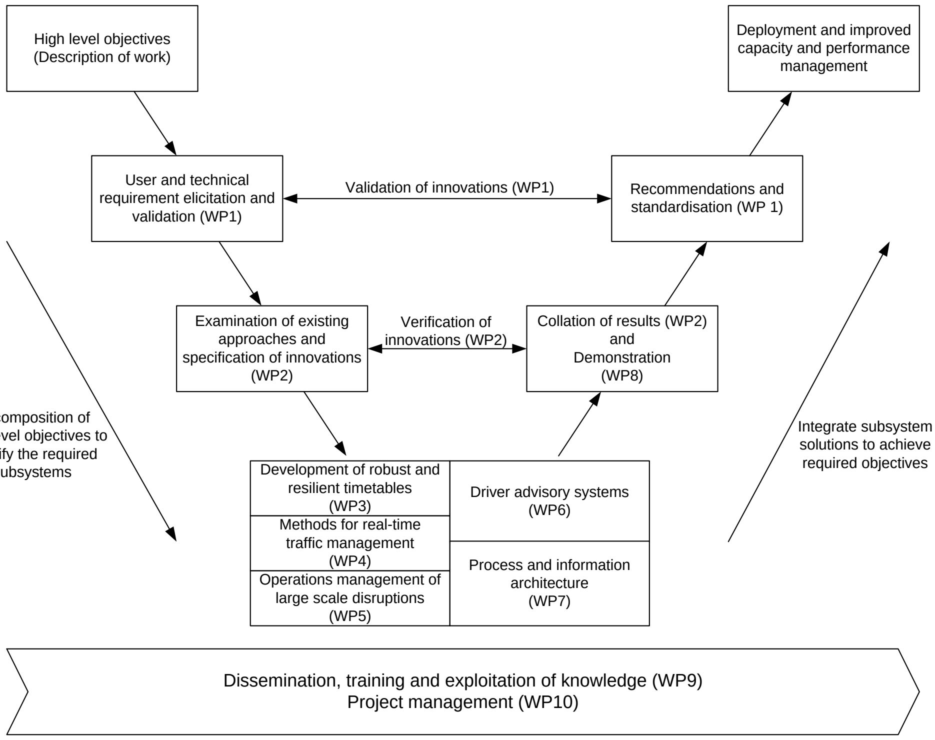

Figure 1 - Project innovations

The document is divided into a number of sections, as follows:

Chapter 2 details the objectives and expected results in the project. The Technology Readiness Levels (TRLs) of the project innovations before the project start are described. The four demonstration locations, namely the East Coast Main Line, Iron Ore Line, Bologna Node and Netherlands network are briefly described in terms of their traffic types and levels and infrastructure.

Chapter 3 provides an overview of the HERMES simulation platform which has been used throughout the project. The evaluation tool that has been developed to undertake quantitative evaluation of the performed simulations is also explained, together with the measures and processes used to provide a quantitative comparator between solutions. Chapter 4 summarises the innovations developed in the project for methods and algorithms; tooling and system integration. These were specified in the original project proposal, and form the key technical outputs of the project. Each innovation is described

in terms of its: (i) objectives; (ii) research activities; (iii) developed algorithms and systems; (iv) tests and demonstrations; and (v) evaluations and results.
Chapter 5 explains the demonstration systems, simulations and demonstrations that have been undertaken in the project. Four key demonstrations were selected during the first phase of the project. The specific demonstrators were selected to allow the developed innovations to be tested on a range of scenarios from across Europe.
Chapter 6 discusses how the results of the project can be put into practice, while Chapter 7 provides a summary of the research undertaken, the achieved TRLs and future tasks.
Chapter 7 summarises the six innovations developed in the project and the demonstration on the Iron Ore Line, Sweden. Each innovation and the demonstration is described in terms of its: (i) state-of-the-art; (ii) research outputs; (iii) deliverables and proceedings; (iv) future tasks.

2 OBJECTIVES AND EXPECTED RESULTS

2.1 Objectives

The overall aim of the project is to improve railway customer satisfaction through increased capacity and decreased delays for both passengers and freight. This is achieved through the following objectives:

Objective 1: Improved management of the flow of traffic through bottlenecks to minimise track occupancy times. This will be addressed through improved timetabling techniques and real-time traffic management.

Objective 2: To reduce overall delays through improved planning techniques that provide robust and resilient timetables capable of coping with normal statistical variations in operations and minor perturbations.

Objective 3: To reduce overall delays and thus service dependability through improved traffic management techniques that can recover operations following minor perturbations as well as major disturbances.

Objective 4: To improve the traffic flow throughout the entire system by providing effective, real-time information to traffic controllers and drivers, thus enhancing system performance.

Objective 5: To provide customers of passenger and freight services with reliable and accurate information that is updated as new traffic management decisions are taken, particularly in the event of disruptions.

Objective 6: To improve and move towards the standardisation of the information provided to drivers to allow improved real-time train management on international corridors and system interoperability; whilst also increasing the energy efficiency of railway operations.

Objective 7: To better understand, manage and optimise the dependencies between train paths by considering connections, turn-around, passenger transit, shunting, etc. in order to allocate more appropriate recovery allowances, at the locations they are needed, during timetable generation.

Objective 8: To provide a means of updating and notifying actors of changes to the timetable in a manner and to timescales that allow them to use the information effectively.

Objective 9: To increase overall transport capacity by demonstrating the benefits of integrating planning and real-time operations, as detailed in Objectives 1-8.as, to mitigate minor disturbances in railway operations (WP4).

2.2 Expected results - Innovations

The planned key outputs of the project are six innovations in the area of railway planning and operations management. These are:

Innovation 1: The development of standardised definitions and methods that can be used to create interoperable processes and tools that facilitate consistent, standardised and cross-border planning and real-time traffic management (WP1 and WP2).

Innovation 2: The development of improved methods for timetable construction that are robust to perturbations and resilient to statistical variations in operations (WP3).

Innovation 3: The development of algorithms to either automatically provide control or provide decision support to controllers, to mitigate minor disturbances in railway operations (WP4).

Innovation 4: The development of methods, processes and algorithms that provide decision support when events occur that require changes to the disposition of assets and resources, potentially across multiple networks, undertakings, operators and/or countries (WP5).

Innovation 5: The development of standardised, interoperable approaches for the communication and presentation of information to drivers and controllers in order to present the right information at the right time in a clear and consistent form (WP6).

Innovation 6: The development of an information architecture to support the communication of standardised and contextualised train control data in order that information can be exchanged between actors (operators, undertakings, networks, countries) (WP7).

2.3 State of art and TRL levels

A summary of the state-of-the-art investigation performed and reported on in D2.1.

A description of the estimated TRLs before the project start is set out in Table 1.

	Innovation	Current TRL	Planned TRL after ON-TIME
Innovation 1:	Standardised definitions and methods	**2**	**7**
Innovation 2:	Improved methods for timetable construction	**3**	**6**
Innovation 3:	Algorithms to either automatically provide control, or provide decision support to controllers	**3**	**7**
Innovation 4:	Methods, processes and algorithms that are able to provide decision support when events occur that require the disposition of assets and resources	**2**	**6**
Innovation 5:	Interoperable approaches for the communication and presentation of information	**3**	**6**
Innovation 6:	An information architecture to support the communication of standardised and contextualised train control data	**2**	**7**

Table 1 - TRL levels before project start and the planned step changes

2.4 Standardisation

Following on from the demonstrations, the relevant project results will be put forward for standardisation. It is anticipated that the key areas for standardisation are:

- A framework evaluation of different solutions;
- Data, procedures and standards;
- Architecture;
- Software interfaces.

2.5 WP contents

The innovations have been performed within different work packages:

- Improved methods for timetabling and traffic planning (WP3);
- Improved methods for perturbation handling in the operational process (WP4);
- Improved methods for handling of disturbances in the operational process (WP5);
- Improved decision support for train driving (WP6);
- Process and information architecture (WP7);
- Demonstration (WP8).

The interactions between the work packages 3, 4, 5 and 6 are described in Figure 2:

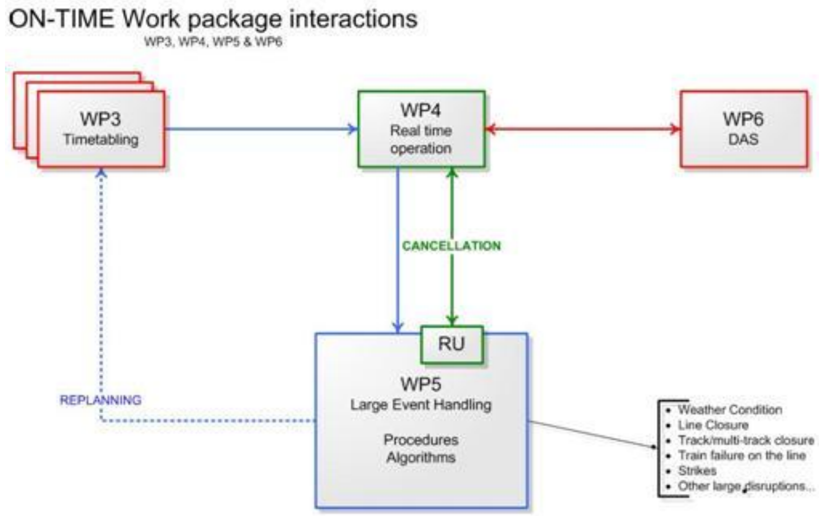

Figure 2 - Interactions between work packages

WP3 is mainly concerned with procedures and algorithms for the annual and ad-hoc timetable processes, producing a multi-layer solution for short term requests.

WP4 covers procedures and algorithms for normal traffic operation with small disturbances. WP4 starts from the timetable today and is the master plan for the real-time timetable. For WP4, the innovation is to develop automatic decision support with human intervention and/or human interaction. In WP4 the decisions are taken by the infrastructure manager.

WP5 covers procedures and algorithms for traffic operation with significant disturbances. The need for WP5 is triggered by WP4. In WP5 the problem to be solved needs decisions from both the infrastructure manager and the railway undertaker. Examples of actions are cancellation of trains, rerouting of trains and new resource plans for rolling stock and train crew.

WP6 is concerned with the provision of information and decision support to drivers. Algorithms are developed to optimise train driving strategies by helping drivers to stay inside the given boundaries developed in WP 4. Thus the total process of traffic control and train driving is optimized.

WP7 develops a Service Oriented Architecture to host software artefacts delivered by WP3, 4, 5 and 6 as web services. This includes an open and common communication and data models based on open standards (such as RailML, InteGRail, etc), common components and data flows between building blocks and services and a common, open, web-based, European service platform for railway operations, able to be further extended and enriched by other contributors.

WP8 will demonstrate that the approaches are valid and are applicable to real-life operations. For all demonstrators, it is important to show how the approach 'closes the loop'. The WP aims to consider and integrate all aspects related both to the development of the project and the real-life context.

2.6 Demonstration Locations

The demonstrators developed will illustrate a number of different locations and infrastructures.

These locations are:

1. ECML, East Coast Main Line, UK:
   - Demonstrates a complex multi track system;
2. IOL, Iron Ore Line Sweden/Norway:
   - Demonstrates a single track line and cross boarder traffic;
3. Bologna Node, Italy:
   - Demonstrates a complex node;
4. Utrecht/Arnhem/Eindhoven, Netherlands:
   - Demonstrates a complex network;

2.6.1 ECML description

Part of the East Coast Main Line (ECML) in the UK, including intersecting routes, used as an example which represents high capacity mixed traffic lines. The section of network used for the case study consists of the southern part of the ECML between London King's Cross and Sandy, and four London and South East commuter branch lines (see Figure 3):

- the Hertford loop line;
- the Northern City line;
- the Cambridge line;
- the North London line (section to the east of the ECML).

Two additional lines joining the ECML just north of London King's Cross are also included in the modelled network area. No stations on these routes are considered within the simulation, but trains may enter and leave via these lines. They are:

- the North London line to the west of the ECML;
- the Canal Tunnels link to the Midland Main line .

The most significant stations in the simulated network area in terms of passenger numbers are London King's Cross, Finsbury Park, Stevenage, Hitchin and Welwyn Garden City (on the ECML), Palmers Green, Winchmore Hill, Hertford North, Enfield Chase and Gordon Hill (on the Hertford loop), Letchworth Garden City and Baldock (on the Cambridge line), and Highbury and Islington (on the Northern City line) (Network Rail, 2008). In total, there are 42 stations included in the simulated area.

The 2018 timetable for a weekday is used, and the period of simulation is the morning peak between 7:00 am and 10:00 am. At its busiest, there are 47 trains on the route; in total during the period of simulation there are 142 passenger services run by 118 trains, and 4 freight trains running a service each.

2.6.1.1 Infrastructure

The southern section of the ECML considered for the case study runs for approximately 71 km between London King's Cross and Sandy. It consists of four tracks, one fast and one slow in each direction, for most of its length, except between Finsbury Park and Alexandra Palace, where there are 6 tracks, and it narrows to two tracks over the

Welwyn Viaduct and through two tunnels north of Welwyn North station, a known bottleneck (Network Rail, 2012).

The Northern City line, which joins the ECML at Finsbury Park South Junctions is 5.5 km long and contains five stations to its terminus at Moorgate. In the case study four of these stations are simulated, with Moorgate excluded; Old Street is on the edge of the simulated area. The line is double track along its length. Highbury and Islington Station on this line has 8 platforms, facilitating interchanges with the London Underground Victoria line, while the other three stations have two platforms each.

The Hertford loop line is approximately 37 km long and is double track along its length. It joins the ECML at both its ends: at Wood Green North Junction, just north of Alexandra Palace at its southern end, and between Knebworth and Stevenage at Langley Junctions at its northern end. The Hertford loop line provides a suburban link between London and the stations along its length, as well as serving as a diversionary route during times when the ECML is unavailable due to an incident or maintenance. It also carries freight trains between Wood Green and Langley Junction.

The simulated section of the Cambridge line, which joins the EMCL north of Hitchin at Cambridge Junction, is a double track section. The final station on this line within the simulated area is Ashwell and Morden.

The section of the North London line that lies east of the ECML is double track. It joins the ECML at Highbury Vale Junction through the single-track Canonbury Tunnel; this connection is used by freight only. Dalston Kingsland is the outer of two stations considered in the simulation on this section of the North London line. The link to the section of the North London line west of the ECML consists of a bidirectional single track line.

The Canal Tunnels link which, as part of the Thameslink 2018 Programme, will provide a link between the ECML at Belle Isle Junction and the Midland Main line at Canal Tunnels Junction, is due to open by 2018. The junction will link the ECML Up Slow line to the Up Canal Tunnel, whilst the ECML Down Slow line will be linked to the Down Canal Tunnel.

2.6.1.2 Timetable

The timetable proposed for use in 2018 is simulated for a weekday between 7:00 am and 10:00 am. The following are the service frequencies in the simulated timetable for passenger trains.

Long distance high speed

- 3 tph King's Cross to Newcastle/Edinburgh;
- 3 tph King's Cross to Leeds;
- 1 train every 2 hours King's Cross to Hull.

Suburban

- 2 tph King's Cross to Cambridge/King's Lynn;
- 4 tph London St Pancras to Cambridge;
- 2 tph London St Pancras to Peterborough;
- 2 tph London St Pancras to Welwyn Garden City;
- 2 tph Moorgate to Welwyn Garden City;

- 4 tph Moorgate to Gordon Hill/Hertford North;
- 2 tph Moorgate to Stevenage via Hertford North.

There are 4 freight trains which run during the period 7:00 to 10:00 am, all travelling between Camden Junction, joining the ECML at Copenhagen Junction and travelling via the Hertford Loop line to Peterborough.

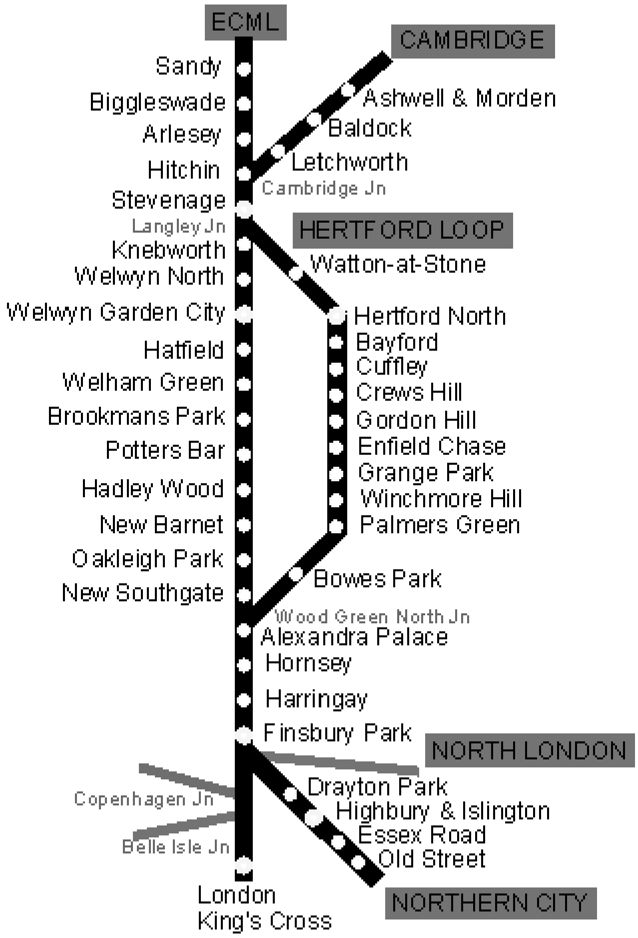

Figure 3 - The section of the East Coast main line used as a case study

2.6.2 Iron Ore Line description

The Iron Ore is a single track railway line between Narvik in Norway and Boden in northern Sweden. The traffic on this line is made up of very heavy iron ore trains (up to 8 500 tons), long trains (750 m) and other mixed traffic. The mixed traffic, and the special requirements for iron ore trains makes the optimality of planning and handling of perturbations extremely important. Delayed or cancelled trains are associated with very high costs. The section studied in the Iron Ore Line demonstrator is from Peuravaara (Kiruna) in Sweden to Narvik in Norway. On this line there are 20 stations/meeting points. The iron ore trains cannot meet at all stations, since some of them are too short. Stations are continuously being rebuilt.

Figure 4 - The Iron Ore Line, Sweden/Norway

Capacity utilization is high. Normal capacity conflicts are single track conflicts with meetings between trains and passing between fast passenger trains and slow iron ore trains.

Iron ore trains have in many aspects, the highest priority. The second priority is for long distance freight trains and passenger trains. The iron ore trains operate over 24 hours. There are several railway undertakings. LKAB Malmtrafik AB run iron ore trains Kiruna - Narvik which are, 750 m long and run at 60 km/h (loaded) or 70 km/h (empty). Green Cargo AB (Northland), run iron ore trains from Svappavaara via Kiruna to Narvik. CargoNet and Green Cargo run 100 km/h container trains, which are 1000 ton or 1800 ton and between 500 and 600 m long, between Oslo and Narvik. There are local passenger trains from Narvik, which run at 160 km/h and a couple of long distance passenger trains running at 160 km/h.

The infrastructure capacity is limited and fully utilized; there is still a high demand for more iron ore traffic. LKAB transported 28 billion tons in 2012 and plan to transport 40 billion tons in 2015 and 45 billion tons per annum by 2020. Northland will transport 5–7 billion tons by 2020. Trafikverket forecasts that Kiruna – Narvik will increase from 32 trains/day (6 passenger and 26 freight) in 2011 to 49 trains/day (6 passenger and 43 freight) in 2015 and 61 trains/day (6 passenger and 55 freight) by 2020.

In Table 2 below the structure of the Iron Ore Line is described. Between Kiruna and Riksgränsen there are short meeting stations, where iron ore trains cannot meet, at Kaisepakte and Rensjön. Between Riksgränsen and Narvik, the short meeting stations are Rombak and Björnfjell.

Station name	**Abb.**	**Distance from Kiruna, km**
Kiruna	*Kmb*	0
Krokvik	*Kv*	9
Rautas	*Rut*	20
Rensjön	*Rsn*	30
Bergfors	*Bfs*	39
Torneträsk	*Tnk*	50
Stenbacken	*Sbk*	60
Kaisepakte	*Kpe*	69
Stordalen	*Soa*	81
Abisko Ö	*Ak*	92
Björkliden	*Bln*	101
Kopparåsen	*Kå*	110
Vassijaure	*Vj*	118
Björnfjell	*Bjf*	130
Katteratt	*Kat*	141
Rombak	*Rom*	148
Straumsnes	*Sms*	156
Narvik	*Nk*	165

Table 2 - Structure of the Iron Ore Line

The timetable used in the simulations studies is from one specific day, Oct 21, 2013. The structure of the timetable, from 0:00am - 12:00pm and 12:00pm – 24:00pm is visualized below.

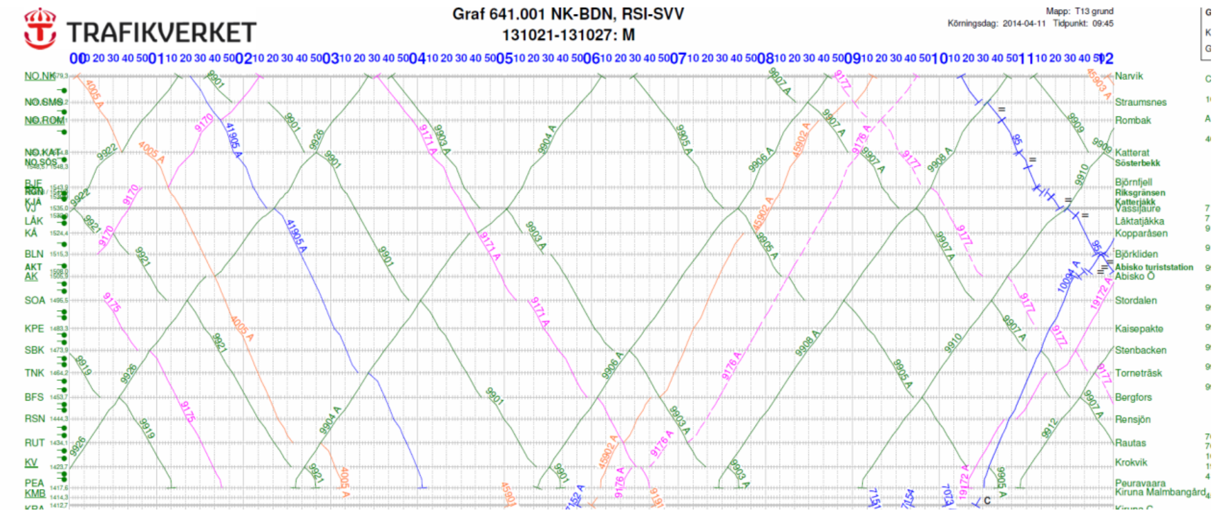

Figure 5 - The timetable graph for the Iron Ore Line, 0:00am – 12:00pm.

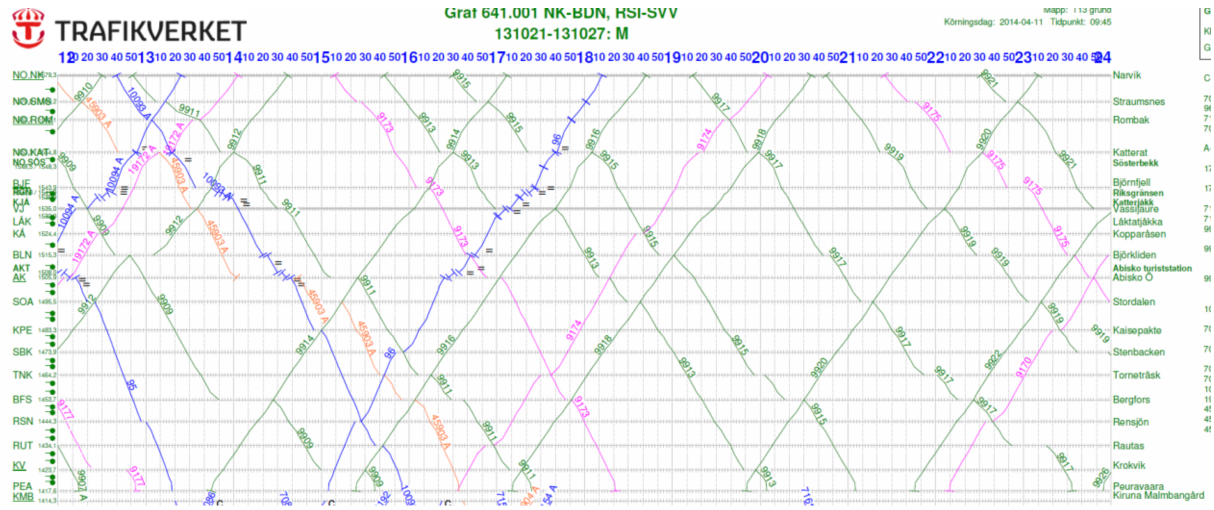

Figure 6 - The timetable graph for the Iron Ore Line, 12:00pm – 24:00pm.

2.6.3 Bologna node description

The railway node of Bologna is one of the major and most complex infrastructural areas of the Italian railways. It is the strategic junction of the Italian railway, where major traffic flows join at the "heart" of the network between the northern and central-southern parts of the peninsula.

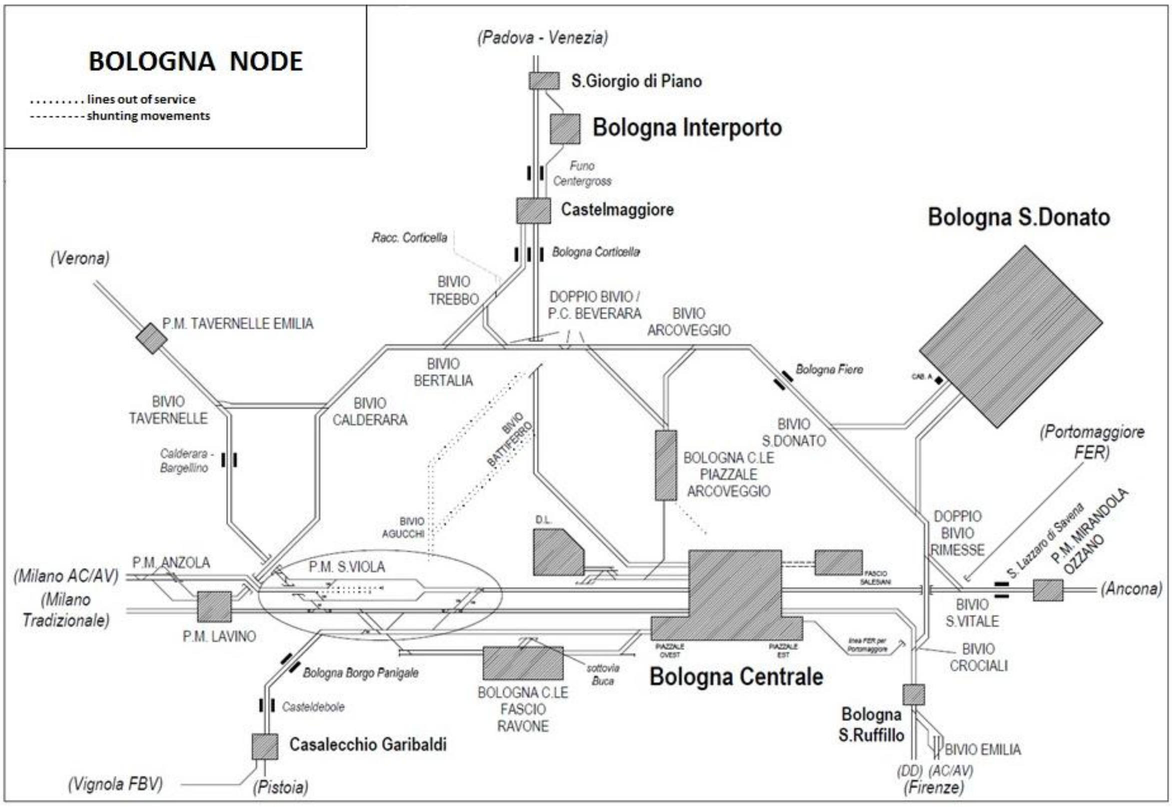

Figure 7 - Bologna node

Six main traffic lines join at Bologna, linking other important centres and lines such as:

- Florence-Rome;
- Ancona-Bari;
- Venezia;
- Verona;
- Milan;
- Pistoia.

The topology of the railway infrastructure is characterized by the "star" pattern of these lines, in addition to the railway belt line (so called "cintura") which in particular allows:

- freight traffic to by-pass the passenger station;
- re-routings to be made in case of disruptions at the "core" node if other direct lines are unavailable.

In addition to the main passenger station of Bologna, an important freight traffic yard is situated at Bologna San Donato, which generates a lot of trains directed to national and international northern borders, linking the Italian networks to other European Countries.

On the main trunk the north-south direction, in addition to conventional lines, is served by the new "parallel" high-speed line (so called AV, *"Alta Velocità"*), which passes through the underground Bologna Station (see Figure 8).

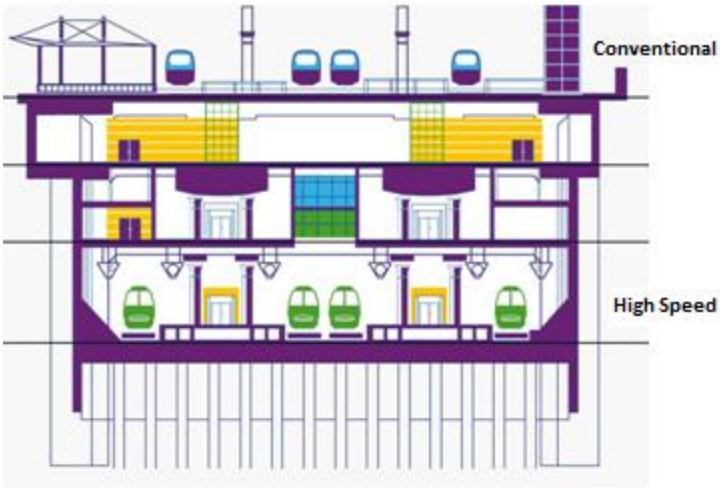

Figure 8 - Underground Bologna Station

Several commercial train operators manage their operations thanks to the Bologna Node. In addition to the major Italian operator, Trenitalia, who provide high-speed services and other passenger and freight services, high speed trains are also provided by NTV, and other services are provided by different regional companies and several freight carriers. The total number of trains is greater than 800 per day.

2.6.4 Utrecht/Arnhem/Eindhoven network description

The Dutch case study consists of a central part of the railway network in the Netherlands. It consists of the railway network bounded by the four main stations, being Utrecht (Ut) in the North, Eindhoven (Ehv) in the South, Tilburg (Tb) in the West, and Nijmegen (Nm) in the East, with a fifth main station 's-Hertogenbosch (Ht) in the middle and 20 additional smaller stations and stops, see Figure 10. Four corridors connect Ht to the other main stations.

The case study considers the timetable for a workday in 2011 between 7:00 am and 9:00 am. There are 36 trains running per hour from eight train lines in both directions, plus ad-hoc freight trains.

2.6.4.1 Infrastructure

Figure 9 shows a macroscopic view of the infrastructure. On the north side of Ht, there is a double-track bridge with one track for each direction. All trains to/from both Ut and Nm traverse this bridge. At the North site of the bridge, a junction splits the double-track into two double-track lines to/from Ut and Nm, respectively. This junction is referred to as the's-Hertogenbosch Diezebrug Aansluiting (Htda). On the South side of Ht, a junction splits a triple-track line into two double-track lines to/from Tb and Ehv. This junction is referred to as Vught Aansluiting (Vga). On the corridor to the East between Oss (O) and Nm, there is a single-track bridge (Mbrvo) which is used in two directions. Finally, at the south of station Gdm on the line Ht-Ut there is a branch line with a single-track between Wadenoijen (Wnn) and Tiel (Tl) which contains the stop Tiel Passewaaij (Tpsw).

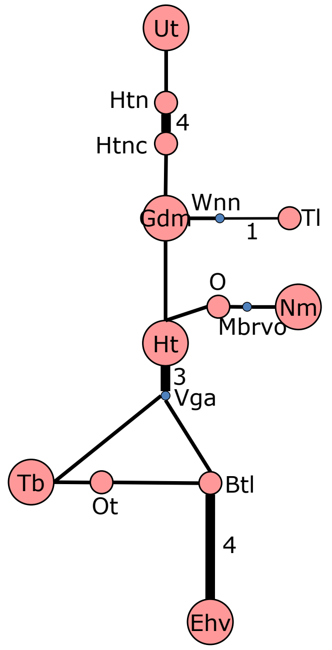

Figure 9 - Schematic macro infrastructure layout Dutch case study

The bridge on the north side of Ht is currently one of the bottlenecks in the network. In 2014, it will be replaced by a new bridge with 4 tracks and a fly-over. This will reduce the number of conflicts between trains running between Ht and Nm/Ut and will allow for a timetable that is more robust. In the ON-TIME project we used the infrastructure situation in 2012.

2.6.4.2 Line plan

The train line plan in this part of the network is taken from the 2011 timetable. It contains the following ten passenger train lines in both directions, see Figure 10:

1) Intercities
   a. Line 800: Ut – Ht – Ehv, twice per hour
   b. Line 3500: Ut – Ht – Ehv, twice per hour
   c. Line 3600: Nm – Ht – Tb, twice per hour
   d. Line 1900: Tb – Ehv, twice per hour
2) Regional trains
   a. Line 6000: Ut – Gdm – Tl, twice per hour
   b. Line 16000: Ut – Ht, twice per hour
   c. Line 13600: Ht – Tb, twice per hour
   d. Line 4400: Nm – Ht, twice times per hour
   e. Line 9600: Ht – Ehv, twice times per hour
   f. Line 5200: Tb – Ehv, twice times per hour

The intercity lines 800 and 3500 offer a regular 15 minute service between Ehv and Ut but have different origin/destinations outside this area. The regional line 13600 from Tb to Ht continues as the line 16000 from Ht to Ut, and vice versa. The line 9600 from Ehv couples in Ht to the line 4400 to Nm, and vice versa. The line 9600 from Ehv couples in Ht to the line 4400 to Nm, and vice versa.

Considering both directions the network thus contains 40 passenger trains running per hour.

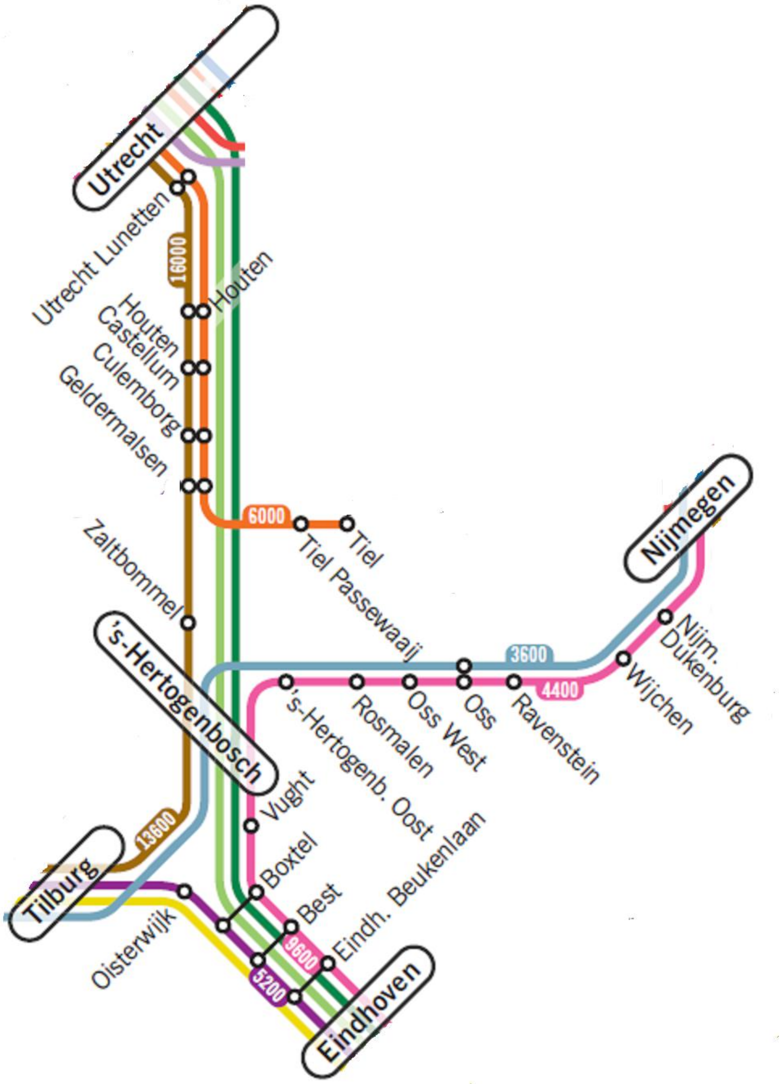

Figure 10 - Passenger line plan of the Netherlands network

In addition to passenger trains, freight trains also use this network. In the Netherlands, freight paths are scheduled in the basic hour patterns, which can be requested by freight operators in the ad-hoc timetabling phase. In theON-TIME project, the freight path was considered on the corridor Utrecht – 's Hertogenbosch – Eindhoven (and further at both ends).

3 HERMES PLATTFORM AND SIMULATION OUTPUT

This chapter gives an overview of the HERMES platform and evaluation of simulation output.

3.1 Hermes overview

The HERMES (Holistic Environment for Railway Modelling, Evaluation and Simulation) rail simulation platform is used by the ON-TIME project to provide a real time source of railway traffic and to provide a source of static data defining the network, the rolling stock and the timetable to operate over the network.

The simulator has been adapted to provide access to internal functionality through a Java based API, accessed through a user developed plug-in module. This module conforms to a functional interface that provides access to the internal data and provides requests to change the internal state of the running simulation. An important output of this module is the interface defined by the HERMES API. Although this can only be considered as a prototype at this stage, the API provides an initial specification of the generic functional properties of railway operations. The functional areas provided in the HERMES API are:

- simulation time control;
- static network specification as RailML;
- static rolling stock definition as RailML;
- static timetable specification as RailML;
- route planning;
- service modification;
- network disruption.

The API forms the external interface which connects the simulator to the WP7 web services that convey the information from HERMES to the respective consuming work packages, and to pass the requests generated by these work packages back into the simulator.

3.2 On Time Scenario Data Management in HERMES

The project required simulations to be performed on a number of disparate networks, each highlighting a particular problem of capacity and punctuality. Data was provided in several different formats including data extracted from RailSys, Open Track, Rail ML and TrainPlan, as well as a number of ad-hoc formats and data modifications.

The internal HERMES data model was populated from these sources, creating all the data needed to run a full simulation, and to create the full RailML data objects required by the work package algorithms. Initially the project standardised its output format on RailML version 2.1. However, subsequent changes to the RailML definition made under the direction of TU Dresden, incorporating the HERMES static interlocking model into the RailML schema resulted in a new version of RailML being developed, Rail ML 2.2.

The Dutch network definition requires the use of RailML+ (a supported extension of RailML for Dutch railways) that has been incorporated into the HERMES output to provide a full description of the available signal aspects when a given route is set. The aspects are defined as part of the route.

3.3 Functional Changes to Support ON-TIME

3.3.1 HERMES API

The HERMES API allows access to both static and real time information from the running simulation, including the state of the primary actors in the simulation: trains, interlocking, timetabled services and crew. This information is then made available to third party programmers either through direct access to the data structures or by registering interest in changes to the data, which are then notified at the moment the information changes. The User Module is provided by a user and defines the implementation of an API plug-in module which provides the third party access to the simulation data objects, notification of changes of state in those objects and the request interface to make changes in the running simulation.

The HERMES API has been developed as an independent HERMES component, called the HERMES Extension Point (HXP) to support the evaluation of ON-TIME work packages. The API interface and its underlying object data model architecture is illustrated in the Figure 11 below:

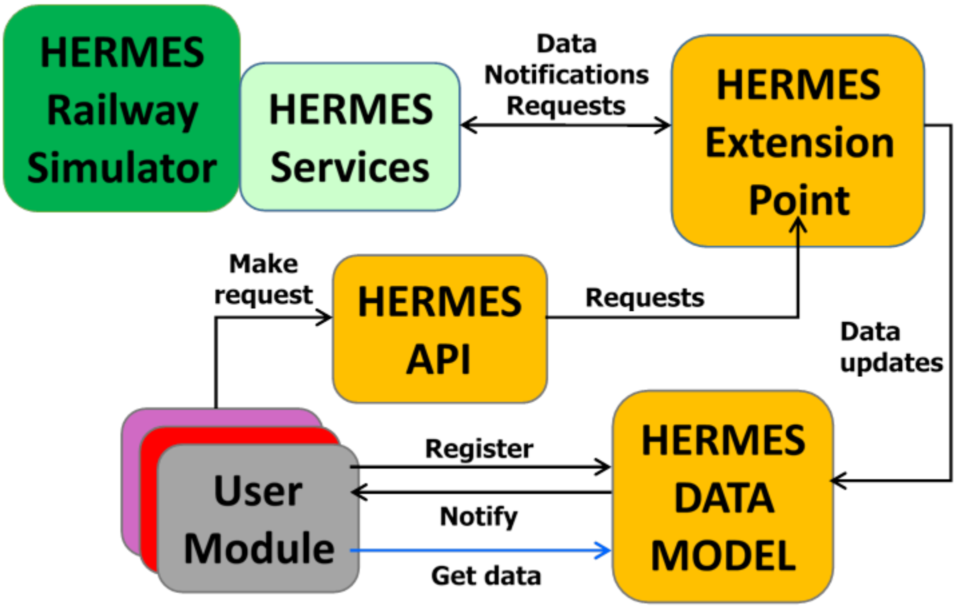

Figure 11 - HERMES API Architecture

3.3.2 Real Time Route Planning

The ability to reorder and reroute trains through the network to reduce delays and to make use of available network capacity was a key requirement of WP4. This necessitated an interface to accept a complete routeing plan for the network generated by the WP4 algorithms, and to convert these into individual routeing requests to be processed within the HERMES signalling component.

The Real Time Traffic Plan (RTTP) generated by WP4 is converted into an ordered list of train routeing requests that shall be processed in the order dictated by the plan. The plan is continuously re-evaluated by WP4 and the updated plan sent back into HERMES to perform the required routeing. HERMES sets the routes as the trains approach the respective junctions (typically while at least two green signals show between the train and the entry signal) and should the route fail to set, the signaller will make another attempt to set the route a short time later.

3.3.3 Driver Advisory System (DAS)

The HERMES simulator has been extended to provide information in the form of driver advisory messages into the core driving model. An interface has also been implemented in the HERMES API to allow third parties to provide traction and brake settings for a given train, in a DAS message. A DAS driver behaviour module is activated inside HERMES which simply passes the DAS request on to the kinematics to calculate the revised motion of the train according to the given traction or braking value in the message.

The onboard DAS module developed in WP6 requires frequent actual train position updates to be provided in order to calculate accurate and timely traction and brake requirements for the train. The HERMES API has been further extended to provide high frequency train position updates (one per second) to enable the DAS module to receive the train state data required.

A train will automatically enter DAS/ATO control when a DAS message is received and messages are provided continuously to maintain the train in DAS mode If the stream of DAS messages completes, after a configurable timeout interval, HERMES automatically switches the mode back to normal driver behaviour mode (re-installing the previous driver behaviour module). The DAS/ATO implementation also provides a simple Automatic Train Protection system (ATP) that will intervene by applying the service or emergency brake if the train passes a signal at danger, or is travelling faster than the line speed or the advisory speed indicated through the signalling system (see Dutch signalling below).

3.3.4 Dutch Signalling Model

The HERMES signalling required an implementation of the Dutch Signalling model in order to provide an accurate representation of the traffic management model when deployed on the Dutch network. This required a new signal aspect model to be developed. This model requires the setting of the signal aspects together with concomitant speed restrictions on the approach to a previously set route.

The aspect model establishes which aspect to show on the basis of the available aspects defined by the routes available in a bespoke "RailML+" format, and using a look-up table of signal to signal block length against the train speed to establish the required combination of aspect and speed advisory to show on the signal.

3.3.5 Scripted Disruptions

The API allows users to schedule a variety of generic network disruption scenarios. Each disruption is of a given type, affecting a specific element of infrastructure or rolling stock and is scheduled to start and end at scripted times. The available disruption types include:

- Dwell time disruption;
- Points failure;
- Service cancellation;
- Signal failure;
- Train speed disruption.

3.4 HERMES Evaluation

HERMES has been modified through a series of incremental changes and releases to the ON-TIME partners, who have then connected to the simulator to validate and verify the content of the HERMES output.

The RailML output from the simulator has undergone numerous changes to conform with detailed user needs and expectations, and has required some changes to the functionalities inside HERMES to meet these requirements. The resultant RailML has been formally analysed for syntactic and semantic correctness, and the data has been checked by the work packages to ensure data consistency as each incremental release of HERMES has been made to the project.

3.5 Quantitative evaluation

Quantitative evaluation uses a set of standard measures to assess the impact of the innovations developed within the ON-TIME project. It results in a set of numerical values that can be used to measure success against the aims of the project. The ON-TIME key performance indicators (KPIs) were outlined in the Quality of Service (QoS) framework, which was introduced in deliverable D1.2. Each KPI has one or more key measures, for which numerical values are obtained through the quantitative evaluation process. The QoS framework's for KPIs and their key measures are shown in Table 3. Not all of the KPIs are considered within the evaluation of each of work packages 3 to 6. Depending on the objective of the work package, only the relevant KPIs are evaluated, as shown in Table 4.

KPI	Key measures
TV	available passenger/cargo tonne km
JT	average journey time
CN	average passenger interchange time
PT	total departure delays of services at departing a station
RS	time to recover
	maximum delay
	delay area
PC	jerk above EU specified level
EG	total energy consumption by passenger/freight vehicles
RU	track usage: number of signal passes per hour rolling stock usage: number of vehicles used during simulation period

Table 3 - The key performance indicators and their key measures

	TV	JT	CN	PT	RS	PC	EG	RU
WP3	✓	✓	✓		✓		✓	✓
WP4		✓	✓	✓	✓		✓	✓
WP5	✓	✓	✓	✓	✓			✓
WP6		✓	✓	✓	✓	✓	✓	✓

Table 4 - The work packages in which the KPIs are considered

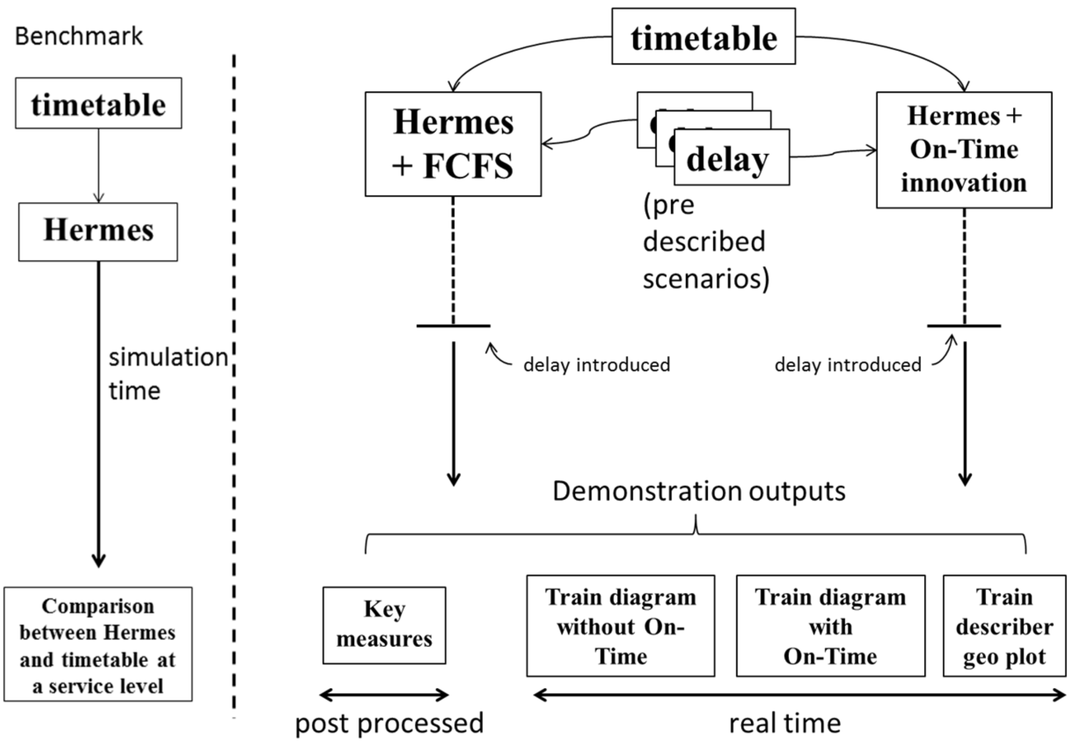

Figure 12 - Quantitative evaluation in ON-TIME

Figure 12 is a schematic of the benchmarking and quantitative evaluation process. The left hand side of the diagram describes the simulator benchmarking. In this process, the original timetable with baseline scenario (i.e. no service or infrastructure disruptions) is run in the HERMES simulator for the period specified for the given scenario. Inevitably, the simulated baseline scenario will show some small differences compared to the timetable. This process allows a comparison between the HERMES simulation and the timetable, in which any differences are quantified. The benchmarked baseline scenario for each case study location is then taken as the basis against which comparisons are made within the quantitative evaluation process, where it is described as the *reference simulation*. The quantitative evaluation makes a quantitative comparison between the reference simulation and:

- a simulation where a delayed scenario is introduced and basic operational rules are applied (*delay scenario simulation*);
- a simulation where exactly the same delayed scenario is introduced and ON-TIME WP4 algorithms are applied in order to minimise the effect of the delay on the whole system (*delay scenario simulation with algorithm*).

as shown on the right hand side of Figure 13. The delayed scenarios for each of the case study locations are described in D4.3 "Benchmark analysis for algorithms, methods, human machine interfaces using simulator tests".

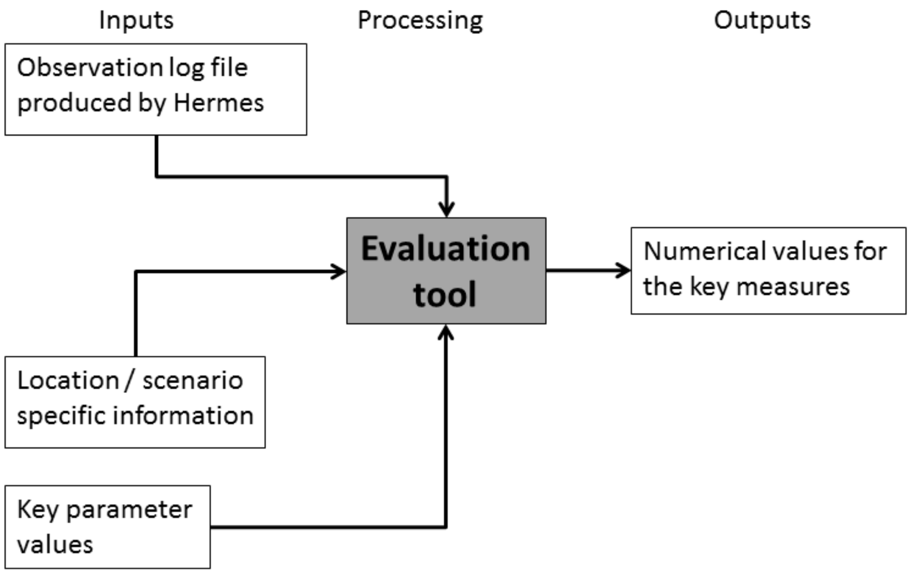

Figure 13 - Schematic of input and outputs of Matlab-based quantitative evaluation tool

The quantitative evaluation is carried out using a Matlab-based tool developed for the project. HERMES can be configured to produce an observation log file that reports the traffic events which took place in the simulation. The quantitative evaluation requires observation log files from the reference, delayed scenario and delayed scenario with algorithm simulations as inputs. It takes these and, together with certain additional information specific to the scenario, processes the simulation data to produce numerical values for the key measures, which are the outputs of the process. The quantitative evaluation tool is described further in Section 3.8.1. The benchmarking of the East Coast Main Line, Iron Ore Line and the Utrecht/Arnhem/Eindhoven network is described in Chapter 5.2.

3.5.1 Quantitative evaluation tool

The quantitative evaluation tool is written in Matlab and works on a post processing basis. It takes as input observation log files produced by HERMES and some additional tabulated information required for the computation of certain of the key measures. The observation log file is in comma separated variable format and contains the following fields that are used for the quantitative evaluation:

- observation type (station arrival/station departure/signal pass);
- train ID;
- [station/signal] [arrived at/departed from/passed];
- Time;
- train's first stop station name;
- train's last stop station name;
- unique service ID;
- cumulative energy consumption.

Each time a train either arrives at or departs from a station or passes a signal, a line containing the above fields is written to the observation log. HERMES must be configured to output log files, and given a list of stations and signals at which to produce an observation in the log file. The default used throughout this work is that observations are taken at every station and signal within the simulated network area.

Two tables are loaded to Matlab and used in the processing as follows:

- A table containing details for all the rolling stock configured for each network containing the train class, type (passenger or freight), number of carriages and seats, or freight tonne capacity;
- A table listing all the stations within the network and to which line they belong, as well as the distances between stations.

The majority of the key measures require configuration to select the stations, signals and journeys at which they are quantified, as follows:

- Journey time, energy consumption: origin-destination pairs;
- Connectivity: two leg journey (origin – connecting station – destination);
- Punctuality: stations at which total delay to be calculated;
- Resource usage (track usage): selected signals.

The key measure parameters for each of the case study locations are shown in Section 3.5. This information is an input to the quantitative evaluation tool.

The output from the evaluation tool is numerical values for each of the key measures for the reference, delayed, and delayed with algorithm applied scenarios. It is represented in three ways: stored within the structure KPI; output as a report in the Matlab console screen: and it is graphically represented for each KPI.

4 EVALUATIONS

In this chapter information is given about:

- Objectives;
- Research activities;
- Evaluations, simulations, systems and results.

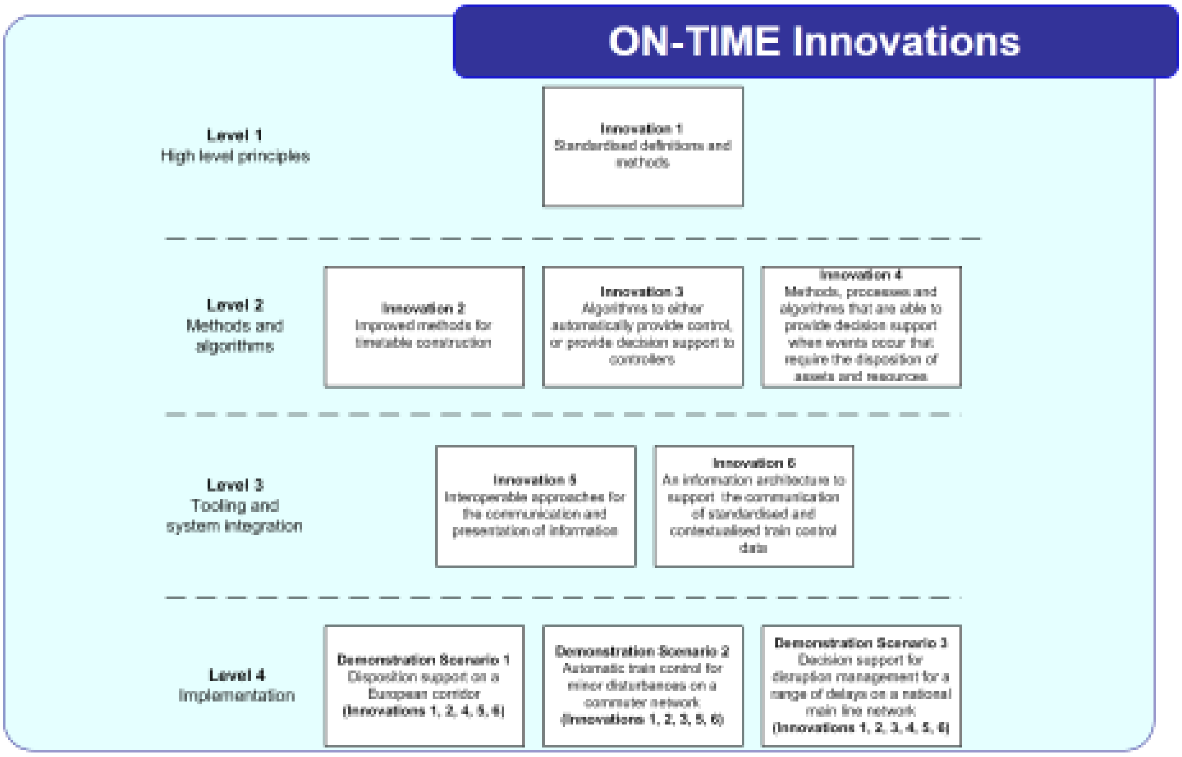

Figure 14 - ON-TIME Innovations

The results are for innovations:

- Methods and algorithms innovations 2, 3 and 4;
- Tooling and system integration innovations 5 and 6.

The workpackages covering the innovations are:

- WP 3 Timetable planning;
- WP 4 decision support operational traffic handling minor perturbations;
- WP 5 decision support for large scale disruptions;
- WP6 handling Driving advisory systems;
- WP7 IT architecture and standardised data.

4.1 Innovation 2 - Improved methods for timetable construction (WP3)

4.1.1 Objectives

The key objective of the ON-TIME timetabling work package was to reduce overall delays through the use of improved planning techniques to provide timetables that are robust, i.e., capable of coping with normal statistical variations in operations, as well as resilient to minor perturbations. The specific objectives were to:

- Develop common railway timetabling and capacity estimation methods for EU member states that reflect customers' satisfaction and enable interoperability, more efficient use of capacity, higher punctuality and less energy consumption;
- Further develop methods for robust cross-border timetables and integration of timetables between different regional and national networks improving interoperability and efficient corridor management including standardised approaches for exchanging timetable information between stakeholders;
- Design resilient timetables that can recover or reduce consequences from incidents or disturbances by exploiting feedback of performance data from operations;
- Improve timetable quality, stability, robustness, reliability and effectiveness;
- Validate the developed methods, through benchmarking, using a number of real-world case studies.

4.1.2 Research activities

The research was carried out in six tasks which are summarized below.

4.1.2.1 Task 3.1: State-of-the-art of timetabling

A review of the state-of-the-art of timetabling was carried out including an analysis of actors, processes and procedures in the seven countries involved in the ON-TIME project, as well as a literature review. The scientific literature on railway timetabling mainly considers macroscopic optimisation models without concern as to how to get accurate input parameters to set up the macroscopic model. On the other hand, the railway operations literature considers microscopic methods for calculating (energy-efficient) running times and blocking times given any infrastructure and signalling configuration, as well as microscopic methods for conflict detection and computing capacity consumption using timetable compression. The timetabling practice shows a similar separation, with either macroscopic models to compute network timetables using normative input, or microscopic blocking-time based tools for detailed planning on corridors and stations but without support for network optimisation. Timetable evaluation on feasibility, stability or robustness is typically applied – if at all – after the timetable construction using simulation tools with unclear procedures as to how the results are used to improve the timetable design. The state-of-the-art led to several recommendations for the ON-TIME timetabling research from which the 'challenging' one was taken up in the project. The results of this task were documented in the state-of-the-art report (ONT-WP03-I-EPF-008-03).

4.1.2.2 Task 3.2: Microscopic timetable computations

Task 3.2 analysed and described the parameters and computational methods for the basic building blocks of a railway timetable, including running times, dwell times, turn-around times, transfer times, and minimum headway times. Likewise, the parameters and computational methods were analysed and described for infrastructure capacity, including blocking times and the UIC timetable compression method for any signalling and ATP system. The results were included in the functional design report (ONT-WP03-I-UDB-009-03). An innovative method to estimate the rolling stock characteristics and train driver behaviour from operational data was published in Besinovic *et al*. (2013). These train dynamics parameters showed stochastic behaviour. Stochastic distributions were obtained which can be used for stochastic running time computations or sensitivity analyses to obtained validated running times rather than using just the deterministic parameters provided by the rolling stock manufacturers.

4.1.2.3 Task 3.3: Integration of timetabling and traffic control

In Task 3.3, problems in the integration of timetabling and traffic control were investigated for Sweden and the UK based on interviews with timetable planners and traffic controllers. A number of quite complicated and often interrelated problems must be solved in order to have an effective integration of timetabling and operational control processes. The causes of these problems can be classified into rules and regulations, timetabling and its tools, the quality of the timetable, the usability of the timetable for operational control, and unsatisfying feedback from operations to the timetabling process. The recommendations are all in line with the proposed ON-TIME timetabling approach. The Swedish results were reported in the integration of timetabling and traffic control report (ONT-WP03-T-UOU-021-01) and general issues and recommendations together with the UK experiences in D3.2.

4.1.2.4 Task 3.4: Functional design of robust and resilient timetable models

Task 3.4 derived formal definitions of the various levels of a timetable as well as the main timetable performance indicators, including timetable feasibility, stability, robustness, resilience, energy-efficiency, and infrastructure occupation. The variables, constraints and objectives for timetabling were described. Also a classification of Timetabling Design Levels (TDLs) was developed based on the timetable performance indicators which are explicitly considered in the timetabling design process with a benchmark of the TDLs of the seven countries involved in the ON-TIME project. Most countries operate only at TDL 1, because of a lack of conflict detection functionalities in the timetabling tools. Robustness is only considered in a limited way. Finally, a performance-based three-level timetabling framework was proposed to deliver robust conflict-free and stable timetables, with an additional procedure based on a multi-speed freight path catalogue to compute multilayer timetables that are resilient to inserting ad-hoc freight trains. The results from this task were documented in the functional design report (ONT-WP03-I-UDB-009-03).

4.1.2.5 Task 3.5: Methods and algorithms for robust and resilient timetables

In Task 3.5 an integrated timetabling architecture was developed on three levels: microscopic for highly detailed local computations, macroscopic for aggregated network optimisation, and fine-tuning for corridors. Central to the approach was the explicit incorporation of timetable performance indicators in the design process, including feasibility, infrastructure occupation, stability, robustness, resilience, travel times and energy efficiency. A common internal data format was established with consistent transformations between microscopic, macroscopic and corridor network models. The algorithms for each level were implemented, interfaces defined, and the interaction between the levels tested. The RailML standard was used for input of infrastructure, rolling stock, signalling and train line characteristics, as well as timetable output at the level of track sections including scheduled speed profiles. For this an active participation in the RailML community led to some extensions to RailML to better suit the microscopic level of detail required. The results from this task were documented in D3.1.

4.1.2.6 Task 3.6: Testing and system integration

In Task 3.6 the timetabling module was integrated with the WP7 architecture and HERMES. Functions were defined and implemented to import RailML input files provided by the HERMES data provider from the architecture into the timetabling module. Likewise, a function was defined and implemented to import a new RailML timetable file from the timetable module back into HERMES. The system was tested on the case study from the Netherlands, which led to improved timetables. A qualitative assessment of the developed timetabling approach was also executed, showing that the ON-TIME cost functions defined in D1.2 were all explicitly taken into account, the timetabling module was well integrated with the other ON-TIME modules and subsystems, and a jump from TRL 3 to TRL 6 was realized. In addition, an expert judgment of the eleven main functionalities of the developed timetabling approach was carried out by timetable planners from Sweden and the UK, resulting in a positive evaluation of the timetabling approach as a whole, as well as in terms of the individual functions. The results from this task are documented in D3.2.

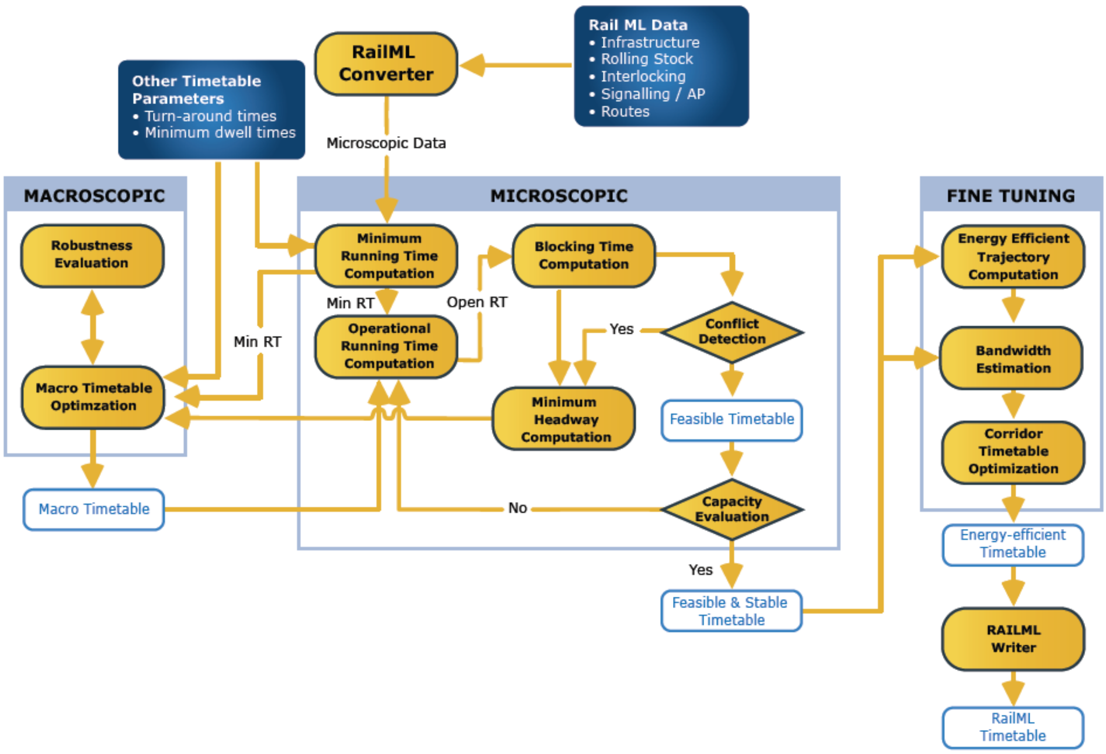

Figure 15 - ON-TIME three-level timetabling architecture

4.1.3 Developed algorithms and systems

WP3 developed an integrated set of algorithms resulting in a performance-based timetabling system, see Figure 15. Feasibility and stability are guaranteed by microscopic blocking time models, which feed a macroscopic model that optimises the timetable at the network level incorporating a Monte Carlo stochastic simulation model for robustness evaluation. These two levels iteratively compute a robust conflict-free and stable timetable. The fine-tuning model computes energy-efficient speed profiles and optimises the timetable of the local trains on the corridors between main stations using stochastic optimisation with respect to dwell time distributions and energy consumption. This represents a sustainability dimension on top of the performance with respect to delays and disruptions.

The key algorithms and systems that have been developed are:

- TU Delft: Consistent micro-macro network transformations;
- TU Delft: Microscopic running time and blocking time computations based on realisable speed profiles taking into account characteristics of rolling stock, infrastructure (slopes, speed limits), signalling (block systems, ATP), and running time supplements;
- TU Delft: Microscopic conflict detection and capacity consumption computations using blocking time theory;
- UdB: Macroscopic network timetable optimization including Monte Carlo simulation for robustness evaluation;
- TU Dresden: Energy-efficient speed profile computations;
- TU Dresden: Stochastic optimization of robust and energy-efficient timetables for local trains on corridors between main nodes using dynamic programming;
- TU Delft: Input and output using standardized RailML files with some extensions;
- TU Delft/TU Dresden: Extensions of RailML;
- TU Delft: Integration of all of the algorithms into a consistent timetabling system.

4.1.4 Tests and demonstrations

The timetabling module has been integrated with the WP7 architecture and HERMES and applied to the case study of the Netherlands, which represents a complex heavily used mixed-traffic synchronized railway network. The timetabling interface with the architecture and HERMES is only static according to the role of a timetable in railway operations. Adjusting the timetable or the scheduled operations during actual operations or simulation is the task of WP4 and WP5. Timetables for several scenarios were computed including many passenger trains and paths for fast and slow freight trains. As an example, Figure 16 shows the computed time-distance diagram of a basic hour for the corridor Utrecht-Eindhoven. Note that there is a four-track line around Houten (Htn) and between Boxtel (Btl) and Eindhoven (Ehv).

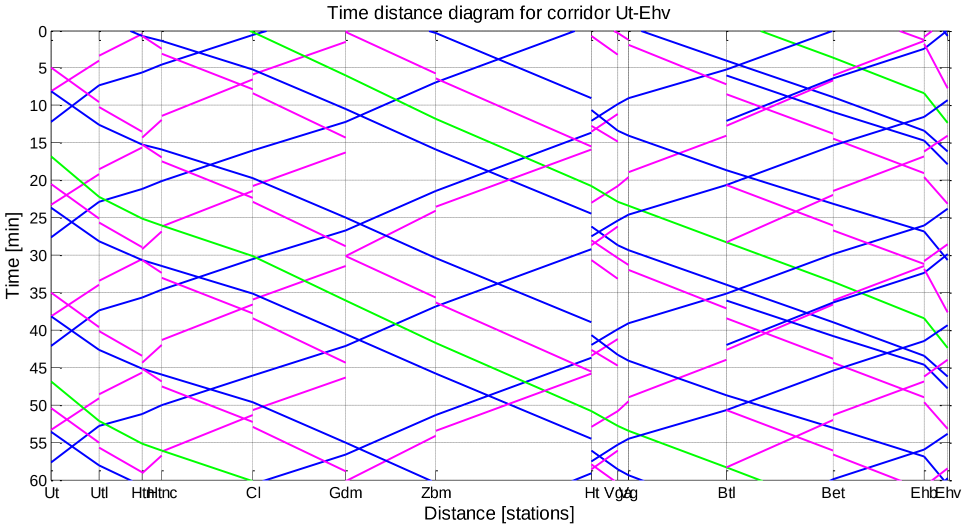

Figure 16 - Computed timetable for the corridor Utrecht-Eindhoven

The computation time for a complete, stable and robust conflict-free timetable with energy efficient speed profiles for the case study takes less than half an hour. The published timetable will, in addition, require the stochastic optimization of the local trains in the corridor, which takes more time, but this does not change the static traffic plan. The initial computations to set up the model and compute the speed profiles associated to all minimum running times and the operational running times (with running time supplements) takes 35 s, then the micro-macro iterations start. Each micro-macro iteration takes on average 2 min, with 80 s for the macroscopic model to compute the best out of 1000 solutions including the Monte Carlo simulations, and 40 s for the microscopic model to re-compute the operational speed profiles and blocking times based on the new macroscopic scheduled running times, and to set-up the new macroscopic network model with updated minimum headway times. After 9 iterations a solution is found in 1080 s. Finally, the fine-tuning model starts with 5 s to set up the corridor models, and 210 s to compute all energy-efficient speed profiles. A stable, robust conflict-free and energy-efficient timetable is thus computed in 1330 s (approximately 22 min).

4.1.5 Evaluations and results

The timetabling algorithms have been evaluated both qualitatively and quantitatively, as reported in Deliverable D3.2. The qualitative evaluation contains four components. First, it is shown that all KPIs are incorporated in the timetabling approach explicitly. Second, the integration of the timetabling module with the other ON-TIME modules is considered. All modules use a common RailML data exchange format guaranteeing consistency. Furthermore, the timetabling module shares components with other modules from WP5 (disruption management) and WP6 (driver advisory systems). Third, the applicability of the developed performance-based timetabling approach is considered including the step change from TRL 3 to TRL 6. The fourth component is that, the developed timetabling functionalities are positively evaluated in an expert judgment study.

A benchmark of all the key performance indicators defined in D1.2 has been realized based on the Dutch network relative to a reference scenario consisting of the original timetable (D3.2). The benchmarking was carried out with simulations using the HERMES simulation software. The results show that the computed timetables perform well with the same transport volume, resource usage and number of passenger trains scheduled as in the reference scenario. The journey times are sometimes slightly longer, corresponding with the aim of developing robust and energy-efficient timetables, whereas other journey times are slightly shorter. These differences also depend on the original time allowances in the reference timetable. The ON-TIME robust conflict-free timetables perform much better to perturbations in running times, whereas the reference timetable is not completely conflict-free. In the simulations the ON-TIME timetable reduced the average departure delays by 0.5 to 3.5 minutes at the five benchmark stations. Energy consumption can be reduced by 25%-28% using the provided scheduled energy-efficient speed profiles. Furthermore, the ON-TIME timetables improved connectivity by a decrease of 2 minutes of mean transfer time at the benchmark transfer station‘s-Hertogenbosch.

4.2 Innovation 3 - Methods for real time traffic management (WP4)

4.2.1 Objectives

For many years, algorithms for real-time conflict detection and resolution have been described in the scientific literature. Only recently these algorithms have been able to solve problem of practical relevance in real-time. They are, however, still not applied in daily operation in large railway networks for two main reasons:

- The benefit of the algorithms is difficult to predict;
- Operational traffic control systems (TCSs) currently in operation by nationally-acting railway infrastructure managers are not easy to extend.

In order to change this, the project defined and tested a flexible system design for railway traffic management based on extensible interfaces. This will lead to a situation where hardware equipment installed on the track-side remains usable for long periods of time, but software optimisation components and hardware used for non-safety-critical calculations such as traffic management can be easily exchanged and extended, depending on the current and future state of the art.

The objective of this part of the project was to develop a framework for a modular traffic management system (Perturbation Management Module), where independent modules collaborate.

Therefore, the following methods and tools for real-time:

- traffic state monitoring and prediction;
- conflict detection;
- conflict resolution including train speed optimization.

had to be developed or expanded to fulfil the requirements of this modular architecture (see Figure 17) and to test how a closed loop control of real-time perturbation management could work.

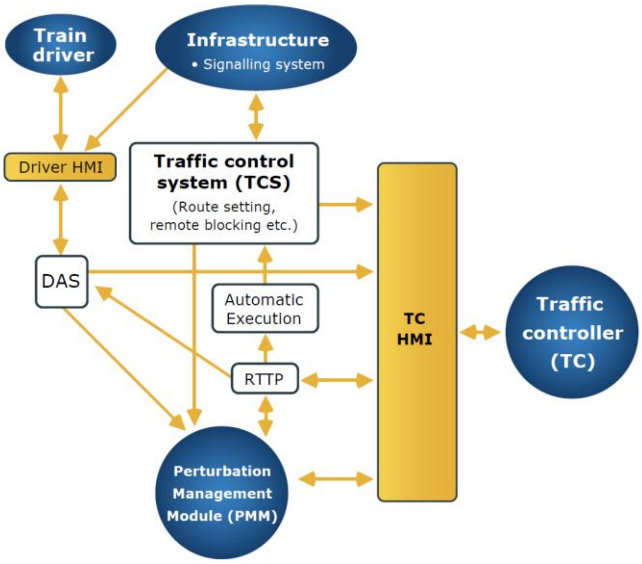

Figure 17 - Real-time perturbation management and its targeted integration in an overall railway traffic management framework

4.2.2 Research activities

An architecture for an optimal modular automatic real-time perturbation management was developed. This so-called Perturbation Management Module (PMM) can be divided into five main sub-modules that allow traffic to be effectively managed in real-time when perturbations are observed in the network (see Figure 18).

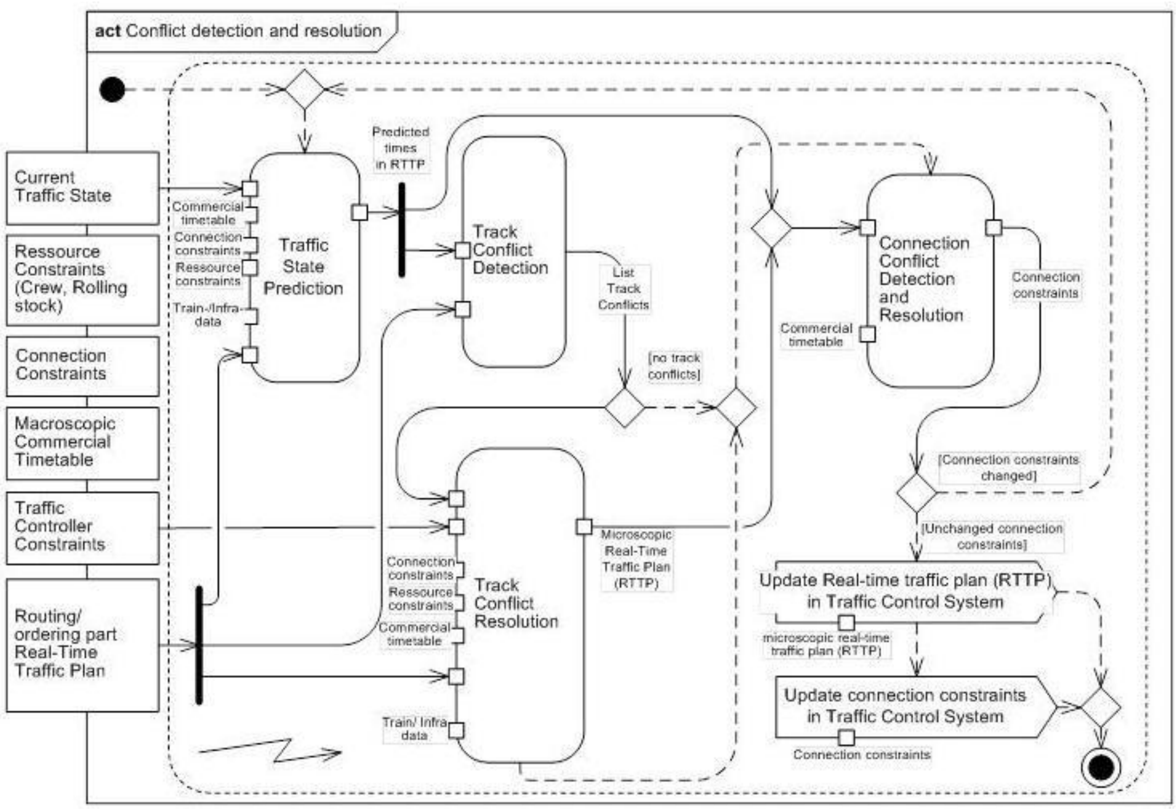

Figure 18 - Representation of main module interaction flows using SysML

The first sub-module is the so-called Traffic State Monitoring (TSM) module, which is responsible for monitoring current traffic conditions by collecting, via track-side and train-side sensors, all the information relative to both the traffic and the infrastructure.

The second sub-module is called Conflict Detection and Resolution (CDR) and represents the most important part of the PMM. It is triggered cyclically in normal operating conditions and first involves a call to the Traffic State Prediction function, which forecasts the state evolution of traffic (positions, speeds of trains) within a certain time period ahead called the "prediction horizon". If conflicts exist, the Track Conflict Resolution function is executed, which computes a new Real-Time Traffic Plan (RTTP). This real-time traffic plan is used to derive route setting commands (Automatic Execution of the Real-Time Traffic Plan).

The third sub-module is the Train Path Envelope Computation (TPEC) that aims to identify the time allowances available in real operation that can be exploited by a train to adopt an energy-saving driving strategy without running late with respect to the timetable.

A fourth sub-module is the Human-Machine Interface (HMI), which is focused on giving real-time information to the operators (dispatchers, traffic controllers) by a screen visualization of the current traffic state, e.g. through a schematic infrastructure view, as well as the predicted traffic state from the RTTP, e.g. through the so-called train graph (time-distance diagram).

The fifth sub-module is the Automatic Execution Function (AEF) of the real-time traffic plan (not given in

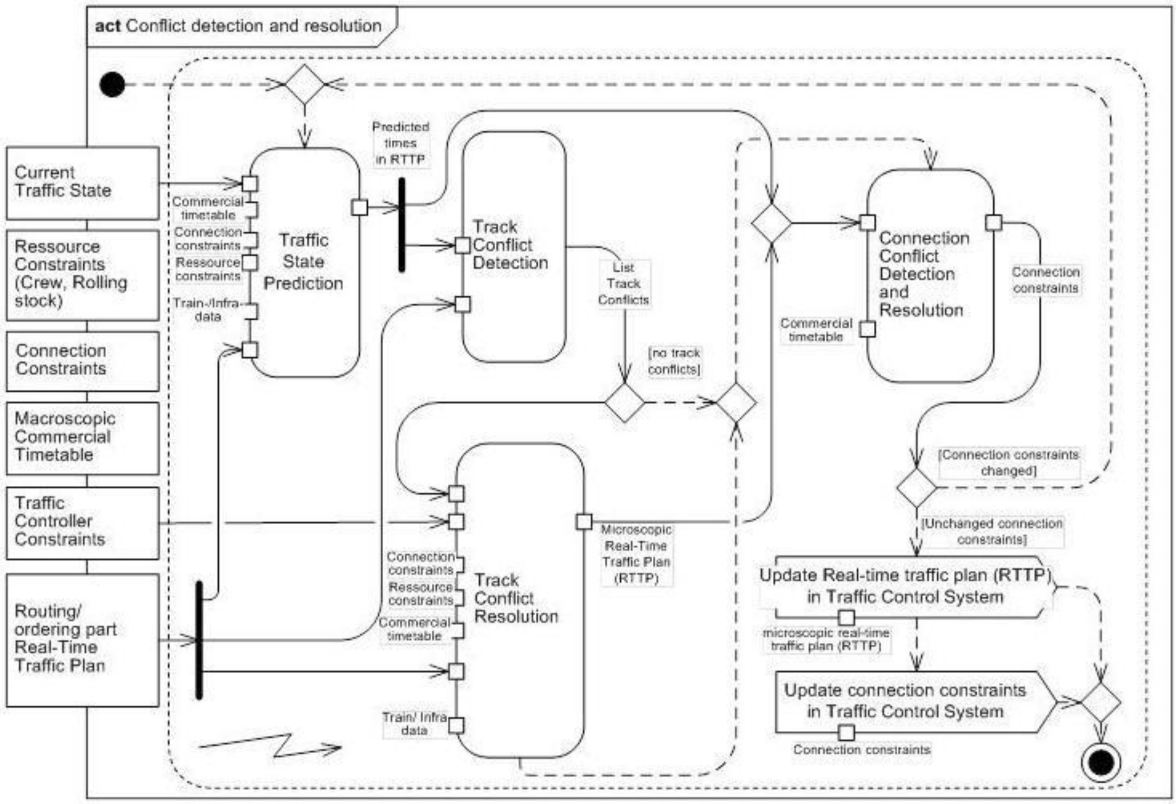

Figure 18). This module has for purpose to implement the real-time traffic plan in the field, i.e. to actually set the routes in the order and at the time described in the real-time traffic plan. It should be noted, that no local intelligence for route setting like in current automatic route setting systems is required nor supported in the developed approach, as those could contradict the solutions found by the conflict detection and resolution algorithms.

The main research and development activities were:

- Development of new software module;
- Adaptation and extension of new and existing modules to architecture and real-time requirements, i.e. all modules should rely only on data which can actually be provided by real-world railway systems and all modules need to cope with this input stream of data in real-time. Additionally, mechanisms needed to be developed to deal with varying computation times of the different modules and the consistency of the results obtained;
- Adaptation and extension of new and existing modules to RailML input data, i.e. all static (quasi-static) data about the network infrastructure, the timetable, the rolling stock characteristics and the interlocking system should only be taken from the RailML files as defined on the official website and extended within WP7. That meant significant data transformation processes within the different modules in different programming languages and with different level of detail depending on the modelling depth of the algorithms considered.

4.2.3 Developed algorithms and systems

The following list gives a brief overview over the developed algorithms and systems for real-time perturbation management:

- TSM (TUD): Speed estimation based on track occupation data, made real-time during project;
- TSP (TUD): Train running simulation based on TSM information, updated in real-time;
- TSP (TR): Stochastic short-term prediction module;
- RECIFE (IFSTTAR): provide real-time traffic plan (order and routes of train);
- ROMA (TUDelft): provide real-time traffic plan (order and routes of train);
- DEJRM (UoB): using Differential Evolution Algorithms;
- Macro-based CDR (UdB): conflict resolution by abstracting to macroscopic level;
- Connection management (EUR): concept for real-time connection management in cooperation with track conflict detection and resolution;
- TPEC (TUD): computation of robust and energy-efficient real-time train path envelopes which allow energy-efficient operation of trains under consideration of capacity restrictions;
- Automatic Execution of Real-time traffic plan (TUD): Automatic Route setting based on real-time traffic plan and prediction contained therein.
- Concept of real-time traffic plan including its generation from RailML data, update and change procedures

4.2.4 Tests and demonstrations

The entire approach was demonstrated in different architecture settings. The Conflict detection and resolution algorithms are the core optimization modules and the chosen architectures are listed with respect to these core modules.

DEJRM, ROMA and RECIFE were applied in real-time with HERMES. The effects of the algorithms with respect to the key performance indicators defined in WP1 were evaluated using an evaluation tool external to the simulation package but based on the simulation output data. Details of the evaluation can be found in the report D4.3.

DEJRM was coupled directly to HERMES to use a HERMES proprietary application programming interface. It was applied in the ECML scenario.

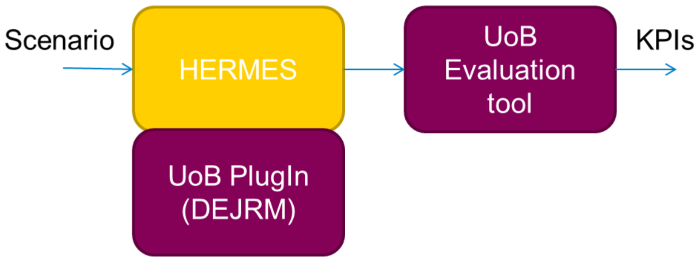

Figure 19 - Demonstration of DEJRM closely coupled to HERMES

ROMA and RECIFE were connected via the architecture (DataProvider) to HERMES using different modules developed in WP4. They were applied in the scenarios ECML, Netherlands and Iron Ore Line. It should be noted that in the figures below all green arrows represent indirect communication via the WP7 architecture module.

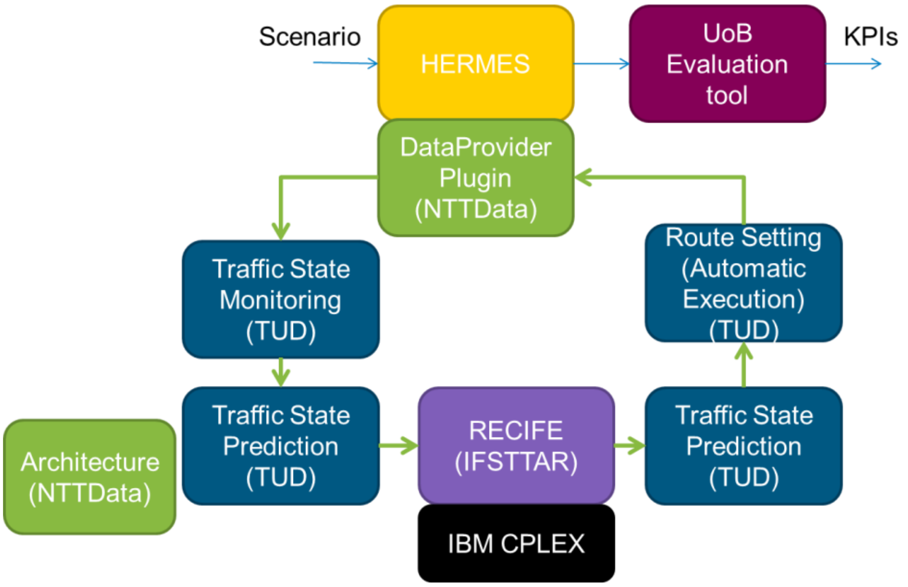

Figure 20 - Demonstration architecture for RECIFE optimization module using the project architecture

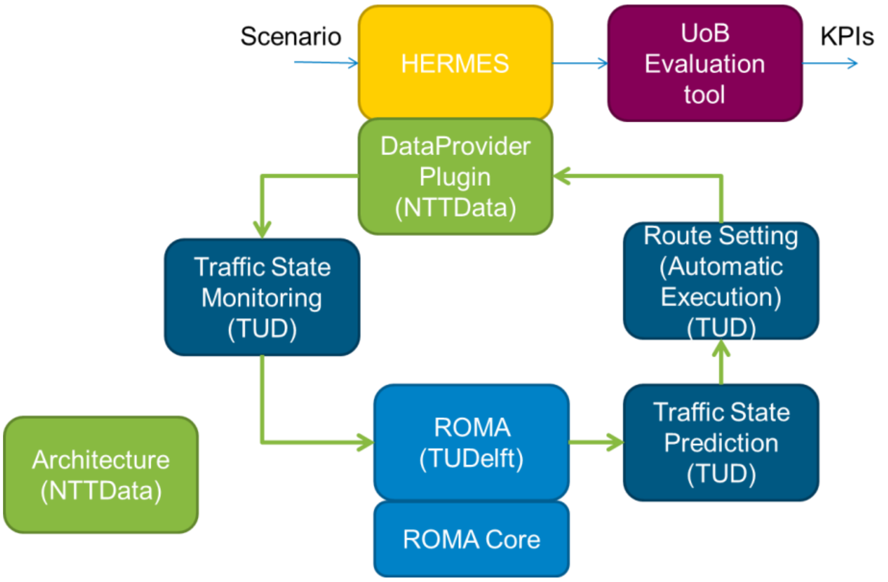

Figure 21 - Demonstration architecture for ROMA optimization module using the project architecture

The train path envelope computation module (TPEC) was tested using input data from the ROMA based control loop offline for the Dutch scenario.

The macroscopic conflict detection and resolution algorithm was tested offline using real-world problem instances provided by RFI on the Bologna node scenario.

4.2.5 Evaluations and results

The developed framework for optimal modular real-time automatic perturbation management was tested in different scenarios and at different locations. All three algorithms for conflict detection and resolution which were evaluated in HERMES simulations show significant improvements in the performance indicators:

- Deviation area (resilience) (between 30 and 70%);
- Maximum delay (up to 70%);
- Departure punctuality in stations (up to 100% reduction in some cases, on some stations and scenarios, an increase in departure delay in the order of a few seconds (smaller than 2 minutes) was found).

Because of the nature of the KPIs, less significant, but still positive impact was seen on:

- Time to recover (small reduction only, as the time to recover the big initial delays defined in the scenarios is often the determining delay for the computation of the time to recover);
- Journey time (reduction as the number of delayed trains is reduced).

All measurements were made relative to the HERMES inherent traffic management behaviour. However, the quantitative amount that can be gained for the individual key performance indicators varies strongly between algorithms and for each scenario. From the limited amount of case studies which could be carried out during the project it cannot be concluded whether one algorithm performs better than another in general.

The resource usage KPIs, the energy consumption and the transport volume KPIs remain practically identical. This was expected due to the way in which these KPIs were defined.

From this analysis it can be concluded, that the approach chosen for WP4 fully fulfils the requirements of an automatic modular real-time perturbation management system.

The undertaken simulation experiments do not allow a sound conclusion about which improvements can be made in real-world railway networks. In order to get this kind of quantitative results the control behaviour of real dispatchers must be considered in the simulation instead of the HERMES behaviour in the benchmark simulations. It is estimated that a human performs significantly better than the assumed rather simplistic behaviour implemented in HERMES. The full potential of the optimization algorithms is not yet known either, as it was only possible to implement and test a limited part of the functionality described in the functional and technical requirement specification.

4.3 Innovation 4 - Methods for operations management of large scale disruption (WP5)

4.3.1 Objectives

Recovering from a disrupted situation to a feasible state in the network requires railway operators to perform changes in the timetable such as cancelling, rerouting or re-timing trains, changing the order of departure at stations, maintaining or dropping connections between trains, and also performing the reallocation of rolling stock and changes in crew schedules. Various forms of objective functions are considered, which focus mainly on minimizing customer dissatisfaction by minimizing deviations from the intended timetable or minimizing expected delays. Further objectives include minimizing deviations from the original rolling stock allocation plan, as well as costs related to the rescheduling and cancellation of crew tasks. This recovery problem is very complex and needs to be solved in real-time; it is therefore often heuristically solved manually by the railway operators or by using fast combinatorial optimization algorithms. Furthermore, the problem is usually split up into three main phases that may be defined as timetable rescheduling, rolling stock rescheduling and crew rescheduling.

The timetable rescheduling problem is solved with a list of emergency scenarios. However, there is no emergency scenario available when several disruptions occur at the same time. A combination of contingency plans has to be used in such cases. This is often done in a non-automatic way, by using the experience of the practitioners, especially when large disruptions occur, such as the unpredictable unavailability of some tracks or train failure or line fault leading to a complete closure of the line.

Most current solutions deal with a single rescheduling phase. There are just a few approaches that integrate two phases, namely either timetable and rolling stock rescheduling, or timetable and crew rescheduling. One research goal of ON-TIME was to work on further integration of the three main rescheduling phases (timetable, rolling stock and crew rescheduling).

The main objective of this part of the project was to focus on traffic changes and resource management strategies to deal with large scale disruptions. These objectives were:

- To design and validate effective intelligent decision support strategies and tools for the optimal human supervisory control of the recovery processes in case of a large disruption;
- To evaluate the state-of-the-art in optimisation algorithms strategies and stakeholders processes and information flow for managing large scale disruptions;
- To specify the integration of the real-time traffic and asset management procedures, optimisation models and tools;
- To develop algorithms for resource management in the case of a large disruption;
- To design and validate effective intelligent decision support strategies and tools for the optimal human supervisory control of the recovery processes in case of a large disruption.

4.3.2 Research activities

4.3.2.1 Best practices

In order to evaluate the state-of-the-art in strategies, methods, stakeholders processes and information flow for managing large scale disruptions, a questionnaire was defined which aims to identify best practices which are put in use in case of railway disruptions, mainly from the point of view of the Infrastructure Manager. The questionnaire and answers are available in reports (ONT-WP05-I-RFI-0 [17,18,19,20]).

4.3.2.2 Decision making studies

The following studies into human and organisational aspects of incident management were carried out:

- Incident selection study: In order to drive user requirements capture and model building for WP5, it was necessary to determine a set of representative incidents that incur major disruptions to service. The study which determines a set of representative incidents is reported in (ONT-WP05-T-UON-008-01).
- A task analysed real incidents to identify the stages of incident management and associated activity. The results of this study are reported in (ONT-WP05-D-IFS-015-01) and published in (Golightly at al. 2014).
- The repertory grid technique was used to explore the key characteristics of selected railway disruptions. The results from this task are documented in (ONT-WP05-T-UON-011-01).
- The Critical Decision Method is a retrospective interview technique with a focus on exploring decision making. This interview method was used to formulate a list of key criteria for decision making, to identify typical decisions of operators and the information needs. The results of this task are detailed in (ONT-WP05-T-UON-011-01).

4.3.2.3 SysML specifications

In this task, the specific data structure used to manage resources has been described using the system engineering language SysML. Structural constructs such as constraint blocks were used to represent the rules that constrain the properties of the system. Moreover, all functions involved in the disruption management process have been described with SysML activity diagrams. The description includes loop interactions, object flows, control flows and synchronous/asynchronous communications between actions. These specifications are reported in (ONT-WP05-I-IFS-015-01).

4.3.2.4 State of the art of Recovery Algorithms for Real-time Railway Optimization

A review of the scientific literature on recovery algorithms for disruptions in railway management systems was carried out. This review covered four general topics, namely algorithms for train timetable rescheduling, algorithms for rolling stock rescheduling, algorithms for crew rescheduling, and algorithms that consider the integration of different phases of rescheduling. Various approaches to solve these problems were discussed in terms of type and scale of the disruptions dealt with, network infrastructure and topology, objective function and constraints considered, and optimization methods utilized. This review was reported in (ONT-WP05-I-UDB-013-01) and published in (Cacciani et al. 2014).

4.3.2.5 A framework for integration of the timetable, rolling stock and crew rescheduling phases

To obtain a full integration of the different phases of rescheduling, a framework was designed, which is shown in Figure 22. The framework consists of a closed loop in which each rescheduling phase is solved by an efficient algorithm to find a good feasible solution and get feedback from the other phases in order to obtain a good feasible solution for the whole system.

The first module updates the timetable at macroscopic level. It considers stations and important junctions in the network as nodes and open tracks as arcs. In case of a large disruption, the objective of the macroscopic timetable module is to modify the (now infeasible) planned timetable in accordance with macroscopic information on the available capacity in the network in such a way that as many train services as possible can still be operated. The possible measures are retiming arrivals and departures, short-turning trains and reordering trains. The objectives are to minimize the train delays from their planned arrival and departure times, to minimize the number of cancelled trains and to ensure a feasible rolling stock schedule.

The second module also updates the timetable, but at a very high level of detail to get an accurate model of train dynamics. The aim of this module is therefore to:

- evaluate the timetable given by the macroscopic module;
- detect potential track conflicts at the level of block section;
- compute accurate train running times;
- estimate headways among trains that consent conflict-free train services.

The objective of the third module is to adapt the (now infeasible) scheduled rolling stock circulation such that it serves the adapted timetable of the previous timetabling modules. The algorithm appoints rolling stock to every trip in the adapted timetable or, if no rolling stock can be appointed to a trip, it cancels a trip. The optimisation criteria are to minimize the cancelled trips and to minimize deviations from the original schedule.

The fourth module is a crew rescheduling module. When train services in previous modules are cancelled, some of the crew duties become infeasible. This module generates new duties for the crew in such a way that as many trains as possible in the rescheduled timetable are covered and all crew members have a feasible duty for the remainder of the day. If no crew member can be found for a certain task (or train), that task must be cancelled. The optimisation criteria are to minimize the cancelled tasks and to minimize deviations from original schedule.

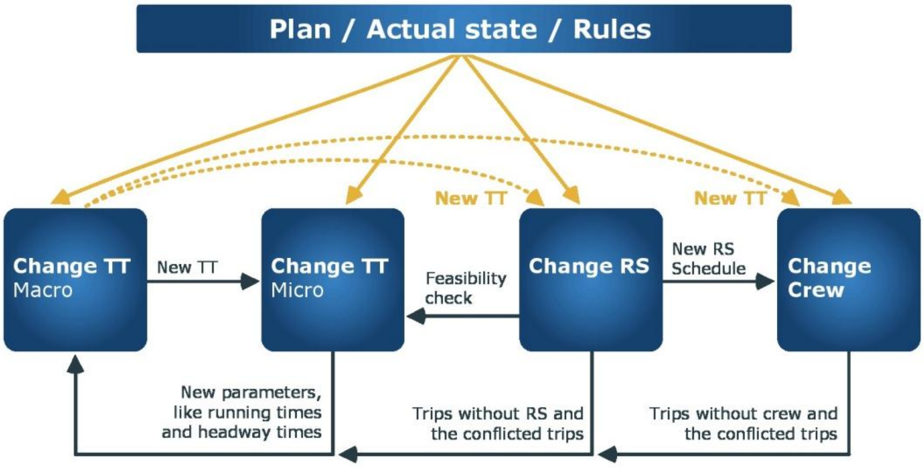

Figure 22 - Framework of closed loop for integration of the rescheduling phases

4.3.3 Developed algorithms and systems

The developed algorithms and systems for the disruptions management module are as follows:

- Macroscopic timetabling (EUR): update arrival and departure times of trains, turning trains before disrupted track sections, cancel train services;
- Microscopic timetabling (TUDelft): detect track conflicts at microscopic level, compute new running time and headways that ensure conflict-free train services;
- Rolling stock rescheduling (EUR): re-assign rolling stock to every train in the adapted timetable for the entire country, cancel train services for which no rolling stock can be assigned;
- Crew rescheduling (EUR): re-assign crew to every train in the adapted timetable for the entire country, cancel train services for which no crew can be assigned.

4.3.4 Tests and demonstrations

To test the WP5 algorithms, a case study of the resource schedules for a complete day (June 2012) from Netherlands Railways was considered. In more detail, the timetable rescheduling considers a central part of the railway network and the rolling stock and crew rescheduling considers the full network.

Tests of the WP5 framework were performed in two phases.

During the first phase, the framework was run in a laboratory environment with a large set of scenarios to evaluate the convergence of the loop and the computational performance of the framework. To generate the set of disruption scenarios, the following scenario parameters values were combined:

- Disruption locations: 's-Hertogenbosch – Oss or Utrecht – Geldermalsen;
- Blockage types: Partial or Complete blockage;
- Durations: 60 / 80 / 100 / 120 minutes;

- 61 start times between 7:00 and 17:00.

A total of 976 disruption scenarios were generated.

In the second test phase, the WP5 framework was applied in real-time with HERMES. For this test, the evaluation was carried out using a Matlab-based tool developed for the project. HERMES was configured to produce an observation log file which reports the traffic events that took place in the simulation. The Matlab tool processed the log file as input to produce numerical values for the key performance indicators defined in WP1. Details of the evaluation method can be found in the report D5.3.

4.3.5 Evaluations and results

1. Laboratory environment tests:
Over the 976 disruption scenarios, the convergence of the loop was very fast. At most two iterations were needed, because rolling stock rescheduling never cancelled additional trips. A second iteration was required in only 24% of the cases.

Regarding the optimisation criteria, on average, 12.6 trips were cancelled and the maximum number of cancelled trips was 18.

The computation performances are shown in Figure 23. For a large set of disruptions the timetable, rolling stock, and crew were rescheduled within minutes. This shows that the algorithms, both individually and combined, can be used to solve any disruption in a few minutes within a large set of practical scenarios.

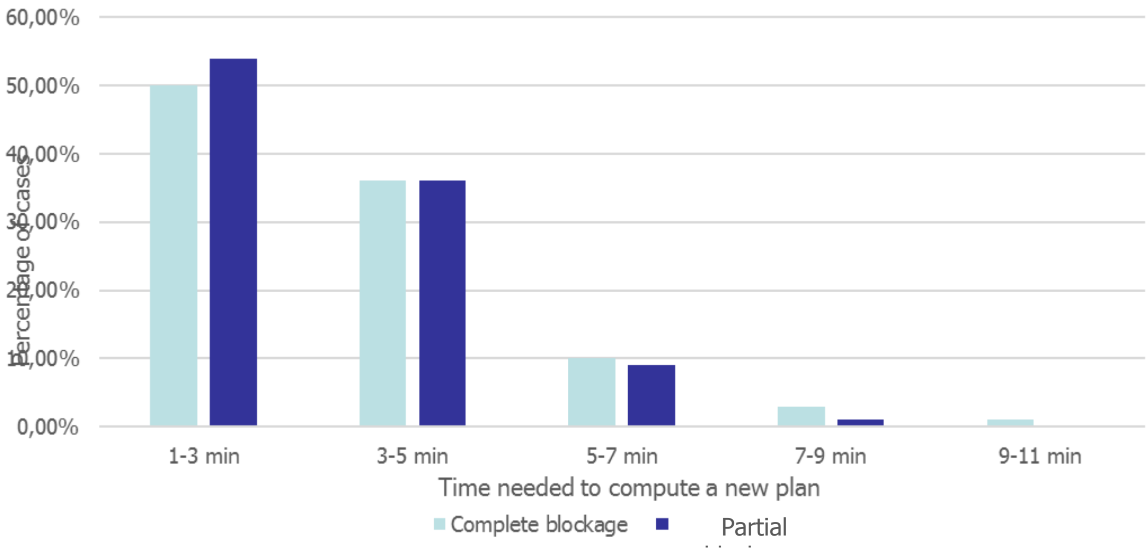

Figure 23 - Computational performances of WP5 algorithms

2. HERMES simulation test:

The evaluation with a simulation model was complex. Only one scenario could be achieved completely. The scenario was a between's-Hertogenbosch and Oss stations. The duration of the blockage was from 6:35 until 8:30.

Table 5 shows the values of the key performance indicators of the disruption scenario simulations with and without WP5 algorithms. The first column reports the values when we simulate the reference scenario, i.e. without perturbations. The second and third columns report the values when we simulate the scenario without and with the WP5 algorithms respectively. These figures show that for this scenario, the WP5 algorithms allow the traffic volume to be maintained at a high level when we compare this indicator to the situation where the disruption is not dealt with adequately. Similarly, the journey time does not increase using WP5 algorithms. Almost all the other indicator values show that the WP5 algorithms keep a good level of quality of service for the operation.

	Ref.	**Without WP5 PMM**	**With WP5 PMM**
TV(seats.km)	713782	388526	676360 (74%)
TV(#services)	26	11	22 (+11)
JT	82,38	223,13	82,34 (-63%)
CN(duration)	15	14	16
CN(options)	8	2	4
RS(deviation area)	0	1.2021e+08	1.389e+05
RS(max delays)	0	52855	212
RS(time to recover)	0	3801	0
RU(tracks)	19,8	12	16,6
RU(rolling stock)	154	146	151

Table 5 - KPI values for the scenario tested

4.3.6 Conclusions in relation to innovations

The improved decision support handling major perturbation (innovation 4) has realized a step change in terms of Technology Readiness Level. At the beginning of the project the TRL for decision support handling major perturbation was assessed at TRL3 (Analytical and experimental critical function and/or characteristic proof of concept). The expected step was to reach TRL6 (System/subsystem simulation or prototype demonstration in a railway environment).

The state-of-the-art of decision support handling major perturbation considered algorithms for train timetabling rescheduling, algorithms for rolling stock rescheduling, algorithms for crew rescheduling. Most of the papers deal with a single rescheduling phase and few integrate two phases. This shows that active research is initiated that laboratory studies validate the aims of separate elements of the innovation and that not all components are yet integrated. This state-of-the-art confirms the assessment of a TRL3.

The design a framework that consists of a closed loop in which each rescheduling phase is solved by an efficient algorithm to find a good feasible solution and gets feedback from the other phases in order to obtain a good feasible solution for the whole system establish that the modules of theses phases can work together. As set of tests of the framework has been carried out in a laboratory environment, theses framework tests can correspond to an assessment as a TRL4.

The rolling stock and the crew rescheduling should consider the entire country. This large extension of the problem can be considered as a reasonably realistic environment for testing the integration of the modules. Therefore this validation in a laboratory environment led to TRL5.

The laboratory testing of the framework done within HERMES environment that is near a railway environment has been carried out with one scenario. The few number of scenarios tested and the open issues regarding the HERMES logging function are significant obstacles to led to TRL6.

To conclude this analysis of the steps of TRL, it must be noted that the TRL6 has not been reached.

4.4 Innovation 5 - Centrally Guided Train Operation (CGTO – WP6)

4.4.1 What do we need to improve train driving?

WP6 aims to improve the information and support for train drivers. These are separated from signallers and traffic controllers and within the cab traditionally have two main sources of information they rely on when driving the train:

- Timetable and train data for their host train (e.g. a paper “timetable book” including information on the lines the train path has been assigned to, planned stops and the timing at certain intermediate points);
- Line-side (or in some cases cab-integrated) signals.

Additional radio communication may be used in certain cases (e.g. in case of perturbations with major rerouting or rescheduling of the train), but a great number of operational conditions usually cannot be transmitted by radio.

Train drivers with a lot of experience, driving the same train sets on identical routes and timetables every day may develop very effective strategies to drive their train on time and still energy efficiently in undisturbed conditions. On the other hand, less-experienced drivers or those travelling with different rolling stock on different lines (especially drivers of freight train operators) have great difficulties to perform efficiently. They are likely to drive according to the maximum speed profile unless they are running significantly ahead of schedule or forced to brake due to restrictive signal aspects. In disturbed conditions, other trains may hinder the path of the host train. Due to a lack of knowledge about these changing conditions even the most experienced drivers on their home-routes will have difficulties to perform efficiently.

In order to improve train driving we need systems, that provide drivers with the right information at the right time in a consistent way.

4.4.2 From Driving Advisory to Centrally Guided Train Operation (CGTO)

In recent years the use of Driving Advisory Systems has increased. Most of these systems focus on energy savings in undisturbed operational situations and work more or less the same way:

- Based on a comparison between the scheduled and current position (and speed) of the train, an optimized speed profile is calculated;
- The optimization uses scheduled time supplements in the timetable (allowance time and recovery time) and includes the adaption of a speed lower than the maximum permitted line/train speed and/or coasting periods in certain parts of the journey, in order to reduce the energy consumption;
- Punctuality (minimization of delay on arrival at the next scheduled stop) is usually taken into account as a hard restriction or the first optimization criterion in the calculation.

There are three important limitations of the systems described above:

- The optimization disregards other trains and the current operational situation. Hence there are no mechanisms to assure that the advised driving style is conflict-free and advice can turn out to be counterproductive:
  - A train running slow or coasting according to its advice to save energy might hinder another delayed train running behind.
  - A train running according to its advice can be hindered by another train ahead and forced to brake or stop in front of a red signal leading to a significant increase in energy consumption.
- In case of hindering a train ahead (not running according to schedule), there is usually potential to save energy for the hindered train, by adopting a slower speed or coasting. Due to the lack of knowledge about other trains in a static-timetable-based system, these potentials cannot be realised. Contrarily, once the train was hindered by a signal aspect and has a delay, a static-timetable-based system would

even advise to speed up in order to decrease the delay, even if the hindering train is still ahead. The next signal may introduce another slow down or stop if approached too early – the advice can thus be counterproductive and decrease the acceptance of driving advice by train drivers.

- In case of a disturbed situation (trains running delayed or ahead schedule), an optimal guidance for trains can reduce occupation times in bottlenecks and this way increase capacity and reduce overall delays.

To overcome these limitations, DAS need to take into account the current operational situation. Some IMs and RUs have performed significant research activities and tests on the communication of control centre decisions to train drivers. However, only one control centre connected DAS in an operationally specific area had been applied to real operations at the start of the ON-TIME project.[1] Notwithstanding the great progress in DAS, a general approach to consider the current operational conditions had remained an open issue.

The significant interest of IMs, RUs and industry in developing driving advisory systems with control centre connection went along with the need to agree on common standards of communication to ensure interoperability and prevent different developments leading to a wide range of incompatible systems within the European railway network.

It is proposed that the name "Centrally Guided Train Operation" (CGTO) will be used for the aspired DAS with real-time connection to traffic management and an interoperable standardized communication interface.

4.4.3 Objectives of research in Centrally Guided Train Operation (CGTO)

The main objective is to prove the concept of CGTO to generate and communicate driving advice based on and taking into account control centre decisions. This advice shall lead to smoother traffic flow in order to:

- decrease track occupancy in bottlenecks; and
- increase energy efficiency.

To allow the interoperable use of DAS throughout Europe the second objective is to propose a standardized data format for the communication of operational decisions (e.g. speed advice) between control centres and trains.

4.4.4 Research and development carried out

4.4.4.1 Task 6.1: State-of-the-art and relevant approaches

Altogether 22 existing driver advisory systems have been analysed (some productive and many only in state of development or prototype). The focus was placed on DAS with a real-time connection to the traffic management system – these were only 8 out of the

[1] After project start, another system came into operation, also covering the highly specific iron ore line in Sweden/Norway

22 analysed systems. For more details see reference [T6.1 report Final published on the ON-TIME website].

Based on these existing solutions the central functions of CGTO and their interaction have been identified and described.

4.4.4.2 Task 6.2 Information flow and data formats:

The data flow between these identified functions has been studied leading to the result that existing solutions and approaches distribute these functions differently between central and on-board components. This leads to the description of three alternative system architectures, which might be adapted for CGTO:

- DAS-C (mainly central intelligence);
- DAS-I (distributed intelligence);
- DAS-O (mainly on-board intelligence).

All three system architectures are promising due to the TRL they achieved in existing solutions, but have their advantages and disadvantages in certain operational environments.

In order to move towards interoperability of CGTO, a specification of an XML-interface data format has been developed. This proposed standard CGTO interface supports the three alternative system architectures enabling bidirectional communication between central and on-board components. Data flow and XML specification have been documented in deliverable D6.1.

The XML-interface and a corresponding parser have been implemented as a Java-library and tested.

4.4.4.3 Task 6.3 On-board algorithms

A benchmarking study of running time simulation algorithms as well as a feasibility check for the interface to the traffic management system was done jointly with WP4. The results have been incorporated in an existing software framework for the optimization and simulation of driving styles. This framework has been enhanced to consider the time targets of control centre decisions contained in the train path envelope. The necessary input data about rolling stock as well as track and train itinerary characteristics has been transformed from the RailML format (network view of the data) into the necessary format for train optimization (train view of the data).

The modules for trajectory computation and advice generation have been connected to the communication library developed in task 6.2 in all three possible architectures and to the HMI developed in task 6.4. Data exchange with these modules has been tested and demonstrated in real-time.

Furthermore, an ATO (automatic train operation) module has been developed which translates the driving recommendations into driving commands which are directly considered inside the HERMES railway simulator – simulating the action a driver would take given the calculated advice. This module was developed for the purpose of automatic testing and demonstration of the CGTO system in a closed loop with the simulator.

4.4.4.4 Task 6.4 On-board HMIs

The concept of Centrally Guided Train Operation relies on the presentation of information and advice to drivers enabling the driver to control the train according to the current traffic situation (guided driving). Since the driver remains totally responsible for carrying out any control commands, it is crucial to provide advice and information in an appropriate way.

Throughout the development of a human machine interface for the driver (HMI), the approach taken is to involve drivers or driver representatives as closely as possible in the design work. This has been achieved with the following methods:

- *Interviews:* Interviews have been held regularly throughout the project with professional test subjects (i.e. train drivers) representing both RUs and experts in DAS from UK, Sweden, Netherlands, Switzerland and Germany. These have highlighted general operational constraints such as the difference between freight and passenger operations, the importance of eco-driving, and experiences with existing DAS, that need to be taken into account when designing and adopting to CGTO technology.
- *Driver Workshops:* Driver workshops have been held to evaluate and validate specific elements and the overall layout of proposed HMI sample interface designs. These have also been used to investigate more generic issues associated with presenting advice to drivers, such as update frequency and timing and alerting mechanism. Also, to determine the most suitable naming / terminology conventions, especially associated with human-readable text presented to drivers at stations.
- *Driving Simulation Study:* Driving simulator studies have been carried out using the train simulator facility at the University of Nottingham. This work has been used to determine requirements for the most appropriate type of advice (e.g. speed versus timetable) and the impact that this has on driver workload and performance. Overall, there was a natural increase in workload using DAS or CGTO, but this is not necessarily a negative issue given the underload that many drivers experience. Speed-advice lead to improved performance, in comparison to both timetable and no advice.

The deliverable D6.2 describes the proposed design for a CGTO HMI. This is presented as a set of implementable HMI elements. The elements have been determined based on Human Factors constraints, such as designing for minimal workload/distraction, etc. The proposed design also considers the factors that need to be taken into account so that the HMI is successfully integrated into driving work, including acceptance by the drivers. The key elements of the proposed HMI include:

- Target speed: proposed as the primary element of the HMI (displayed as mph or km/h, depending on local preferences/standards). Where coasting is advised the target speed will be replaced by the word 'COAST' (or similar in other languages).
- Graphical preview representations (speed-distance graph) to provide an indication of driving advice over a longer period of time, thereby allowing drivers to prepare to adjust their speed in anticipation of future targets.

- Contextual advice: Iconography and text providing additional information regarding current operational condition (e.g. whether the reason for the advice is a certain conflict resolution or energy optimization in undisturbed operation).

Given the large variance of operational conditions throughout Europe, the proposed HMI elements allow some scope and flexibility to reconfigure the design. Example might be:

- To limit output to text-only, if on-board equipment does not allow graphical representation.
- To adjust colours, fonts etc to comply with local restrictions, regulations and preferences.

A prototypical implementation of the HMI has been developed and demonstrated. The prototype is connected to real-time data feeds from the on-board algorithms (see task 6.3). The primary aim is to demonstrate the concept of CGTO to support and improve both, capacity management and energy efficiency. The final design of the proposed HMI (presented at INNOTRANS) is shown in Figure 24.

Figure 24 - An Example of the ON-TIME Sample HMI

4.4.5 Development carried out for test, demonstration and evaluation

Work package 6 aimed to provide a complete demonstrator for CGTO working in closed loop with the conflict detection and resolution algorithms (WP4) and the simulator. Besides the implementation of the CGTO communication interface (see chapter 4.4.4.2), on-board-algorithms (chapter 4.4.4.3) and HMI (chapter 4.4.4.4) there was the need to implement certain pieces of software which would not be needed in a real-world system, but for the purpose of demonstration, validation and evaluation within the ON-TIME project:

- A plugin for the HERMES simulator has been developed, delivering train position data at short time intervals in order to simulate an accurate working on-board positioning system, based on which the CGTO advice can be calculated.
- The HERMES driver module has been extended to simulate more realistic driver behaviour. This included changing from only few discrete control lever positions available to semi-continuous lever settings as used in modern real-world rolling stock.
- The existing HERMES driver machine interface, enabling manual control of a train in the simulation, has been adapted according to the changes mentioned above.
- Since for evaluation a number of trains should run with CGTO in a scenario, it is not possible to control them manually. Therefore a module has been developed, which translates the driving recommendations directly into action on machine controls which are directly considered inside the HERMES railway simulator. Although the purpose of this module was to simulate the action a driver would take given the calculated advice, in fact it offers the functionality which would be needed to get from CGTO to ATO (automatic train operation, where the driver does not act on train controls in normal operation).

4.4.6 Test and evaluation results

The specified and implemented communication interface (see chapter 4.4.4.2), the extended on-board algorithms (chapter 4.4.4.3) and the HMI demo-implementation (chapter 4.4.4.4) have been tested and validated independently as well as working together in the CGTO control loop. Given an existing Train Path Envelope the validation was successful for all steps from trajectory calculation until displaying the advice, including re-calculation of advice and/or trajectory in case that the train did not perform according to the advice.

There are two steps missing to actually achieve complete validation of the CGTO concept:

- Drivability of the advice could not be validated, due to several issues in the interaction of the provided data with the HERMES simulator simulated signal aspects and ATP functions were not consistent to the real-world performance. These issues may have their origin in the existing RailML infrastructure data, where signaled speed restrictions are still not fully included. Unfortunately these issues could not be fixes until the end of the project which made it impossible to prove drivability of the advice given.
- CGTO based on dynamically re-calculated train path envelopes (TPE) could not be tested. Due to a lot of bug-fixes which had to be carried out to reach the closed-loop for WP4 there was no time left to finish the connecting of the TPE computation module to WP4 algorithms. Since this would have to be carried out by the same researchers as working on WP4, the focus was to get WP4 working first which took until October 2014.

There are at least two systems working according to the CGTO concept in real world operations. ON-TIME proposes a standard for the communication between IMs and RUs in CGTO systems (see chapter 4.4.4.2) which has been validated. The project has also delivered a demo-implementation of CGTO including this standardized interface. Since all of the particular functions of the designed CGTO system have been validated, the project team is strongly convinced that the CGTO based on this standardized interface can be used in the future. Nevertheless, the goal to evaluate the whole CGTO system in closed loop could finally not be achieved within ON-TIME and should therefore be addressed in future projects. Once the missing links in the closed loop are filled the KPIs as defined in WP1 could be used to evaluate the impact of CGTO on energy efficiency as well as punctuality/ journey time.

4.5 Innovation 6 - Process and information architecture (WP7)

4.5.1 Objectives

The main objectives of the Process and Information Architecture were to create a configurable, reliable and efficient service network to integrate all the algorithms and modules developed in the ON-TIME project. The point of introducing innovation into Railway management systems is that most of current systems are very critical to the management and operations of the network and, just because of that, the idea to swap them with something more innovative is not a realistic scenario. ON-TIME proposes a different approach: a distributed network of modules that will initially complement the current TMS and could progressively integrate different key processes beginning the gradual modernization of the railway IT infrastructures.

The last key objective was to create a common data format and standards to promote software interoperability within the European Union.

4.5.2 Research Activities

During the research phase, several technologies and middlewares were analyzed to determine the best solutions to implement a TMS Cloud of services. One of the first tasks at hand was to determine a common data modelling for the domain. The most promising alternatives were the OpenTrack format and the RailML format. RailML was preferred because the format is completely open and it was possible to cooperate with several different players in the Railway industry and the System Integration industry to refine it. During the ON-TIME project, the consortium took an active stand with the RailML initiative and several improvements were made, such as the introduction of a proper interlocking model, the detailing of the timetable formats and several different tunings, for example adding more information for timetable statistics and delay management.

On the technological side, the architecture needed an engine able to interpret the context of railway operations, to understand the meaning of a massive number of different messages that need to be interpreted to infer key information, such as: traffic flow, delays, conflicts and disruptions.

The first choice was to use a Complex Event Processor, such as NEsper, which provides a rule-based engine capable of carrying out causal and temporal correlation between a massive number of messages. During the research period, it was clear that some modules of WP4 needed to have several CEP functionalities to cope with small disruptions, therefore, for the sake of efficiency, the CEP module became just another module (in this case the WP4 TMS) plugged inside the architecture. To provide an efficient method of dispatching messages and create specific data flow from message producers to message consumers, the architecture adopted a high-speed queueing system called RabbitMQ.

To cope with massive amounts of data, we needed an equally capable data storage. Since the ON-TIME messages were defined with dynamic payloads, in order to maximise flexibility, the architecture employed a NoSQL solution and MongoDB was chosen from several competitors because of its own history of enterprise applications.

Since the goal of the project was to implement an infrastructure for distributed services with a great deal of flexibility, modularity and ease of integration, it was decided to use REST-based services to provide a common data protocol for all communication. The architecture not only exposed APIs but also described the resources, such as the timetables and the state of the network collected from the systems and modules connected to the Architecture.

The last part of the research was focused on the consolidation of the given technologies on a reference architecture, with implementation standards and detailed specifications on the integration patterns between modules and TMS, in this case emulated by the HERMES simulator.

D7.1 and D7.2 detail the results of the research during the activities of WP7.

4.5.3 Developed Components and Systems

The first developed component of WP7 was the basic building block of all the other components: a massive model library that maps RailML entities in actual OOP objects and allows for fast serialization and deserialization of such objects into and JSON or XML format, used by the platform APIs and modules.

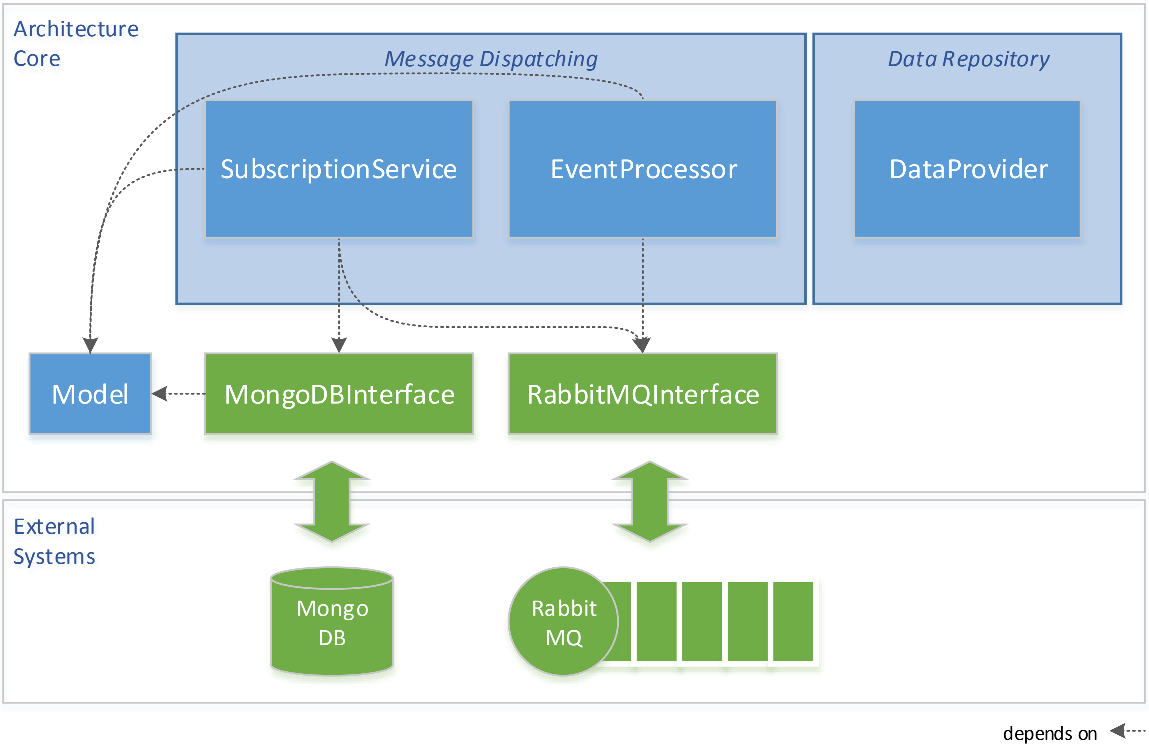

Figure 25 - The ON-TIME Architecture Core Modules

The actual ON-TIME Architecture is a collection of .NET components, and it is based on these modules:

- A *Subscription Service*, to allow modules to subscribe and/or publish messages;
- An *EventProcessor* that will dispatch and route messages, integrated with the RabbitMQ queues;
- A *DataProvider* that wraps data owned by external systems (in the demonstration case, the TMS simulated by HERMES), such as timetables and infrastructure state;
- A *Security Layer*, to profile and manage subscription/publishing and message affinity between module;
- A *Dashboard* to configure the security and messaging components and monitor basic operational parameters, like the number of active communication channels, queue length, number of subscribers and number of publishers.

To ease the integration of modules and to have a template for the architecture usage, an *Integration Library* was used as a base for the test-driven development of the architecture. It is a framework which eases common tasks and interactions with the architecture (such as registering a service, a user, publishing or consuming messages, and so on). It was used to integrate the following modules with the architecture:

- TMS (TUD): General context management, analysis of events and dispatching of Optimization Events;
- ROMA (UOD): Predictive analysis of delays, reaction to events from TMS for optimization of small perturbations;
- Recife (IFSTTAR): Predictions for energy efficiency, reaction to events from TMS;

- HERMES (Graffica): General simulation of a Railway node.

To integrate the modules, the following methodology was adopted to standardize the activity flow and maximize efficiency:

- **Interview:** the first step for integration was to understand module specific needs in terms of data flows exchanged with the other actors and with the architecture itself. The aim of this phase was to completely define all the data flows and transform them in static data to be provided or events to be implemented.
- **Handlers and services creation**: once data flow requisites had been defined, the required events were modelled and the data needed was provided.
- **Module Integration:** when all the necessary handlers were implemented in the architecture and defined in the Integration Library, the integration with the module started. The goal of this phase was to produce a working prototype for the module to proceed to the test phase.

To demonstrate the flexibility of RailML, a module was developed for the Italian Railway Network able to dump RailML or HERMES native data from the AutoCAD® schematics of Railway Infrastructures.

For a valid scenario creation the following data creation steps were needed:

- **Infrastructure**: starting from the AutoCAD® schematics and from additional data structures relative to railway components, an adjacency graph was created. Using this graph, all the components needed by the infrastructure were inferred, proceeding with gradual steps from the less complex entities, like Segments and Sections, to the most complex and interconnected ones, like TDSections and Switches. In order to test the goodness of the data produced, both RailML and HERMES native data were generated. The latter was used to check visually and programmatically the correctness of the output.
- **Rolling Stocks**: using official data provided by RFI on rolling stocks, a standardized data format was generated. The format was agreed with RFI to be easily produced by their systems and easily manageable by the algorithms.
- **Timetable:** the generation of a correct timetable was divided into two interconnected phases:
    - **Service generation:** a standardized data format was generated using official data provided by RFI on timetables. The format was agreed with RFI so that it could be easily produced by their systems and easily managed by the algorithms.
    - **Journey generation:** the AutoCAD® infrastructure model was complemented with information from other RFI systems, to select only the permitted journeys inside the stations. The process was completed with support from Graffica for the necessary heuristics in the route generation of the trains within the scenario.

4.5.4 Evaluations and Results

Aside from integration tests, to evaluate the reliability of the architecture, a session of load tests was executed, simulating different scenarios using standard Intel Core i3 computers.

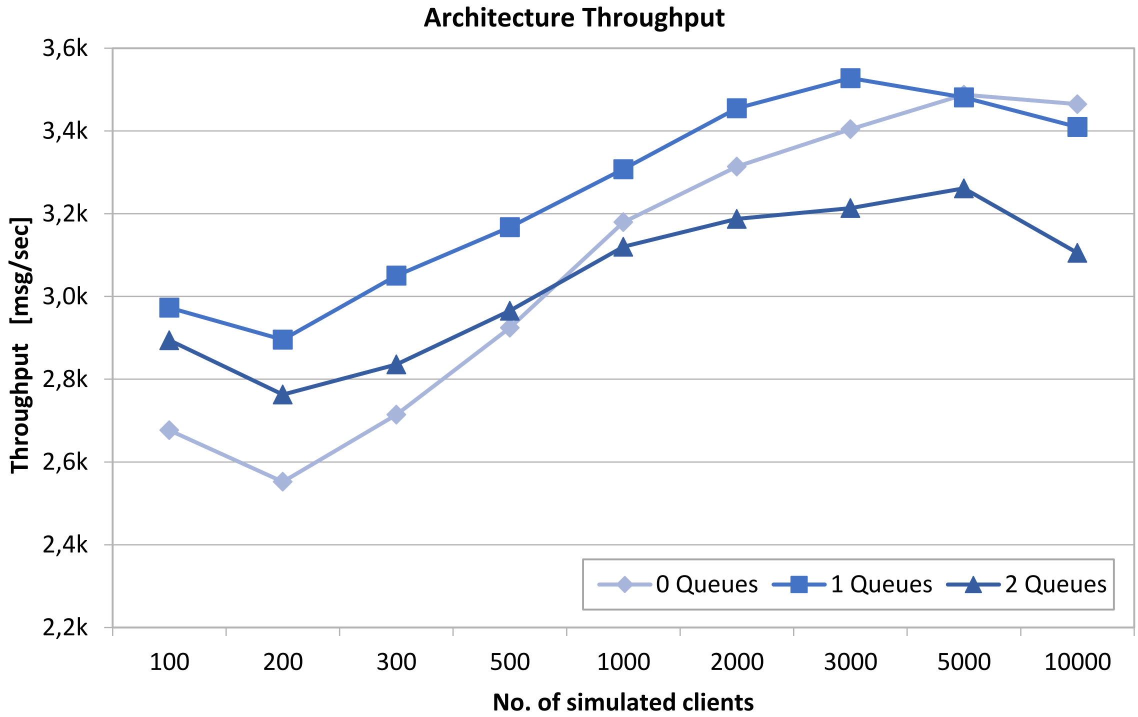

Figure 26 - Load test results of the Architecture

Tests simulated 10 clients with a number of parallel requests, ranging from 1000 to 10000 each. The tests certified the ability of the architecture to receive about 3500 messages per node per seconds. The average size of the messages was approximately 5 KB, since the bulk of the messages exchanged during the scenarios are just section occupation messages, followed by occasional bigger data packets involving modifications to the timetables and to the route setting plans.

The architecture does not un-pack the messages but only routes them, so it was clear that the most important bottleneck of the system, even with huge workloads, was the network capacity. Tests made on a complete loop revealed approximately the same capacity of message handling, this time split between message received and message sent. Even considering an average throughput of 2500 messages per seconds, the architecture is able to process the impressive number of 9 millions messages per hour.

The architecture revealed fairly good scalability: adding nodes proportionally improves the ability to handle more messages.

4.5.5 Conclusions in relation to innovations

The improved decision support handling major perturbation (innovation 4) has realized a step change in terms of Technology Readiness Level. At the beginning of the project the TRL for decision support handling major perturbation was assessed at TRL3 (Analytical and experimental critical function and/or characteristic proof of concept). The expected step was to reach TRL6 (System/subsystem simulation or prototype demonstration in a railway environment).

The state-of-the-art of decision support handling major perturbation considered algorithms for train timetabling rescheduling, algorithms for rolling stock rescheduling, algorithms for crew rescheduling. Most of the papers deal with a single rescheduling phase and few integrate two phases. This shows that active research is initiated that laboratory studies validate the aims of separate elements of the innovation and that not all components are yet integrated. This state-of-the-art confirms the assessment of a TRL3.

The design a framework that consists of a closed loop in which each rescheduling phase is solved by an efficient algorithm to find a good feasible solution and gets feedback from the other phases in order to obtain a good feasible solution for the whole system establish that the modules of theses phases can work together. As set of tests of the framework has been carried out in a laboratory environment, theses framework tests can correspond to an assessment as a TRL4.

The rolling stock and the crew rescheduling should consider the entire country. This large extension of the problem can be considered as a reasonably realistic environment for testing the integration of the modules. Therefore this validation in a laboratory environment led to TRL5.

The laboratory testing of the framework done within HERMES environment that is near a railway environment has been carried out with one scenario. The few number of scenarios tested and the open issues regarding the HERMES logging function are significant obstacles to led to TRL6.

To conclude this analysis of the steps of TRL, it must be noted that the TRL 6 has not been reached.

5 DEMONSTRATOR SYSTEM, SIMULATIONS AND DEMONSTRATIONS

5.1 The demonstrator system

The state-of-the-art of railway GUIs reflect the standard approach in using graphical workstations. They provide a mouse, a keyboard and one or more monitors in which the user can open only one full screen application at the same time.
Starting from a previous industrial HMI implementation, a new user interface has been produced which uses the old controller core. The data input manager has been substituted with a new RailML module and the graphical engine with the better performing, touch-ready, JavaFx library. The use of a light and fast graphic engine allowed the creation of an innovative touch interface, in which a lot of different graphic modules have been collapsed into just two: the Train Graph and the Train Describer. The first one integrates, in just one module, platform allocation, selection path and disruption manager modules in a completely touch environment.
The testing of the new HMI in an existing production environment began with good results. In the next year it can be used in a real central post.

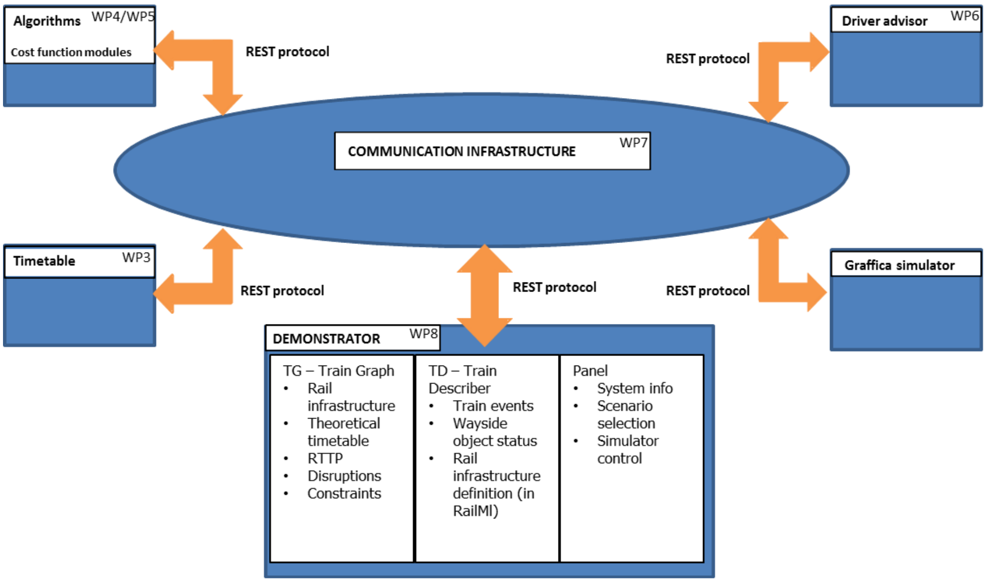

Figure 27 - Demonstrator main schema

Figure 27 shows the demonstrator main schema along with the connection between the full system and the demonstrator itself.

Precondition

The minimum requirement is a workstation (Intel i7 - 24Gb). A Wi-Fi connection must be on to use tablets (optional). Three monitors (46", 46" and 19") must be connected and switched on.

Expected layout

The ON-TIME system shall show:

1) A TD on the first 46" monitor;
2) a Control Panel on the bottom of the first 46" monitor;
3) a TG on the second 46" monitor;
4) two empty KPI half pages on the 19" monitor.

In the following section, the full demonstration path will be shown step-by-step.

Phase 1 – Start up

Scope

Create all the necessary conditions to start the system.

Phase 2 – Select the simulation

Scope

Select the simulation using the proper panel selector.

Precondition

The expected result of the previous phase must be accomplished. The disposition, number and usage of monitors can be arranged.

Expected result

The TD shall show the scenario chosen. No trains are visible yet on TD or TG.

Phase 3 – Select the TG grid

Scope

At least a TG grid must be selected, otherwise no trains will be visible on this interface. This operation is always possible after the completion of Phase 1.

Precondition

TG must be up and running.

Expected result

The TG shall show the trains in the selected grid according to their theoretical timetable.

Phase 4 – Start the simulation (without WP4/WP5)

Scope

The user starts the simulation without WP4/WP5 modules. The simulation speed must be chosen in such a way that the screen results are comprehensible. Remember that, up until now, high speeds can produce stochastic results.

Disruption/delay data shall be applied by a script at the proper time or by means of a TG/TD interface.

Precondition

The simulator is up and running, along with all interfaces.

Expected result

TD shall show the running trains and the signal status. A section block occupation shall be indicated in red. TG shall show a thick line for trains that have passed a station and a thin line according with the related theoretical timetable provided by the simulator.

Phase 5 – Disruption/delay application (without WP4/WP5)

Scope

Apply the disruptions/delays to the running system. This will be done automatically via script.

Precondition

A simulation is running.

Expected result

Disruptions shall be visible on TD/TG. Delays shall be visible on TG only.

Phase 6 – End the simulation (without WP4/WP5)

Scope

The simulation ends automatically when there are no more data to play.

Precondition

A simulation is running.

Expected result

TD and TG shall show the final status. TG results shall be frozen so that they can be compared with the next result. The first KPI half page on the top of the 19” monitor shall be filled by data related to the just ended simulation. The KPI generator shall provide to WP8 the address of a web page (or the path of a task to run) in order to allow the interface to show what is needed.

Phase 7 – Start the simulation (with WP4/WP5)

Scope

The user starts the simulation with WP4/WP5 modules. The simulation speed must be chosen in such a way that the screen results are comprehensible. Remember that, up until now, high speeds can produce stochastic results.

Disruption/delay data shall be applied by a script at the proper time or by means of TG/TD interface.

Precondition

The simulator must be reset completely before it is restarted again.

Expected result

TD shall show the running trains and the signal status. A section block occupation shall be indicated in red. TG shall show a thick line for trains that have passed a station and a thin line according with the related RTTP.

The system shall produce the first RTTP (theoretical timetable) at time zero. Every 3 minutes this data shall be updated.

Phase 8 – Disruptions/delays application (with WP4/WP5)

Scope

Apply the disruptions/delays to the running system. This will be done automatically via script.

Precondition

A simulation is running.

Expected result

Disruptions shall be visible on TD/TG. Delays shall be visible on TG only. WP4/WP5 shall resolve potential conflicts.

Phase 6 – End the simulation (with WP4/WP5)

Scope

The simulation ends automatically when there are no more data to play.

Precondition

A simulation is running.

Expected result

TD and TG shall show the final status. TG results shall be frozen so that they can be compared with the previous result. The second KPI half page on the bottom of the 19" monitor shall be filled by data related to the just ended simulation. The KPI generator shall provide to WP8 the address of a web page (or the path of a task to run) in order to allow the interface to show what is needed.

Screenshots

Some screenshots from previous simulations are set out below.

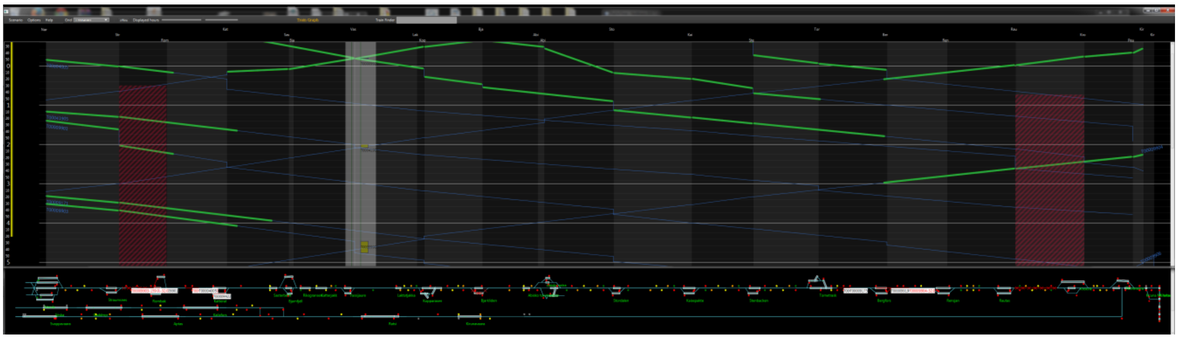

Figure 28 - Kiruna-Narvik with platform occupation in Vassijaure and two disruptions

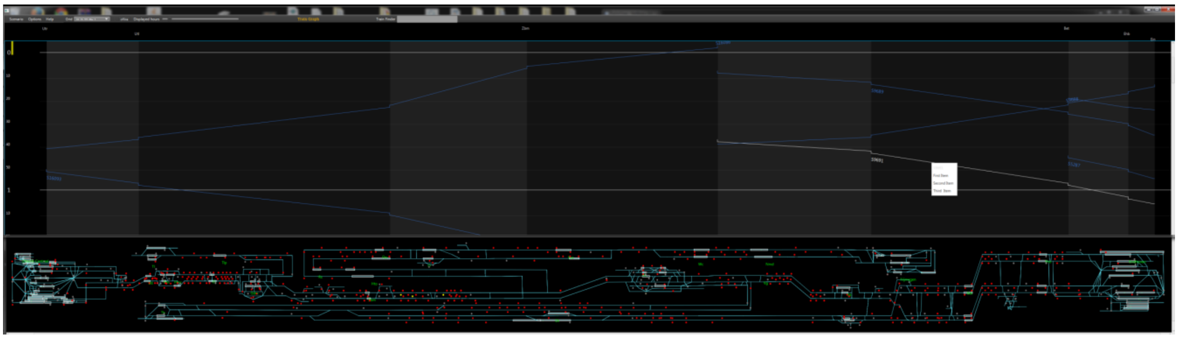

Figure 29 - Netherlands with a train selection

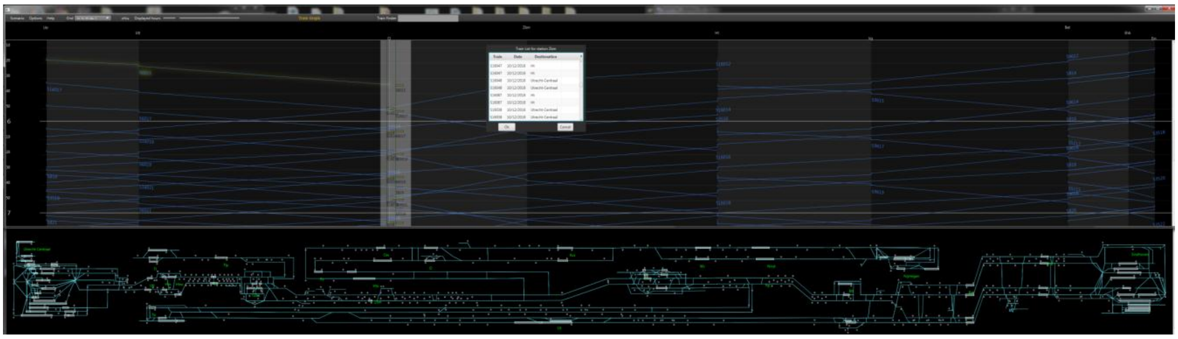

Figure 30 - Netherlands with train list in a station

Requirements matrix

Table 6 sets out a technical evaluation of the system against the requirements written in D8.1. For every named requirement, the percentage of realization is given along with, where relevant, the reason why the requirement has not been satisfied.

Name	%	Note
Performance requirements (1.3)	100%	
Environmental requirements (1.5)	100%	
RailML to TD converter (1.6 – 1.9)	100%	Due to the inefficiency of SVG, we preferred to use JavaFx. A tool to modify the RailML infrastructure data by hand has also been developed.
Get Infrastructure Data (3.1.1.1; 3.2.1.1)	100%	
Get Infrastructure Unavailability (3.1.1.2; 3.2.1.2)	100%	The unavailabilities (disruptions) are sent by the system using a proper event.
Get Timetables (3.1.1.3)	100%	
Get Train Delay Info (3.1.1.4)	100%	Obtained by RTTP
Get Timetable Delay Info (3.1.1.5)	100%	As for the previous requirement, obtained by RTTP
Get Real Time Traffic Plan (3.1.1.6)	80%	To be tested
Train Position Change Event (3.1.1.7; 3.2.1.3)	100%	Not used any more
Train Suppressed Event (3.1.1.8; 3.2.1.4)	0%	
Line Disruption Event (3.1.1.9; 3.2.1.7)	100%	Station, Line and Track disruptions collapsed into a single event (Track disruption)
Track Disruption Event (3.1.1.10; 3.2.1.10)	100%	Station, Line and Track disruptions collapsed into a single event (Track disruption)
Station Disruption Event (3.1.1.11; 3.2.1.8)	100%	Station, Line and Track disruptions collapsed into a single event (Track disruption)
Temporary Speed restriction Event (3.1.1.12)	0%	

Name	%	Note
Connection Conflict Event (3.1.1.13)	0%	
Train Enter Event (3.2.1.5)	100%	Received but not used. Occupation event is used instead.
Train Exit Event (3.2.1.6)	100%	
Platform Disruption Event (3.2.1.9)	100%	Substituted by Track disruption event
Singal State Change Event (3.2.1.11)	100%	
Td Section Occupation Event (3.2.1.12)	100%	
Td Section Release Event (3.2.1.13)	100%	
Set Route Event (3.2.1.14)	0%	This means a section can be occupied but not set.
Unset Route Event (3.2.1.15)	0%	This means a section can be released but not unset.
Interactions (3.1.2; 3.2.2)	0%	The system provides only a reduced interaction (*)
User Request Event (3.3.1.1)	0%	(*)
System Request Event (3.3.1.2)	0%	(*)
Operations (3.3.1.3)	0%	(*)
Changing parameters through HMI (3.3.2.1)	0%	(*)
CF Parameters Change Event (3.3.2.2)	0%	(*)
CF Parameters Config Available Event (3.3.2.3)	0%	(*)
Get Last Parameters Request (3.3.2.4)	0%	(*)
Parameters Response	0%	(*)
Protocols- Data Provider (3.3.3)	0%	(*)

Table 6 - Technical evaluation of D8.1 requirements

5.2 Benchmarking simulations ECML, Iron Ore line and Dutch network

Benchmarking of simulations have been done for East coast main line, Iron Ore line and Utrecht/Arnhem/Eindhoven network in the Netherlands. The results are presented in this chapter.

Using the times recorded in the HERMES log file for departures from station stops and the scheduled departure times in the timetable, a comparison is made between the two values. For each service, the difference in departure time, if any, is calculated for every timetabled departure. The results for the Iron Ore line, the ECML and the Dutch network, the networks for which the quantitative evaluation is carried out, are shown in Figure 31 to Figure 35 and summarised in Table 7.

The graphs show the services in the simulation along the x axis and the stations on the y axis. The services are numbered and the service name to which the number corresponds is listed in the appendix for each of the case study networks. The services are listed in ascending order of their first timetabled stop time.

The markers on the graphs are either:

- A light grey circle with no border, indicating that a departure was scheduled in the timetable, but was not recorded in the simulation;
- A white circle with a black border, which indicates that the timetabled departure time and the simulated departure time were identical;
- A grey circle with black border, whose shade of grey may vary, which indicates that a departure was both scheduled in the timetable and recorded in the simulation. The depth of the shade of grey indicates the severity of the delay: lighter shades for smaller differences, and darker shades for greater differences. The grey shading is on a logarithmic scale and should be read in conjunction with the colourbar of the figures.

The light grey circles with no border, indicating a timetabled stop but not a simulated stop, may be recorded because in the case where a service is timetabled to begin before the simulation time period (e.g. 7 – 10 am for ECML), the train will not enter the simulation. Only those which are timetabled to begin their journey after the start of the simulation will appear. This explains the pattern of the first 24 services in the ECML simulation, and first 10 services in the Dutch network simulation.

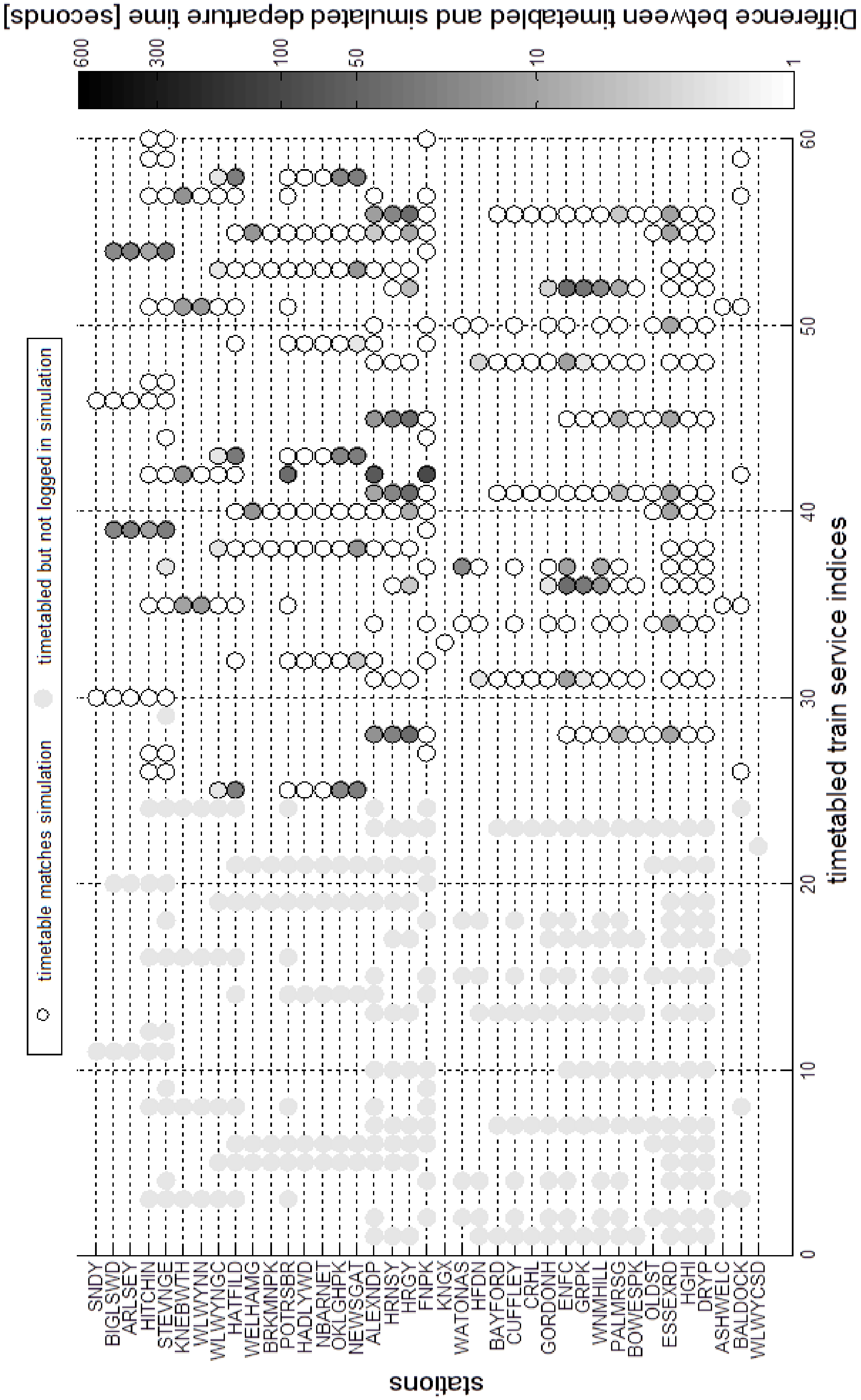

Figure 31 - Comparison of departures between timetable and HERMES baseline scenario for the ECML - Part 1

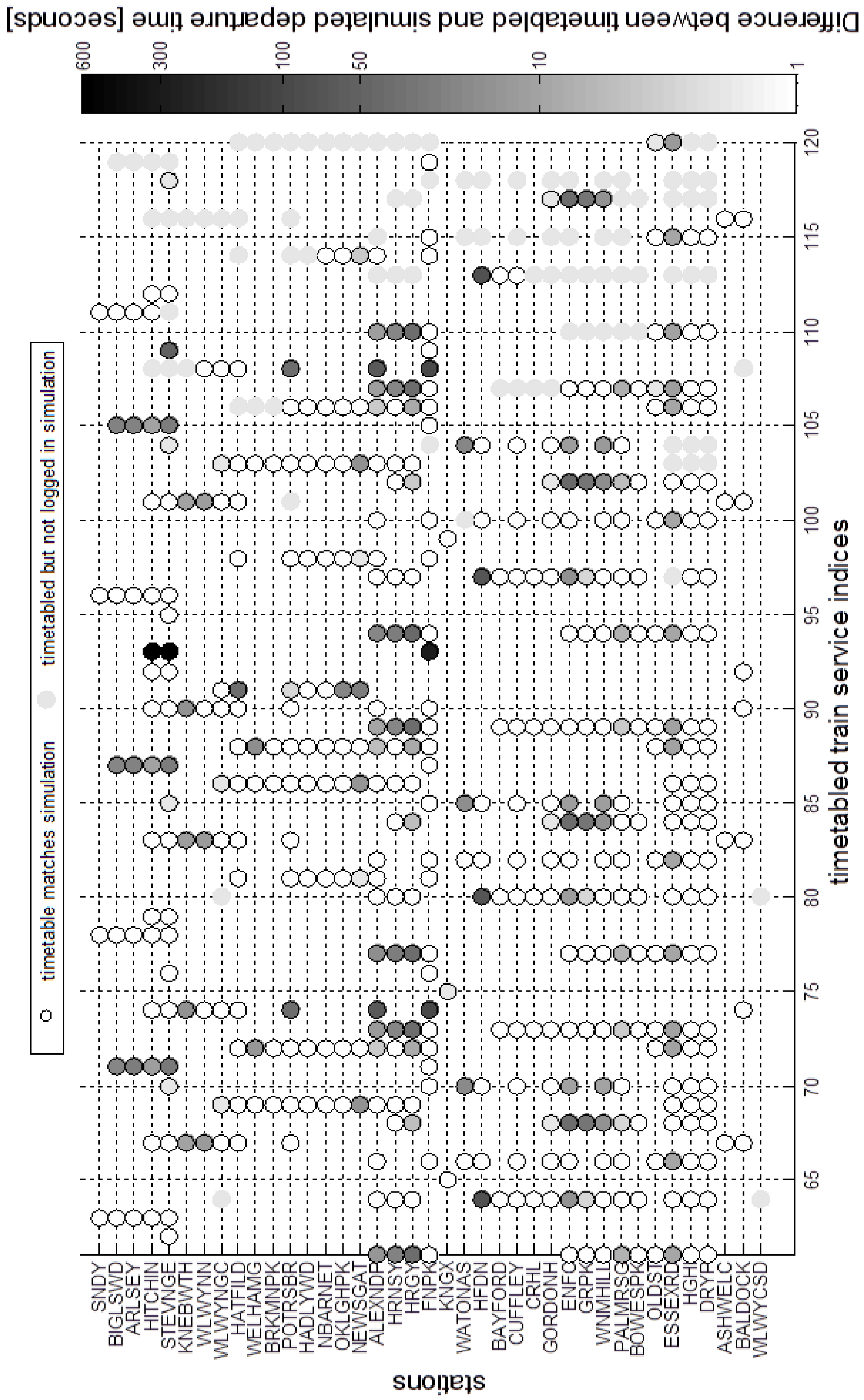

Figure 32 - Comparison of departures between timetable and HERMES baseline scenario for the ECML - Part 2

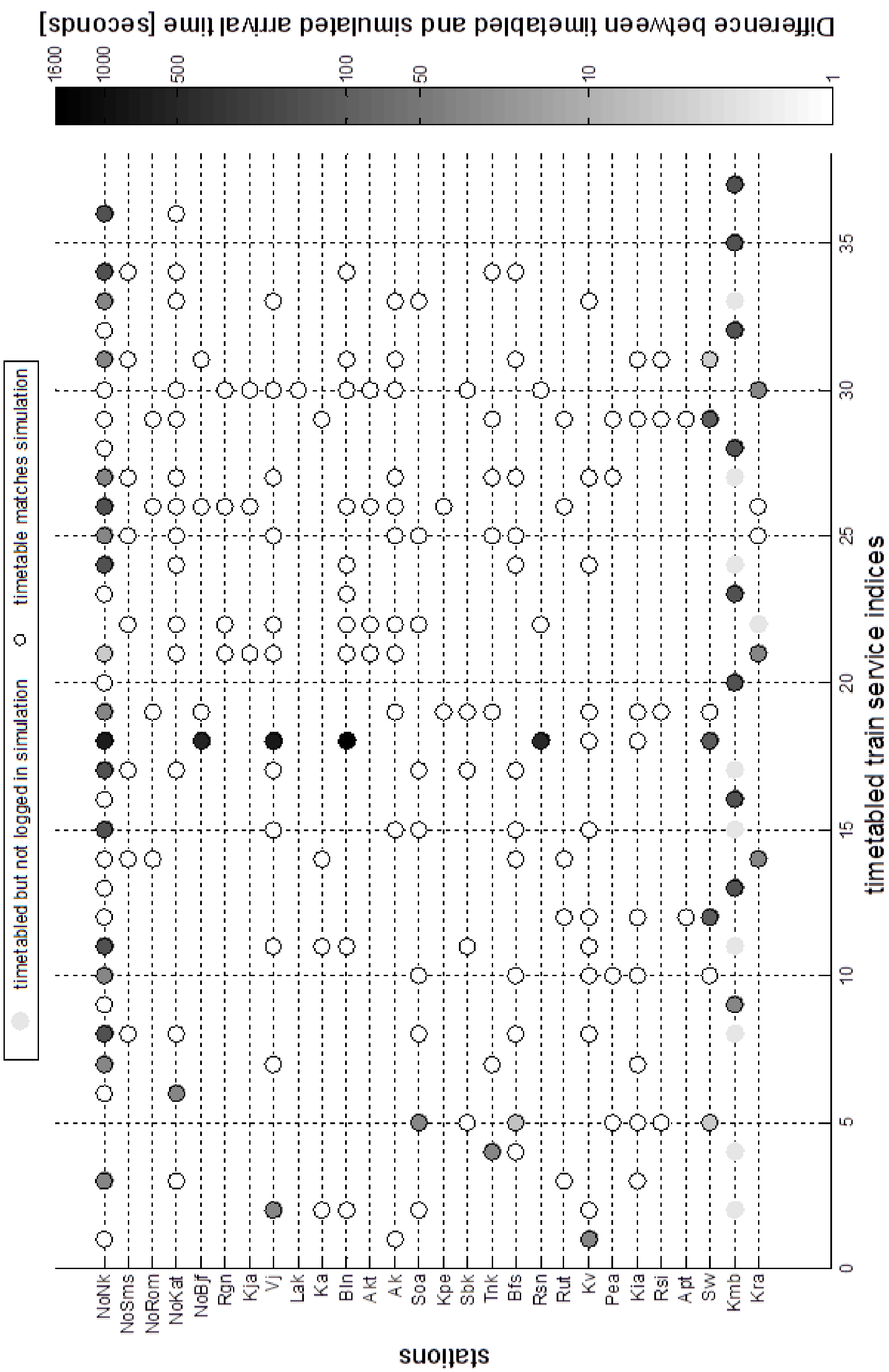

Figure 33 - Comparison of departures between timetable and HERMES baseline scenario for the Iron Ore line

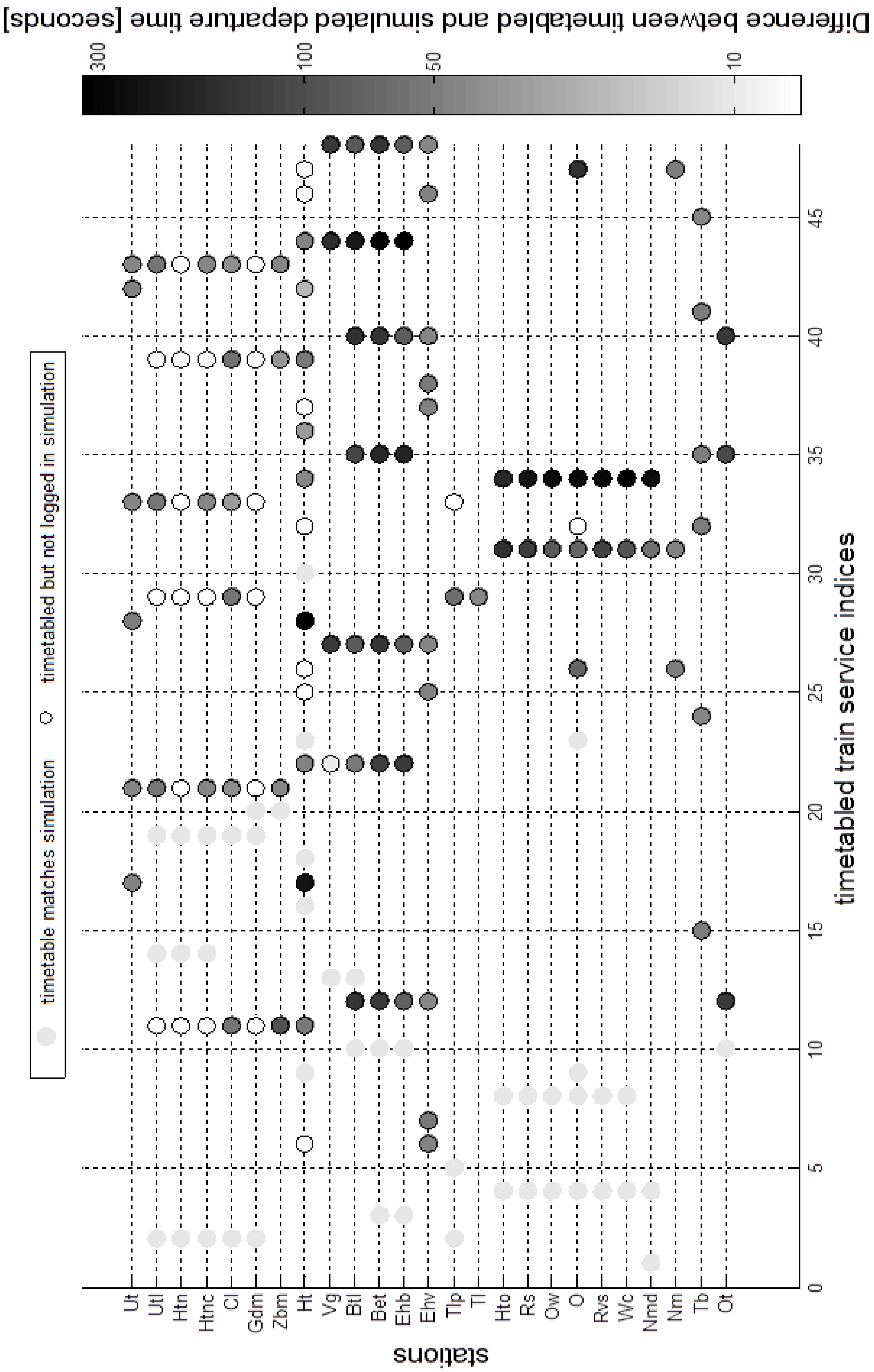
timetable matches simulation
timetabled but not logged in simulation
Difference between timetabled and simulated departure time [seconds]
300
100
50
10
timetabled train service indices
0
5
10
15
20
25
30
35
40
45
stations
Ut
Utl
Htn
Htnc
Cl
Gdm
Zbm
Ht
Vg
Btl
Bet
Ehb
Ehv
Tlp
Tl
Hto
Rs
Ow
O
Rvs
Wc
Nmd
Nm
Tb
Ot

Figure 34 - Comparison of departures between timetable and HERMES baseline scenario for the Dutch network - Part 1

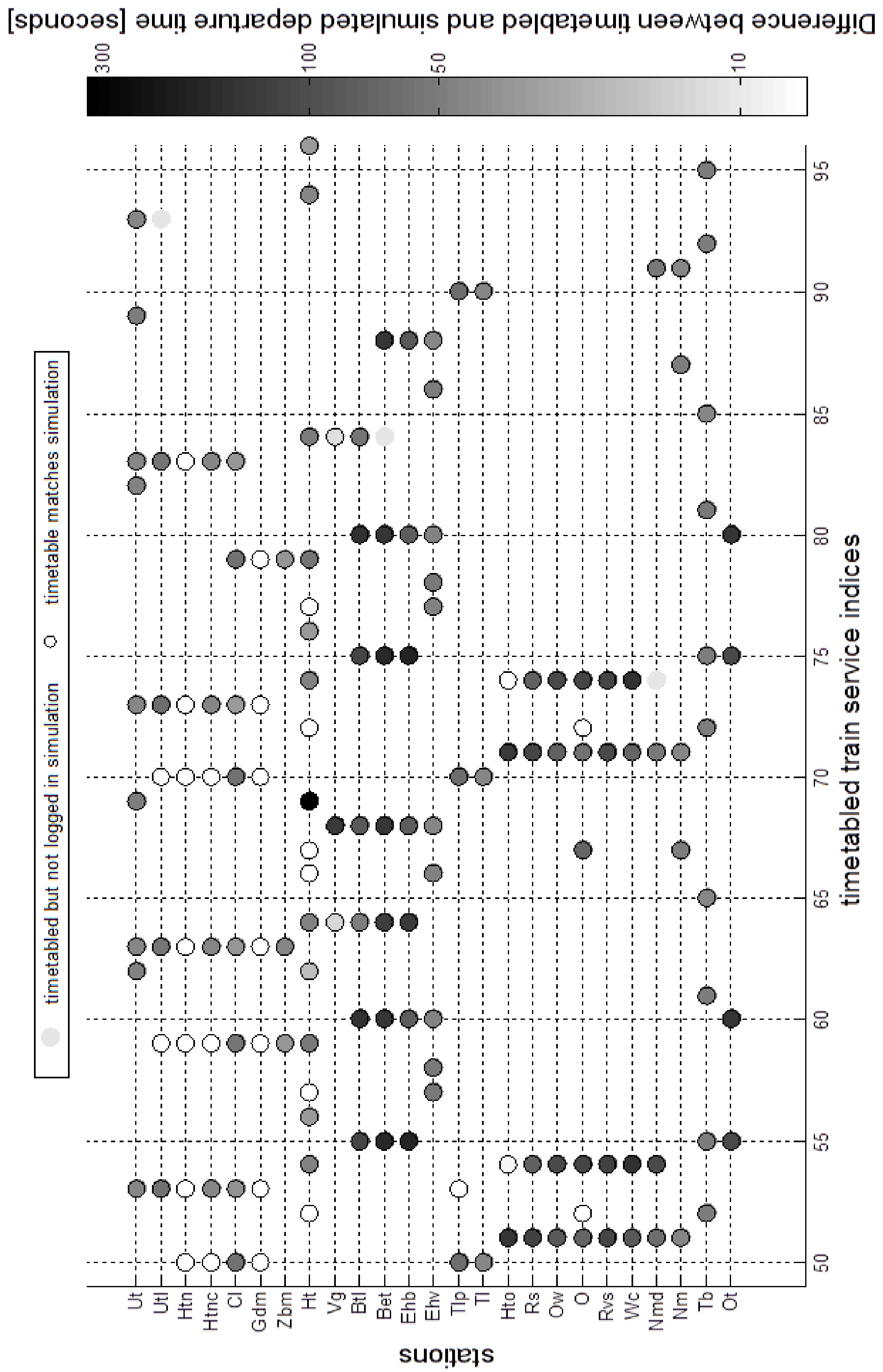
Difference between timetabled and simulated departure time [seconds]
300
100
50
10
timetabled but not logged in simulation
timetable matches simulation
timetabled train service indices
50
55
60
65
70
75
80
85
90
95
stations
Ut
Utl
Htn
Htnc
Cl
Gdm
Zbm
Ht
Vg
Btl
Bet
Ehb
Ehv
Tlp
Tl
Hto
Rs
Ow
O
Rvs
Wc
Nmd
Nm
Tb
Ot

Figure 35 - Comparison of departures between timetable and HERMES baseline scenario for the Dutch network - Part 2

Difference, d [seconds]	ECML		IOL		Dutch	
	Number of occurrences	Percentage of total	Number of occurrences	Percentage of total	Number of occurrences	Percentage of total
0	579	71.7	162	78.3	55	19.6
≤5	54	6.7	3	1.4	0	0.0
5<d≤10	31	3.8	1	0.5	3	1.1
10<d≤30	104	12.9	0	0.0	4	1.4
30<d≤60	27	3.3	17	8.2	118	42.0
60<d≤300	11	1.4	19	9.2	95	33.8
300<d≤1800	2	0.2	5	2.4	6	2.1

Table 7 - Distribution of departure differences between timetable and HERMES simulation

5.3 Demonstration and simulation East coast main line (United Kingdom)

5.3.1 Purpose

Part of the East Coast Main Line (ECML) in the UK and its intersecting routes are used as an example that represents high capacity mixed traffic lines. The geography of and traffic on section of network used for the case study has been described in Section 2.6 above.

Perturbation scenarios were identified in order to evaluate and compare the impact of ON-TIME systems on handling small perturbations. This section is a brief summary of the perturbation scenarios identified and simulated have been discussed in detail in deliverable D 4.3.

The timetable proposed for use in 2018 is simulated for a weekday between 7:00 am and 10:00 am. The scenarios considered apart from the base case with no disruptions are:

- Scenario 1: Entry delay to single train;
- Scenario 2: Multiple trains with entry delays;
- Scenario 3: Conflict at junction.

The KPIs that are considered for this study are : the journey time, punctuality, total delay, resilience, energy consumption and resource usage. The KPIs obtained with the simulator using the ON-TIME algorithms are compared with those using pure timetable order. This provides a like for like comparison of the algorithms.

The East Coast Main Line is evaluated for the first two perturbed scenarios considering an observation period of 2 hours. In particular the conflict detection and resolution algorithms control the traffic in real-time within a rolling horizon framework. The parameters of the rolling horizon like rescheduling interval and prediction horizon can vary according to the algorithm. For a given algorithm these parameters are anyway the same for all the disturbed scenarios. Specifically for the ROMA algorithm we adopted a rescheduling interval of 2 minutes and a prediction horizon of 30 minutes. This means that each two minutes an updated real-time traffic plan is computed by ROMA, establishing the control measures (train orders, routes and departure/arrival times) to be applied in the next 30 minutes. The average computation time required by ROMA to compute the RTTP over the next 30 minutes is around 12 s. The size of the rolling horizon used for RECIFE is 30 minutes and its average computation time is 10 seconds.

5.3.1.1 Scenario 1: Train S488 has an entrance delay of 10 minutes in London Kings Cross

<u>Performance of the RECIFE algorithm</u>

The results show that the control strategy proposed by RECIFE allows eliminating delays almost completely. Remark that King's Cross station (the departing point of train S488) is not considered in these results. Hence, the primary delay of 600 seconds of train 488 is not detectable here.

The Time to recover is the only negatively impacted resilience KPI's (-7%). The Deviation area improves by 58% and the **Maximum delay improves by 55%**. This worsening is possible despite the optimization since the objective function used by RECIFE is the minimization of the total delay. The great majority of the trains with a non-zero deviation get to stations earlier than in the baseline simulation (they are represented by curves in the deviation negative subspace). The train with the largest deviation is 488, which suffers the primary delay and never recovers.

The journey time, connectivity, energy consumption and resource usage are not impacted by the traffic control strategy.

<u>Performance of the ROMA algorithm</u>

The average train journey time when applying ROMA practically equals the one obtained within undisturbed traffic conditions. The positive impact on traffic given by the usage of the algorithm is immediately visible when looking at the departure delays at stations. It is observed ROMA is able to remove completely the departure delays for all the stations. In general the total departure delay over all the network is incredibly reduced by more than 99% from 966 s to 8 s.

Of course this result is also observable from the resilience KPI reported in the corresponding resilience table and graphically in deliverable D4.3. In this case study ROMA is able to reduce the deviation area by 38% and the maximum delay by 33%, while the time to recover increases very slightly. When using the algorithm the energy consumption remains practically unchanged with respect to the undisturbed scenario. From the resource usage perspective for some signals, the control strategy implemented by ROMA increases the average number of trains per hour passing the signal (see the track usage table). This implicitly means a higher capacity for those railway corridors with respect to the case in which trains follow the timetable order. The rolling stock usage instead remains constant independent of from the control strategy used.

5.3.1.2 Scenario 2: Multiple trains have entrance delays throughout the whole network

Performance of the RECIFE algorithm

In this scenario, implementing the control strategy proposed by **RECIFE improves over the timetable order by 16% in terms of total delay** and of 9% and 29% in terms of the resilience KPI's Deviation area and Maximum delay. The Time to recover instead increases by 7%. In fact, when implementing the control strategy proposed by RECIFE, a non-negligible deviation remains until the end of the time horizon considered. However, the deviations occurring in the last half an hour are mostly negative: some trains get to some stations earlier than in the baseline simulation

The journey time, connectivity, energy consumption and resource usage are not impacted by the traffic control strategy.

Performance of the ROMA algorithm

By using ROMA the average journey time of trains is practically kept very close to the one relative to the undisturbed condition. It is observed that ROMA clearly reduces the departure delay at stations. For this specific scenario the departure delay at Hitchin is strongly reduced, while in the other reported stations the algorithm gives more or less the same delay observed in the case where trains follow the timetable order during the perturbed scenario. This happens because the delay at this station coincides with the train entrance delay set by the scenario itself. Logically the algorithm cannot reduce the departure delays at these stations below the entrance delay set as input to the scenario. In general, the total departure delay over all the stations is reduced by 12%. In this case however the Punctuality KPI is not able to clearly explain the benefit gained by using the algorithm to control traffic. The resilience KPI is instead more explicative and shows how positive is the application of the algorithm. The ROMA algorithm is able to considerably reduce the deviation area by 27% and the maximum delay by 35%, while keeping the recovery time practically the same as the case where trains follow the timetable order. In terms of energy consumption it can be seen that the total energy consumed by the trains is practically the same independently from the control strategy applied (i.e. the timetable order or the one given by ROMA). The resource usage KPI (track and rolling stock usage) instead remains unchanged independently from the kind of control strategy used to control traffic.

5.3.1.3 Scenario 3: Junction conflict - Train S537 has an entrance delay entering the mainline north of Kings Cross

Performance of the DEJRM algorithm

The resource usage is unaffected by the running of the algorithm, because all trains are routed through on the same paths. The resilience and punctuality is improved due to the DEJRM algorithm routing the trains in a different order at the junction.

In the reference scenario, a freight train is blocking a passenger train S537 that is delayed on its way to Finsbury Park and another passenger train S574 is also delayed as it is following the first train. Train S537 reaches Finsbury Park 10 minutes and 33 seconds late and further knock-on delays are incurred by services that follow into Finsbury Park. As a result of the rescheduling, the first passenger train is 6 minutes and 13 seconds late at Finsbury Park, due to the primary delays caused to it by the signal failure disruption. The knock-on delay occurring to train S574 is reduced significantly and the knock-on delay to subsequent trains running on the line is improved. The freight train is not significantly delayed and continues on its journey on time.

A 74% decrease in total departure delay is seen as a result of the rescheduling made by the DEJRM algorithm as well as a significant improvement in punctuality

5.3.2 Conclusions

The results are extremely encouraging.

While the figures are excellent, these are limited tests and will need to be validated and calibrated before use.

However, this activity proves that a simulation platform with open interfaces will provide the ability to evaluate and compare the performance of systems offered for traffic management.

5.4 Demonstration and simulation Iron Ore line (Sweden/Norway)

5.4.1 Purpose

The Iron Ore Line (IOL) demonstrator illustrates the traffic on a single track line with a border crossing. The purpose is to evaluate ON-TIME systems for optimal re-planning, in case of minor perturbations. The traffic between Kiruna in Sweden and Narvik in Norway is simulated. The line has shortly been described in section 2.6 above.

Some perturbation scenarios for the Iron Ore Line have been identified and used for simulations. The scenarios have been specified based on an investigation of most common perturbations in real traffic situations. The main scenarios are:

- One fully loaded iron ore train delayed from original station;
- Extra train added, on short notice;
- Long distance freight train, entering the iron ore line, delayed;
- Speed restriction due to maintenance work between two stations;
- Infrastructure problem. Point out of order at one station.

The results concerning how the re-planning algorithms, developed in the ON-TIME project, WP4, can solve problems in connection with traffic perturbations are especially interesting. Such algorithms will in the future be a part of new traffic control systems in Sweden.

5.4.2 Delimitations

In the demonstrator for the IOL, the interactive HMI for the traffic controller is not implemented. However, in the STEG implementation in Boden, a fully interactive system is in full operation. Future research will combine these two innovations: the fully interactive system and the supportive optimization algorithms (perturbation management module, PMM) developed in ON-TIME.

Not all specified scenarios have been evaluated in the simulator and demonstrator system. Two typical scenarios have been selected for the evaluation studies:

Scenario A: One iron ore train delayed.

- Train 9904 delayed 40 minutes, from its start in Peuravaara (PEA).

Scenario B: Speed restrictions between two stations.

- Speed restriction to 20 km/h between Rensjön (RSN) and Berfors (BFS), for the whole day.

5.4.3 The simulator structure

For the evaluation studies, the focus is on the operational re-planning performed by the PMM, and the connections to the DAS and train drivers have not been included. The HERMES simulator simulates the infrastructure and the traffic control system, see Figure 36.

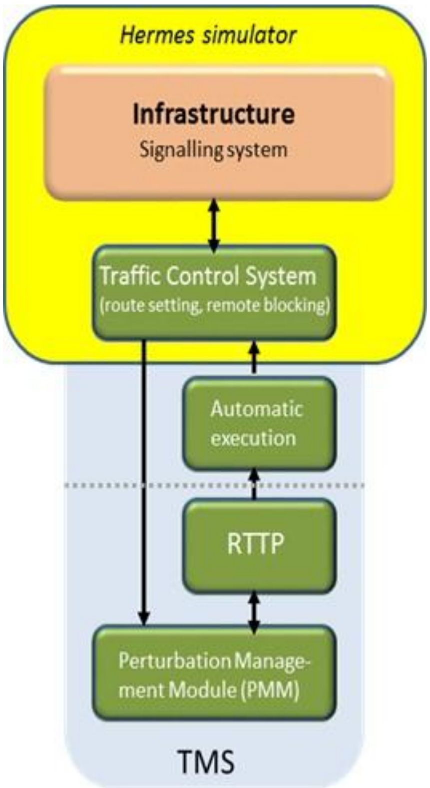

Figure 36 – HERMES simulator

The HERMES simulator simulates the infrastructure, the traffic control system and the train traffic for the specified scenarios. In the evaluation studies the driver advisory systems and the interactive connections to the traffic controllers have been omitted.

5.4.4 Evaluation method

The evaluation consists of several different parts.

1. Evaluation of the simulator system, so that the traffic on the Iron Ore Line is correctly executed. This evaluation of the simulator system as such has mainly been performed in other parts of the ON-TIME project. Here some additional comments are given.
2. Quantitative evaluation of the simulated scenarios, using the Matlab evaluation tool developed in the ON-TIME project.
3. Qualitative evaluation of the new real time traffic plan (RTTP) generated by the PMM. Here the new solutions are evaluated, based on a qualitative analysis of time-distance graphs.

5.4.4.1 Quantitative evaluation

The Matlab tool uses log-files generated by HERMES for calculation of quantitative measures for specified key performance indicators (i.e. journey time, resilience, punctuality, energy consumption, resource usage, etc.). The result can also be visualized in form of a number of diagrams. See examples below.

5.4.4.2 Qualitative evaluation

This is based on analysis of the time-distance graphs. The re-planning performed and the simulations based on the execution of the new RTTP can be studied. In a first step the analysis and evaluation are made by the researchers. In a later step the quality of the re-planning decisions will also be evaluated by experienced traffic controller from the TCC in Boden. A preliminary study, where a human controller was asked to evaluate the re-planning performed by the PMM was made.

5.4.5 Results

The results from the simulations, and the evaluation of the re-planning performed by the PMM, are described in document D8.4 from the ON-TIME project. For studying perturbations, two different scenarios have been used for the evaluation simulations:

5.4.5.1 Scenario A

Train 9904 delayed 40 minutes, from its start in Peuravaara (PEA). For this scenario three different simulations have been made:

- Without PMM;
- With the PMM model ROMA;
- With the model RECIFE.

5.4.5.2 Scenario B

Speed restriction to 20 km/h between Rensjön (RSN) and Berfors (BFS), for the whole day.

Also for this scenario, three different simulations have been made:

- Without PMM;
- With the PMM model ROMA;
- With the model RECIFE.

The ROMA and RECIFE models and algorithms are described in documents from ON-TIME WP4, see deliverables D4.1 and D4.2.

5.4.6 Analysis of the simulations

The evaluation has mainly been qualitative, i.e. we have analysed the quality and the relevance for the Iron Ore Line of the operational re-planning performed by the PMM. The detailed analysis can be found in the document D8.4. The analysis shows that the studied PMM modules can solve the tested scenario perturbations. However, for the IOL the PMM performance must be improved in some important aspects:

- The generated headways are sometimes too short, especially for the loaded iron ore trains;
- All time-table stops are considered as fixed, which is not correct for the IOL. Most stops for iron ore and freight trains are only for meetings, and if a meeting is cancelled, the stop should be removed from the RTTP;
- The priorities between trains must be modelled and implemented. On the IOL it is e.g. important to give priority to loaded iron ore trains. Otherwise they are forced to extra stops, will be further delayed and consume more energy.

As an example, the following figure illustrates the result of a simulation, for scenario A, where the baseline simulation (dashed lines) is compared to the simulation with the ROMA PMM active (solid lines). The problem with unnecessary stops not being removed is illustrated in Figure 37.

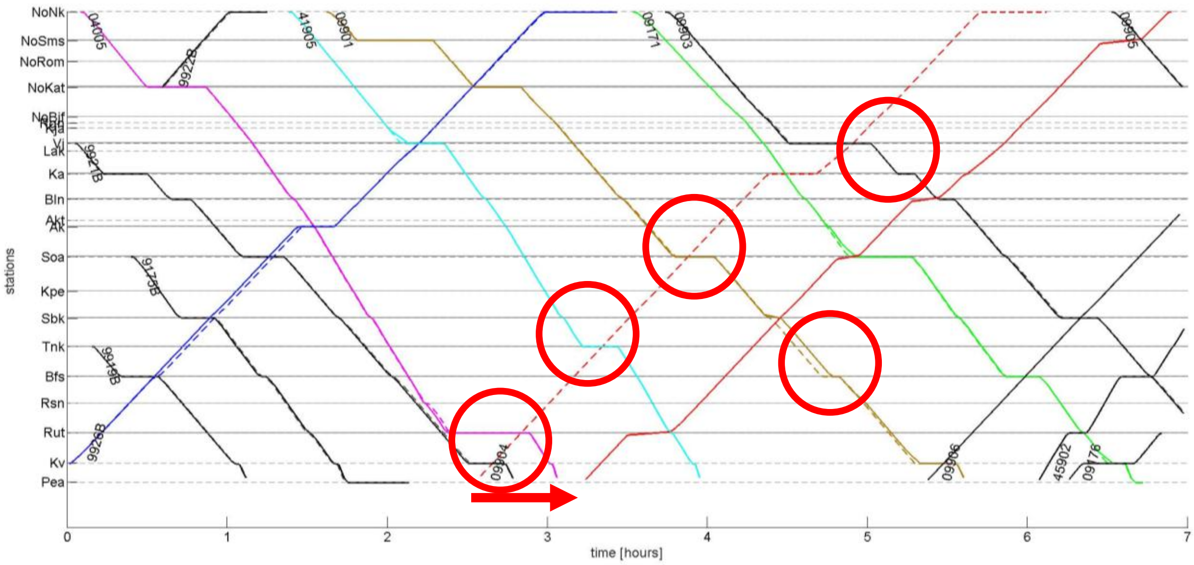

Figure 37 - Unnecessary stops are not eliminated by the PMM

5.4.7 Conclusions

From the qualitative evaluations we can conclude that:

- The HERMES simulator can, with a number of limitations, simulate the traffic on the IOL, for undisturbed traffic as well as for traffic with certain perturbations;
- The developed systems for automatic re-planning, the PMM modules, are able to handle the perturbations specified in some scenario for the IOL;

- That evaluations show that a number of additional requirements must be fulfilled, if the systems are going to be used in real traffic control on the IOL;
- That the results give us a good basis for future research and development;
- One additional problem encountered is that the original timetable contains errors that should be re-moved prior to simulations and evaluations.

For Swedish railways, the results from the ON-TIME project in general, and from the experiments on the IOL in particular, are of great interest and importance. The TCC in Boden, today controlling the IOL using the control system STEG for operational re-planning and control, and the DAS system CATO for train drivers, will be a platform for future development and evaluations, based on the ON-TIME results.

Even if we now see many limitations in the developed systems, e.g. in the PMM modules, these can probably be eliminated in coming development.

Finally, the most important future development will be to integrate the PMM modules in a fully interactive environment. The present system in Boden, with STEG and CATO, can profit from efficient systems for optimal re-planning and decision support. The human controllers' tasks must be coordinated with the more automated functions and their user interfaces must visualize important aspects of the PMM actions. The controllers must also have the possibilities to specify conditions for the automatic modules, so that dynamic requirements can be specified, e.g. regarding priorities, train order, track usage etc.

5.5 Demonstration and simulation Bologna node (Italy)

The use of the Bologna node demonstrates a different aspect of the ON-TIME objectives and results. The focus was on the use of the data architecture to improve the use of data.

The signalling plans of the Bologna node were acquired as cad drawings and related data. These were transformed into the extended RailML format used by the project through the use of a data converter tool. The data created was fed into the simulator to generate the infrastructure used for the simulations and seamlessly into the demonstrator. The detailed results are presented in the deliverable D8.3.

The ability to use data from existing/legacy infrastructure systems in emerging traffic management simulators and operations tools such as train describers/train graphs is one that will provide numerous advantages to the railway.

For the demonstration on the Italian network we selected Bologna node, a very complex rail infrastructure area, where major problems for capacity estimation and traffic management occur, involving large stations, yards and important traffic flows. Moreover railway nodes pose difficult problems in predicting real time propagation of delays and rearranging traffics during perturbations.

Today stochastic optimisation experiments show that standard fixed percentages of running buffer times do not contribute effectively to robustness of train services and the supplements should be varied in size and space according to the specific characteristics of the infrastructure, rolling stock, timetable and expected traffic. In addition the use of reliable simulators require as input a large amount of data related to the infrastructure characteristics.

The first step of our demonstration focused on a new system of uploading infrastructure data into the simulator, through the common standard Rail ML adopted by the project: this in order to reduce the manual and heavy data inputs, as it is today, and realizing a new efficient interface via a process reengineering method.

Following this approach new generic on-line decision support systems could easily be implemented and distributed. In this regard the first activity was to draw the network using a commercial tool (CAD), tailored with railway objects, able to simplify the acquisition of railway infrastructure data and provide them in automatic way to client systems, like simulators.

The other activity was the development of an optimization scheduling algorithm, under-taken by the University of Bologna,as a means of rescheduling rail traffic in case of conflicts, perturbations and major disturbances.

The Conflict Detection and Resolution (CDR) algorithm is a macroscopic rescheduling algorithm, which determines feasible macroscopic timetables after some delays/perturbations have occurred.

The general vision is to set up a general scheduling solution to feed finer tools such as simulators, playing as "validators" or introducing second-order tuning of train running.

It considers a suitable description of the railway infrastructure, which consists of locations (either stations or junctions) and tracks connecting them. It considers a discretization of time in 30 seconds, although a finer discretization is possible. When a delay occurs, CDR checks whether some conflicts between trains have happened. In this case, it starts a rescheduling module which reschedules the trains, with the goal of determining a real-time traffic plan (RTTP) such that the total delay is minimized, in other words a timetable that is "as close as possible" to the planned one (i.e. the one before the delay had occurred).

A train shift refers to a train departing later than planned from its initial location. A train stretch is to hold (delay) the departure of a train at any of its intermediate stations.

The algorithm takes as input the following data:

- essential infrastructure description;
- capacity of the stations;
- headway times for tracks and junctions;
- planned timetable.

In addition, a maximum allowed shift and maximum allowed stretch, for each train, may also be specified. The latter ones are used to limit the total amount of delay of each train.

The following constraints are taken into account:

- for each track, a headway time is specified which has to be maintained by any two trains consecutively using a track;
- for each junction a headway time (also called junction occupation time) and a list of conflicting itineraries are specified;

- overtaking between trains is not allowed between two consecutive locations (it is only allowed at stations);
- the capacity of each station, given as a number of tracks;
- the minimum travel and dwell times must be respected.

The algorithm can make the following changes to the planned timetable, in order to determine a feasible timetable: it can shift a train, stretch a train at a station, change from the planned track to a parallel track if one is available, or cancel a train (if the delay becomes larger than the maximum allowed shift/stretch). The objective is to minimise a weighted sum of the total shift and stretch over all trains, i.e. to determine feasible RTTPs that are as close as possible to the planned one. The algorithm is iterative and, at each iteration, reschedules the trains in a predetermined order (different orders of the trains can provide different RTTPs). The best timetable for the current train in the given order is computed by a dynamic programming approach. The best obtained timetable is given on output. The application takes as input the described data in RailML or in the format of RFI data.
We performed two sets of tests, as discussed below.
In both cases we considered a time period between 9:00 and 11:00, with 78 trains travelling through the node, out of which 16 are high-speed trains. The computing time for each run lasts a few seconds on a standard laptop.

5.5.1 First test: delayed trains

The first test takes into account departure delays of trains, which are generated according to the real-world delay data. Every time a train is delayed, CDR is executed and the conflicts, if any, are resolved, producing a macroscopic feasible timetable. We report the additional delay accumulated by each delayed train at its final station, in order to obtain a feasible timetable.

In the following, we report the results of three scenarios (out of 23 experimented) for which the primary delay affecting a train causes additional delays to the same or to other trains. In particular, we report the code of the train that was delayed, the time at which the delay happened, the number of minutes of delay and the location at which the delay occurred. The CDR algorithm is executed and the resulting timetable is analyzed in order to evaluated the additional delay accumulated by each train. We report, this additional delay at the final station, expressed in minutes, only for those trains for which it is positive and in brackets indicate whether the train is a high-speed (HS) train.

- Scenario 1: train 09465 (HS) delayed at 9:43:00 by 6 minutes at Bologna C.le/AV

Train	Delay
09915	1:00

- Scenario 2: train 02230 delayed at 10:20:00 by 6 minutes at Bologna Centrale

Train	Delay
02229	0:30
06416	0:30

- Scenario 3: train 06428 delayed at 10:57:00 by 5 minutes at PM Mirandola Ozzano

Train	Delay
09810	1:00

5.5.2 Second test: high-speed line blockage

The second set of tests considers the blockage of the high-speed line in the main station of Bologna Centrale for a given time period either in one direction or in the opposite one, or in both directions. Therefore, the high-speed trains must be re-routed and travel along the conventional line. Also in this case we report the additional delay accumulated by each train at its final station, in order to obtain a feasible timetable.

In the following, we report three scenarios corresponding to the blockage in Bologna Centrale of either one direction of the high-speed line, the opposite direction or both directions, For each case, the CDR algorithm is executed and we report the number of trains that need to be rerouted along the normal line and the delays, expressed in minutes, accumulated by trains in order to obtain a feasible timetable.

In the various scenarios are re-rerouted between 7 and 16 trains; the ones with delays more than 1 minute are reported below.

- Scenario A: blockage of the even (south-north) direction of high-speed line. Trains need to be rerouted between Bivio Emilia and Lavino/PM Anzola. 7 train have been rerouted.

Train	Delay
06486	1:30
09462 (HS)	2:00
09508 (HS)	1:30
09566 (HS)	1:30
09614 (HS)	1:30

- Scenario B: blockage of the odd (north-south) direction of the high-speed line in Bologna Centrale. Trains need to be rerouted between PM Anzola/Lavino and Bivio Emilia. 9 trains have been rerouted.

Train	Delay
09513 (HS)	3:30

- Scenario C: blockage of the high-speed lines of Bologna Centrale (both directions). Trains need to be rerouted between Bivio Emilia/PM Anzola and PM Anzola/Bivio Emilia, respectively. 16 trains have been rerouted

Train	Delay
06486	1:30
09462 (HS)	2:00

09508 (HS)	1:30
09513 (HS)	3:30
09566 (HS)	1:30
09614 (HS)	1:30

5.5.3 Analysis of the results

We can conclude from the obtained results that the developed algorithm is effective in re-scheduling trains at a macroscopic level, both when delays occur (e.g. generated by diverse causes) and when part of the infrastructure is blocked. As it can be understood, the latter is a more impacting event on traffic conditions and regularity performance, putting at "stress" the railway node operations. In many of the considered cases, the delays can be feasibly absorbed or kept to tolerable levels. Even if one part of the infrastructure is blocked no train is cancelled and the delays are ½ minute and 3 minutes at most. Although these results could be subject to deeper analysis due to other interfering causes (e.g. perturbed passenger flows) they look like a suitable platform to exploit and integrate such a tool in industrial real-time control systems.

5.6 Demonstration and simulation Utrecht/Arnhem/Eindhoven network (Netherlands)

For the quantitative evaluation, the ON-TIME timetabling module is applied to a Dutch case study consisting of a central part of the railway network in the Netherlands. It consists of the railway network bounded by the four main stations Utrecht (Ut) in the North, Eindhoven (Ehv) in the South, Tilburg (Tb) in the West, and Nijmegen (Nm) in the East, with a fifth main station 's-Hertogenbosch (Ht) in the middle and 20 additional smaller stations and stops, see Figure 9. Four corridors connect Ht to the other main stations.

The case study considers the timetable for a workday in 2011 between 7:00 AM and 9:00 AM. There are 36 running trains per hour from eight train passenger lines in both directions, plus freight trains.

The evaluation of the ON-TIME Conflict Detection and Resolution algorithms is performed versus three different perturbed traffic scenarios. All the perturbed scenarios refer to the morning peak period between 7:00 and 9:00 AM. Specifically:

5.6.1 Scenario 0 (Baseline scenario)

All trains run according to the schedule, in absence of any kind of perturbation, disruption, or any other infrastructure problem. Freight trains are not considered.

5.6.2 Scenario 1 (A single delayed IC):

The IC 821 enters Utrecht station with10 minutes delay. This means that also its departure from Utrecht, originally scheduled at 7:07 AM is affected by this delay. There are not freight trains running nor other perturbations or infrastructure problems.

5.6.3 Scenario 2 (Multiple delayed trains)

Several trains enters Utrecht station with a Weibull distributed delay having mean 90 s and standard deviation 125 s. No freight train is considered, nor other perturbations or infrastructure failure.

5.6.4 Scenario 3 (Speed restriction)

Between the stations Houten Castellum and Culemborg the speed limit is decreased from 130 km/h to 40 km/h because of infrastructure maintenance works. The speed restriction is applied for the whole period of analysis. No freight trains are considered, nor other perturbations or infrastructure failures.

6 IMPLEMENTATION INTO PRACTICE

The implementation of ON-TIME results into practice from an organisation and human factors perspective is described in this chapter.

In Chapter 7 implementation, technology readiness levels and future tasks for each innovation are described.

6.1 Implementation improved methods and algorithms

The ON-TIME project results have resulted in the implementation of improved methods and algorithms for:

- Timetable planning (WP3);
- Decision support handling minor perturbations (WP4);
- Improved decision support – handling major perturbations (WP5);
- Centrally guided train operation (WP6).

In addition to the algorithms, key developments in data architectures, simulation capabilities and evaluation criterion have allowed the benefits of algorithms to be compared effectively for the first time.

6.1.1 Data architectures

At the project outset it was identified that there was a need to research, validate and standardise approaches for information exchange over a common information architecture. The work undertaken in this area within the project has allowed different processes and algorithms to be integrated to be tested on real-world scenarios. The approach developed has moved knowledge forward to begin to, for the first time, address the requirements set out in EU regulation 913/2010.

Within the project the data architecture has been used to integrate all of the innovations in the project in a common and extensible manner, as shown in Figure 38. The approach developed is in need of further testing and verification, but following this further work could quickly be moved to a standardised solution that could be used by many parties to integrate data between train control systems, traffic management systems, driver advisory, etc. in a dynamic manner.

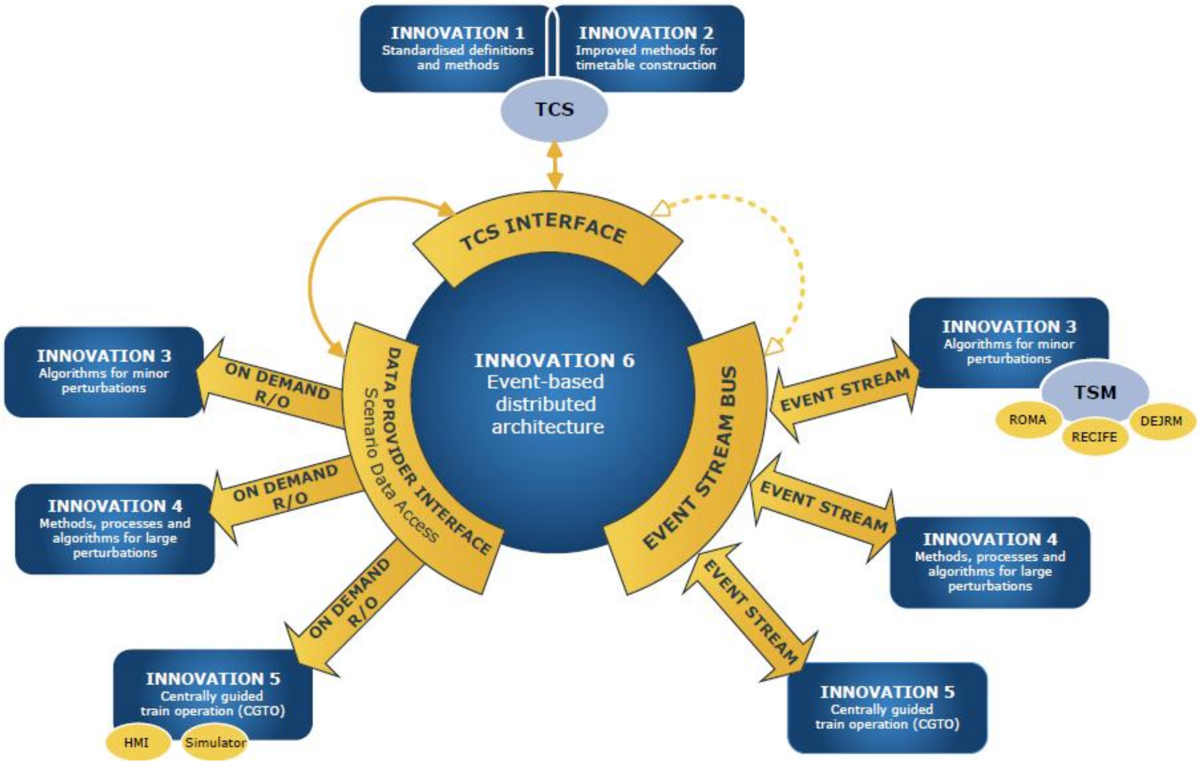

Figure 38 - Graphical representation of the ON-TIME data dictionary

6.1.2 Simulation capabilities

At the outset of the project it was identified that existing capabilities needed to be improved in order to offer fast on-line decision support for the impact evaluation of suitable dispatching measures and the selection of the most efficient one. The project adopted HERMES as the simulation it would use, and this was further developed to fit to a novel architecture that would allow the real-time interaction of the simulator with external process, such as algorithms. The development of this architecture, which is shown at a high level in Figure 39, marks a step-change in how simulators can be used as part of the real-time decision making and evaluation process. The ability to 'subscribe' to variables within the simulation through an open, standardised API (Application Programming Interface) means that: (i) algorithms can be directly linked to simulation; (ii) simulators can be used in real-time as part of the decision making process; (iii) different algorithms can be directly compared using identical simulation runs – all of which are key and powerful functions required to deliver the next generation of intelligent railway operations systems.

In the final stages of the project API has been migrated to be used with an emulation of a real traffic management system developed by Ansaldo. Although there is still significant work to undertake in realising a commercial solution, the work in ON-TIME has shown the benefits of such an approach, and the need to increase (and validate) simulation capabilities in order to realise improved solutions.

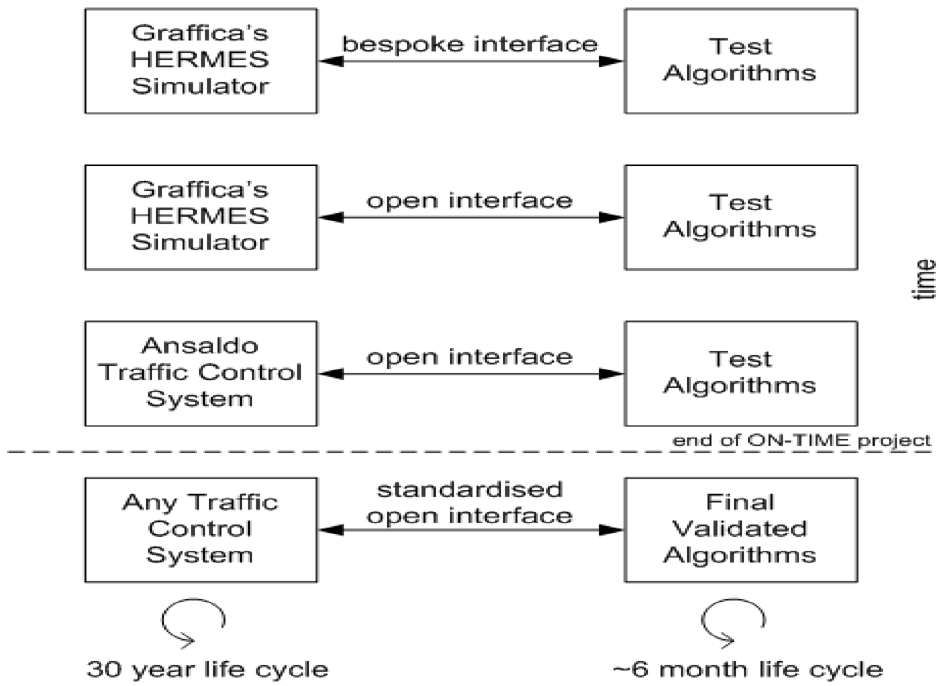

Figure 39 - High level vision algorithm development

6.1.3 Evaluation criterion

The development of a standardised evaluation function allows the direct comparison between different solutions over a wide range of measures from journey time, service connectivity, energy, etc. as shown in Figure 40. The evaluation criterion can be used to assess different types of applications, as shown in Figure 40.

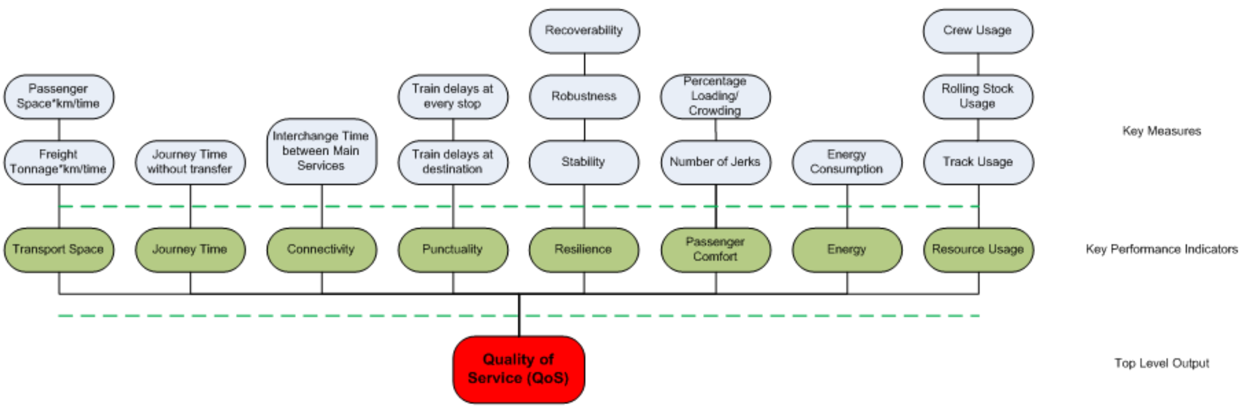

Figure 40 - Quality of services key performance indicators

6.2 Implementation into practice general requirements and guidance

This section will present some general requirements and strategies concerning implementation, deployment, usability, evaluation and integration of systems with regard to organisation and context.

6.2.1 Implementation, integration and deployment

Task 2.4 of the ON-TIME project has the objective to answer the question:

"How to implement developed methods into practice".

This involves several aspects such as:

"To take this step, human, technological and organizational aspects must be considered, as well as political and commercial aspects"

Here we will present some general requirements and strategies concerning implementation, deployment, usability, evaluation and integration of systems into existing organizations.

We will also discuss some experiences from earlier projects where research results have been implemented into practice.

6.2.2 From research to implemented system

Experience shows that there is a long way from research, via prototypes and laboratory tests and evaluations, to fully developed, implemented and deployed systems that contribute to improved operations and services in an organisation in practice. We will here identify some important success factors and common pitfalls.

One fact, that must not be underestimated, is that important knowledge generated, its rationales, must be kept intact during all development and implementation phases. During the research phase, a lot of knowledge concerning present organisation and systems, problems, needs, expectations and requirements is generated. Some parts of this can be formally documented in a structured way, e.g. systems/problem analysis and process models, but much of the information that is relevant for future phases is in the form of knowledge, understanding and new competencies of involved persons in different roles.

It has been shown that it is not possible to *formally* specify all requirements in such a way that the expected system is developed and implemented. A problem that is often encountered is that what is finally delivered to the users differs significantly from what was originally specified and from what the users actually need. One reason for this is that the knowledge needed for complete requirements is not available from the beginning but is generated during the different project phases. It is actually only when the final prototype is evaluated that the final requirements can be fully specified. Another reason is that important requirements, e.g. concerning details in functionality and usability, cannot be formally specified in such a way that development can be based entirely on it. The competencies, knowledge and experiences of people involved are needed in addition to formal specifications.

An important consideration is the user-centred design is more than just Human-Machine Interface (HMI). Indeed, the HMI is a minor consideration in relation to the usability of the algorithms that will support the technologies delivered in On-time. If we take the example of WP6, the 'driveability' of the advice offered by the algorithms is probably a greater consideration than the presentation of that advice (if the advice is not driveable, either because the train cannot be driven to that advice, or it does not fits with the demands of the RU or the expectation of drivers). Therefore, acceptance of the technology rests on thorough checking, and fine-tuning of the algorithms so that they deliver not just technically optimal decisions, but decisions that are achievable when used in cooperation with human operators.

Important conclusions are:

- Use iterative, user centred development models. It is only when prototypes are produced, which are possible to evaluate under realistic circumstances involving skilled professionals, that detailed requirements can be specified.
- Keep the research and project teams and their common understanding intact as much and as long as possible. The researchers and the skilled professionals from the organisation develop a deep understanding for what is important; this knowledge cannot be formally described, but it is important for successful development towards the specified (and unspecified) goals.

Prototypes and demonstrators

User centred iterative development models means that prototypes are continuously developed and tested during the design and development phases. Prototypes are from the beginning often in the form of simple sketches, and later in the form of more or less executable test systems.

A demonstrator can be seen as a prototype, with certain well specified limitations, with the purpose of illustrating and confirming the functionality and interactivity of a future full scale application. When a demonstrator is developed, it is important to specify delimitations and restrictions made, so that evaluations are made in the correct context.

However, a demonstrator only has relevance if it realistic enough. All aspects which are important for later use in real contexts should be implemented in the demonstrator.

A demonstrator has its strength in the possibility to evaluate and support the specifications of the final system, planned to be implemented and deployed in a specific organisation.

6.2.3 Implementation and deployment

By *implementation* we mean the technical part of the introduction of a technical system in an organisation. Methodologies and models for this are well known and applicable. In practice several problems are often encountered, e.g. problems with technical infrastructure, networks, communication with other systems, database performance, etc. It is important to specify requirements for when an installed system shall be accepted as functional by the organisation, to perform risk analysis etc. It is not a part of the ON-TIME project to discuss this in more detail.

By *deployment* we mean the introduction of the new technical system into an existing organisation, its work processes, the required professional skills, work environment, etc. The organisation which the system is implemented in, will not be the same as it was before, but the organisation will always, and should, be developed into something new that can profit from the potential advantages of the new technology. "Do not pave old cow paths": this means that the organisation as such, including management, work processes, roles, competences, work environment, etc, must be developed. It is only when the organisation is changed, utilising the new technology, that the results will be increased efficiency, quality, safety or whatever the development objectives are.

When it comes to deployment, it is important to see the organisation as a social-technical system, where the technology is one component of several. The organisation, work processes, humans in different roles, their competencies, use of the technology, usability, work environment, communication patterns, etc, must also be considered. There are several reasons for this. One is that the use of competencies within the organisation often is an important success factor. When the potential users of the system are engaged, this also has an effect on understanding, acceptance, efficient future use, etc. It is also important because if the different aspects of the organisation are not changed in an appropriate way, the potential benefits will not be reached. People will keep on working according to old habits and rules, but with new tools that they cannot properly make use of.

There are many methods available for including competencies in the organisation in the development process, i.e. user centred development models.

The actual deployment process must be seen as a rather long process, including what is being done prior to, during and after the implementation of the actual technical system. It is only through adjustments *after* the deployment, based on evaluation and user experiences, that the potential benefits are reached. This means that a project should not be considered as finished, and the responsibilities handed over to the operational organisation, too early.

ON-TIME is unique in that it is a closed-loop, networked optimisation approach. It is therefore critical that the evaluation and assessment of new developments is viewed holistically. Improvements in one part of the system may have a negative impact elsewhere. Alternatively, minor improvements may have secondary value elsewhere. For example, driver so far have not necessarily seen value in reason information in the DAS HMI. However, it is anticipated that they may lead to saving of many calls to the controller centre which is a known problem. Therefore, while the savings to the driver are minimal, the savings across the rail control system could be valuable.

6.2.4 Usability and user centred models

It is convenient to use the ISO-9241 definition of usability as a basis for further discussion. Here, usability is defined as:

"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use".

This definition is extremely practical and useful. It states that usability only can be seen in a specific context of use, for specified users and for a specific purpose. It also says that it can, and must, be evaluated with respect to effectiveness (that the planned tasks can be completed), efficiency (that this can be done using reasonable resources) and satisfaction (subjective experiences, work environment, etc).

There are a number of different approaches and models available for user involvement in development and deployment processes. They sometimes have different labels, such as participatory design (PD), user centred systems development (UCSD), or user involvement. User centred development models should follow the ISO 9241-210 standard for *Human-centred design for interactive systems*.

This ISO standard describes some key principles:

- The design is based upon an explicit understanding of users, tasks and environments;
- Users are involved throughout design and development;
- The design is driven and refined by user-centred evaluation;
- The process is iterative;
- The design addresses the whole user experience;
- The design team includes multidisciplinary skills and perspectives.

It is especially necessary to understand the importance of the organisational context. A system that works in one organisation might not work at all in another. If the organisational aspects are known and considered, this obstacle can sometimes easily be overcome by appropriate adaptations, education and training. The use of available competencies within the organisation is here important. "Listen to your users" is an often used expression. This should not be interpreted as only the direct end-users, but all competencies within the local organisation, including management. When it comes to detailed specifications of what the work processes in the new organisation will look like, the competencies of the involved users must be utilised. There exist very efficient models for this that can be found in literature.

One important common experience and lesson, is that usability is built in from the very start of a development project. It can never be added afterwards or even at a late stage. The basis for developing usable, efficient computer and information systems is generated when the goals and the initial requirements are formulated. Usability must be regarded and evaluated continuously during all phases of the development project. The deployment phase must also be user centred.

6.2.5 A vision seminar process

As an example of a method that has shown to be successful, a vision seminar process is here briefly described[2]. The process is useful in a socio-technical context, when interactive systems are to be developed.

The Vision Seminar Process is built up by a series of seminar occasions. They constitute the body of the process where the important user participation takes place. In these

[2] Johansson, N., Olsson, E. and Gulliksen, J. and Sandblad, B.: A Participatory Process Supporting Design of Future Work. In: Ergonomics, An Introduction. Ed: Singh, S.K., ICFAI University Press. 2007. ISBN: 81-314-0832-9, 2007.

seminars a work group with representatives for the future users meets together with one or two designers that also have the role of process leaders. Work in the vision seminars substantially aims at jointly developing a vision, a mutual idea and a visualization of what a future work as a whole should look like.

If new supporting IT systems are to be built in order to support future work, this future work must first be defined, thereafter requirements can be defined regarding suitable and usable IT systems. In such way, a new IT system should not be designed for the organization that exists today, but for the future organization. To make full use of the existing potential of improvement, possibilities to change the organization in different aspects such as work processes, responsibilities, competences, management, etc must be used.

It is important to reach a consensus within the organization about which direction the vision is striving for and what plans are ahead.

When user centred techniques such as this have been used, the resulting goals and requirements for technical development and deployment have been substantially improved. At the same time the engagement has also resulted in common goals for coming changes and better acceptance of the new technical systems and the new work organization.

6.2.6 Implementation in railway organisations. Swedish experiences.

The following refers mainly to experiences from a larger Swedish research and development project, where a new system, STEG, for operational train traffic control was designed, developed and deployed at a Swedish traffic control centre.
In short, the story behind the STEG system is the following:

- A research project, to generate basic knowledge about operational train traffic control, and to specify basic requirements for future control and support systems, was initiated by the Swedish Rail Administration.
- Through a user centered process, using e.g. vision seminar processes, new concepts for operational traffic control were developed and specified. Groups of experienced traffic controllers were active in these groups, during a period of more than three years.
- Prototypes of the new proposed system were developed in an iterative way, with strong user involvement.
- The laboratory prototypes we tested and evaluated by experiences traffic controllers and the results of the evaluation were used to iteratively improve the prototype. This was performed as a user centered design process.
- The Swedish rail administration decided to build an operational system based on the experiences from the prototype. The development process was performed in close cooperation between technical developers, the researchers and future users. The STEG system was deployed at one workstation.
- The deployment at one traffic control centre was performed during a period of several months, involving the future users. Training sessions combined with evaluations using e.g. user diaries were performed.
- During the first months of use the traffic controllers were supported by experts who

both supported the users and collected experiences for future improvements.

- After a long period of use, around two years, the next step in development was taken. The STEG system was developed as a multi user system for a complete traffic control center.
- The introduction into this new traffic control centre was made with a limited effort concerning user support and evaluation, which caused severe problems for the users and for the organization. It took a long time to understand the reasons why this second deployment project was less successful, and to compensate for these shortcomings.

Summarising guidance for the future

To summarize the experiences from this and other similar projects, some important success factors and pitfalls are:

Success factors

- To generate a common understanding of development objectives in the whole organization.
- To have a strong commitment from all levels of management.
- To involve experienced users in all phases of the development and deployment process.
- To keep development teams intact during the development phases, so that there is continuity in using generated knowledge and experiences.
- See the organization as a socio-technical system. Include e.g. usability, work processes, work environment and competence development in the project.
- Enforce a social-technical approach to development, combining technical, organizational and human aspects of the new systems and work processes.
- Look for benefits and issues at the whole system level – not just locally – this is critical given the whole, closed-loop, networked nature of ON-TIME.

Pitfalls, to avoid

- To focus on technical development, without considering user and organizational aspects.
- Not consider usability aspects from start of the project, but trying to add such considerations later.
- Focus on HMI. Not treating algorithm design as also having a user-centred dimension.
- Treating automation as a 'closed box'. Automation needs to 'explain itself' to operators to give them trust, and to allow them to justify their decisions to other skateholders (e.g. for timtablers to justify their decisions to the RU).
- To develop an operational system only based on technical requirement specifications. All aspects of the new systems cannot be formally specified!
- Introduce new technical systems without proper training, not only concerning the new technical system but also concerning the new work processes and roles.
- To introduce large systems at the same time, and not do it stepwise.
- To end user support too early after deployment. Deployment takes time.

7 RESEARCH OUTPUT AND FUTURE TASKS

7.1 Innovation 1: Standardised definitions and methods to create interoperable processes

7.1.1 State of the art

Before the start of ON-TIME project, there was a lack of knowledge and documentation about underlying railway operations management processes for strategic planning, tactical planning, operational traffic control and train driving.

There was also a lack of methods to connect the timetable planning and operational traffic control processes. In the project plan this was divided into WP3: timetable planning, WP4: methods for minor perturbations, WP5: major disruptions and RU asset management and WP6: train driver advisory systems (DAS).

The process for timetable planning and operation is harmonised in Europe by Rail Net Europe, European legislation, Network statements and European corridors for freight traffic. The process for timetable planning is divided into a one year timetable and ad hoc adjustments of the timetable. The ad hoc timetable process meets the operational process. There is a need to close the loop and better connect timetable planning and operational methods. There is also a need for better evaluation and feedback.

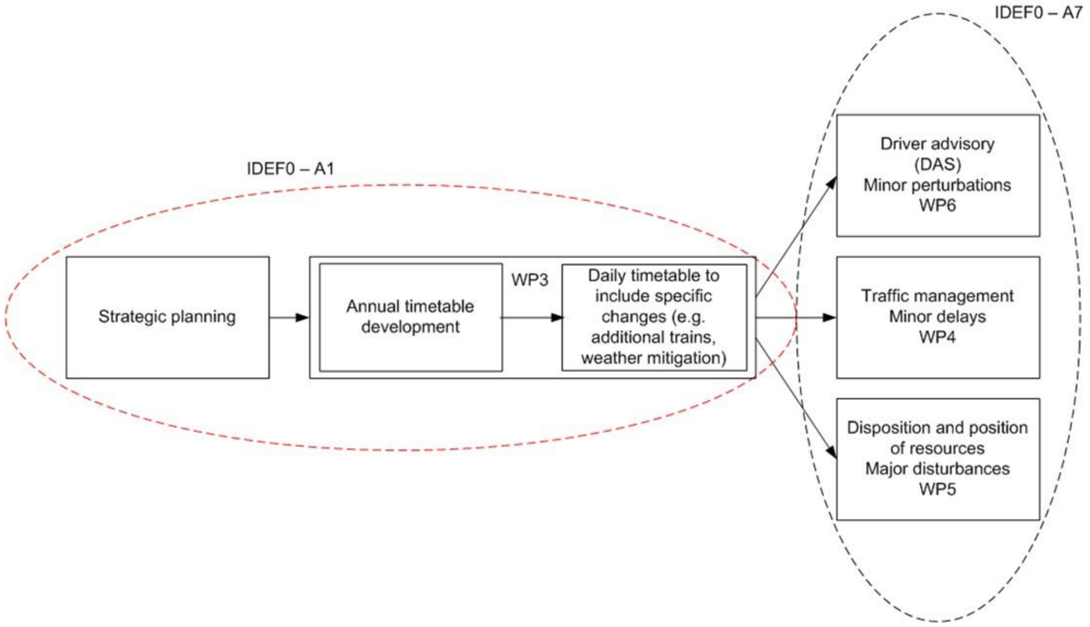

Figure 41 - Timetable planning and operational process (Deliverable 2.1)

To estimate available capacity, the number of possible train paths and the robustness of the system, the IMs have common methods. The UIC 406 method outlines how track occupancy can be calculated. For traffic simulations there exist some commercial tools used by the IMs. Traffic simulation is used to analyse and help the planners to provide quality in the timetable. There is also a need to simulate the operational process. In ON-TIME a simulator system, HERMES, has been used and a demonstrator system has been developed by Ansaldo. Another future need is to have common procedures for IMs, RUs and Industry to share and exchange data. In ON-TIME, data has been transferred from IMs traffic simulation systems to the simulator HERMES and the demonstrator developed by Ansaldo and NTT Data. RailML is a standard for transferring data.

7.1.2 Research aims and objectives

The main objective is to give a process framework for timetable planning and operation, to demonstrate how innovations can be implemented into practice and how we can measure the benefits of innovations. The improved methods in timetable planning and decision support in traffic control and train driving will:

- decrease track occupancy in bottlenecks;
- improve punctuality;
- improve energy efficiency.

A second objective is to outline a framework for future research in timetable planning, methods for handling minor perturbations, major disruptions and train driver advisory systems.

7.1.3 Research outputs

1. A framework for developing an objective function for evaluating work package solutions (Cost function);
2. Functional process descriptions for United Kingdom, Sweden, Germany, Italy, Netherlands and France;
3. State of the art studies and review of present technology readiness levels (TRL) for traffic planning and timetabling (WP3) level 3, traffic control under minor perturbations (WP4) level 3, Operational management in the event of large disruptions (WP5) level 3, Driving Advisor system (WP6) level5. The TRL indicates the level of maturity of the technology and the possibilities for implementation into products and standards;
4. Capability requirements of ON-TIME innovations. 1: Standardised definitions and methods to create interoperable processes, 2: Improved methods for timetable construction, 3: Decision support for traffic controllers to handle minor perturbations, 4: Development of methods and algorithms to handle major perturbations, 5: Standardised communication between traffic controllers and train drivers, and 6: IT architecture and standards for train control data;
5. Specifications for ON-TIME workpackage integration and interaction, see Figure 2. Traffic control for minor perturbations can be automated. Traffic control handling large events needs RU decisions;
6. Specification of locations and scenarios for ON-TIME simulator system and demonstrator system:
   - Sweden – Norway, Iron Ore Line Kiruna – Narvik, single track line and border crossing.
   - United Kingdom, East Coast Main Line, double and multiple track line.
   - Italy, Bologna station, node with many merging lines .
   - Netherlands, corridors through s’ Hertengenboersch, network, complex node.

The Netherlands network was imported to HERMES from network data in RailML. The Iron Ore line was imported to HERMES from a RailSys model.

1. Simulations have been performed, and the systems developed for solving perturbations and disruptions in the different scenarios have been evaluated.
2. Using the demonstrator system, the different scenarios have been further studied and the results visualized using graphical interfaces.
3. Specification of a strategy for putting methods into practice, including guidelines for a deployment process.

7.1.4 Future tasks

The ON-TIME project has connected timetable planning and decision support for the operational process:

- Infrastructure managers and system suppliers need to develop methods to export and import data with de facto standards as RailML.
- There is a need for continuing algorithm development and integration to provide decision support tools for Railway Operations from strategic planning to on-the-day operations, including disruption handling and recovery as well as planning and undertaking maintenance related activities and engineering activities.
- There is a need for further development of interactive solutions, i.e. systems where human traffic controllers can interact with the decision support tools developed in ON-TIME.
- There is a need for further research in timetable planning, methods for handling minor perturbations, major disruptions and train driver advisory systems. Further research will be done in other EU projects, i.e. Capacity4Rail 201310 – 201709.

7.1.5 Deliverables and results

The main results of this part of the project are summarized in reports available publically on the website www.ON-TIME-project.eu:

- Task 2.1: Railway planning and operation process (Level 1) led by UoU;
- Task 2.2: Approach and specification of innovations and technical readiness level for WP3 to WP6 (Level 1) led by UoB;
- D1.2 A framework for developing an objective function for evaluating work package solutions (Cost function);
- D2.1 Review of capacity restrictions, railway planning and operations, problem description and existing approaches (including state-of-the-art);
- D2.2 Approach and specification of system integration and demonstration;
- D2.3 Evaluation of innovations and demonstrators and a strategy for putting methods into practice.

7.2 Innovation 2: Improved methods for timetable construction

7.2.1 Implementation into practice

The current timetabling practice in different countries shows a separation of microscopic and macroscopic approaches with either macroscopic models to compute network timetables using normative input, or microscopic blocking-time based tools for detailed planning on corridors and stations but without support for network optimisation. Timetable evaluation on feasibility, stability or robustness is typically applied – if at all – after the timetable construction using simulation tools with unclear procedures how the results are used to improve the timetable design. The ON-TIME timetabling approach aimed at integrating the microscopic and macroscopic timetabling as well as the timetable evaluation into one timetabling design process with an explicit focus on timetable performance indicators.

WP therefore developed a classification of Timetabling Design Levels depending on the explicit incorporation of performance measures in the timetable design process with increasing performance with respect to dealing with delays and disturbances. The Timetabling Design Levels (TDL) go from TDL 0 of low quality timetables to TDL 4 by successively incorporating stability analysis, conflict detection, robustness analysis, and resilience into the timetabling process, resulting in timetables that are more and more robust and resilient. The IMs can use this classification to evaluate the TDL they currently have and then introduce structural measures or procedures to increase the TDL to achieve better performing timetables.
The developed three-level timetable module demonstrates one path to TDL 3 (stable, conflict-free and robust timetables) and TDL 4 with respect to resilience to inserting ad-hoc freight path requests. The performance-based timetable module includes microscopic running time calculations and conflict detection, macroscopic timetable optimisation, energy-efficient speed profile computations, and timetable performance evaluation of feasibility, stability and robustness integrated in the timetable construction. The algorithms work for standard RailML input and deliver a RailML timetable at microscopic level including energy-efficient speed profiles.
However, the various algorithms within the timetable module may also be replaced by existing tools at various railways to reach the same goal. For instance, railways that rely on normative macroscopic timetabling might integrate a microscopic module to provide more accurate input to the macroscopic approach, and enhancing their functionality with conflict detection and evaluation of infrastructure occupation. Other railways might add macroscopic optimization to their microscopic timetabling approach, and again others might add the fine-tuning module to their timetable process to obtain energy-efficient timetables, all depending on the current functionalities of their timetabling software and the ambition to improve on it. Nevertheless, the integration of explicit checks on conflicts, acceptable infrastructure occupation and robustness is believed to be the key to achieve better timetable performance.

7.2.2 Technology Readiness Level analysis

The ON-TIME project has realized a step change from TRL3 to TRL6 for improved methods for timetable construction (Innovation 2). The state-of-the-art of performance-based railway timetabling before the ON-TIME project was assessed at TRL3 with analytical and experimental critical functions and/or characteristic proof of concept. This means that algorithms for in particular conflict detection, capacity consumption, robustness analysis and energy-efficient speed profiles were available and tested but not in an integrated approach for timetable design.

The ON-TIME timetable module is based on algorithms from literature which were extended, implemented, and integrated into a timetabling architecture that computes a high-quality timetable using an appropriate internal data structure. All components were validated, including (energy-efficient) running time computations, conflict detection, capacity consumption, macroscopic timetable optimization, Monte Carlo robustness analysis, and stochastic timetable optimization in corridors. This component validation in a laboratory environment led to TRL4. TRL5 was realized by connecting the timetabling models to real railway data using the RailML exchange format, which was extended to include the microscopic level of detail required within the ON-TIME project. The overall architecture of the timetable module based on RailML input was tested and validated using RailML data of a real-world railway case study from the Netherlands railway network. This component validation in a railway environment led to TRL5. The integration of the timetable module via the RailML interface into the ON-TIME architecture (WP7) and the HERMES railway simulation environment finally led to TRL6. For this an API was developed that takes an extended timetable RailML delivered by the timetable module and replaces the old timetable with the new one. This enabled testing and evaluating the developed timetables for the case study from the Dutch railway network using the HERMES simulation tool.

In general, the developed timetable module is a step change into technology development for performance-based timetabling leading to high Timetabling Design Levels. A next step to achieve a timetable system prototype at TRL7 requires the development of a User Interface to set basic parameters and interact with the solution algorithms to set basic parameters, judge solutions, and guide the solution process with e.g. fixing some train paths, relaxing some running times or connections, cancelling train path requests, changing routes, etc. Also additional automatic features could be implemented that were out of the scope of the ON-TIME project, such as finding the optimal or most robust platform allocation and/or station routes for all train services. The developed Timetable Design Level classification, timetabling architecture, and algorithms, provide a perfect foundation for future research and development towards TRL7 and higher.

7.2.3 Research outputs

The key outputs of the research have been:

- The development of common railway timetabling and capacity estimation methods appropriate for use by all EU member states that reflect customers' satisfaction and enable interoperability, more efficient use of capacity, higher punctuality and less energy consumption;
- Further development of methods for robust cross-border timetables and integration of timetables between different regional and national networks improving interoperability and efficient corridor management including standardised approaches for exchanging timetable information between stakeholders;
- Improved timetable quality, stability, robustness, reliability and effectiveness;

- Validated development methods, through benchmarking, using a number of standard, real-world case studies.

The key academic work that has been undertaken is:

- Development of micro-macro network transformations;
- Microscopic conflict detection and capacity consumption;
- Macroscopic network timetable optimization including stochastic robustness evaluation;
- Computation of energy-efficient speed profiles;
- Stochastic optimization of optimal energy-efficient timetables on corridors between main nodes.

7.2.4 Future tasks

- The WP3 timetabling approach needs to be extended with a model for robust platforming and routing within station layouts.
- The UIC 406 infrastructure occupation calculations need to be evaluated further and in particular the stability parameters and the different options and limits to decrease saturation. Benchmarking calculations between different tools and for different national networks are of interest.
- The UIC 406 infrastructure occupation calculations need to be extended to advanced interlocking constraints such as overlaps and flank protection. This also needs further extension of RailML to include these interlocking characteristics.
- Further study is required into the interaction of timetabling (WP3) and traffic control (WP4) to obtain effective traffic control to resilient timetables with respect to perturbations.
- The ON-TIME timetabling approach to integrate microscopic and macroscopic timetabling is promising but the construction of stable timetables also needs microscopic stochastic simulation and analysis which needs further study.
- The macroscopic optimization model needs a functionality to fixate constraints or train paths to enable an interactive timetable construction by planners.
- The concept of a multilayer timetable with a multi-speed freight path catalogue needs further study to develop an effective freight path catalogue.

7.2.5 Deliverables and results

The main results of this part of the project are summarized in reports available publically on the website www.ON-TIME-project.eu:

- D3.1 Methods and algorithms for the development of robust and resilient timetables;
- D3.2 Benchmark analysis, test and integration of selected timetable tools.

The following software modules were developed or extended in the project:

- Microscopic timetabling models by TU Delft
- A macroscopic timetable optimization model by University of Bologna
- Energy-efficient speed profile computations and stochastic optimization of corridor timetables by TU Dresden.

They are available from the project partners and will form the basis for new developments after the ON-TIME project.

7.3 Innovation 3: Real-time traffic control algorithms

7.3.1 Implementation into practice

The separation of tasks into different modules has proven to be feasible. In order to obtain reproducible test results with the simulation, the software modules needed to fulfil the highest possible level of automation, i.e. they needed to work in fully automatic mode. That meant several challenges:

- Dealing with uncertainties in data message delivery (e.g. missing track occupation or release messages);
- Dealing with asynchronous data delivery (e.g. computation of a new real-time traffic plan, while the existing one is still being executed);
- Dealing with particular issues in relation to the simulator, e.g. simulator behaviour at different simulation speed, trains appearing and disappearing at the start and end of the simulated time/space.

The obtained solution has been designed to work on all the cases which occurred in the scenarios that were simulated. There might be real situations which are not yet covered by the current framework. The focus of the research project was on showing the feasibility of real-time perturbation management in a realistic, but technically well performing environment. The consequences of malfunctioning technology, in particular in the communication channel between the traffic management module and the traffic control system, have not been considered in this research project and must be solved by system integrators in implementation projects.

During experiments it became clear that the update processes, in particular of the real-time traffic plan as core object of real-time traffic management, needed to be defined more specifically in order to become more robust and for example, surely prevent deadlocks.

The examined scenarios are characterized by typical operational perturbations as can be observed in real-world railway systems. The major class of technical perturbations resulting in limitations in infrastructure availability (e.g. points failure) has in part been examined indirectly (e.g. an ATP failure can be modelled by assuming a smaller track speed on a particular section). For a wider analysis of this kind of technical incident, changes need to be made in the optimization modules and simulator. A full interaction with human dispatchers, with the driving optimization modules, connection management and the management of large disruptions (innovation 4) has been functionally described but has not been demonstrated in the real-time environment and must therefore be examined in more detail in future projects.

To allow better predictions of traffic, more information about train behaviour, interlocking and ATP rules need to be available; this would require further extensions in the data modelling, in the traffic state prediction module and in the simulation environment. For continuing development, simulation engines need to be developed with an open architecture similar to the one proposed in the project, in order to enable testing of smaller modules.

The most critical success factor of the algorithms in WP4 is a correct supply of static and dynamic data. For a real-world implementation, comprehensive processes for data management first need to be established by the railway infrastructure managers (e.g. concerning data changes, construction work, timetable changes). These update processes must then be considered for real-time traffic management.

7.3.2 Technology Readiness Level analysis

At the beginning of the project, the TRL of conflict detection and resolution for real-time perturbation management was estimated to be 3 (analytical and experimental critical function and/ or characteristic proof-of-concept) from the analysis of the state of the art. This means that algorithms, in particular for conflict detection and resolution, were available and tested in offline scenarios or in online scenarios with particular data sets and scenario information, but never tested in a closed loop control environment corresponding to a real-world setting, i.e. using only the information that also exists in real-world railway systems.

During the ON-TIME project, the entire tool chain for automatic perturbation management has been implemented and tested using the architecture environment with the simulation system HERMES acting as a real railway system. This demonstration took place using three entirely different locations concerning infrastructure topology and with different rolling stock and timetables, particularly the mix of trains and the headway requirements in the systems.

This corresponds to the targeted technology readiness level of 7 – a system prototype demonstration in a railway environment, with the HERMES simulation as the representation of a railway environment.

In order to reach the next TRL 8, for a completed and qualified system through tests and demonstration, the implementation and integration steps described above need to be addressed.

7.3.3 Future tasks

Within the scope of the project, it was not possible to test all designed functional modules of the real-time perturbation management in closed loop in simulations. This should be done in a next step and concerns in particular:

- The module of train path envelope computation in combination with driver advisory systems/ centrally guided train operation;
- The module of connection management, which is part of perturbation management for a few infrastructure managers (e.g. the Netherlands, Switzerland).

Furthermore, existing modules should be extended to incorporate more control options of perturbation management (e.g. re-routing, skipping stops): This would concern all existing modules of Traffic State Monitoring, Traffic State Prediction, Conflict detection and resolution, Automatic execution of RTTP and above all the simulation environment.

Furthermore, the module capabilities can only be realistically assessed, if more scenarios are computed. Therefore, all functional modules of the PMM needed improvements in computation time and in particular the simulation speed needs to be increased. In order to get better insight into the possible effects of PMM in real railway environments, more practically relevant dispatching models should be implemented in the simulator to benchmark the PMM against. These would also require the implementation of deadlock prevention measures. Furthermore it could be interesting, to directly include a human traffic controller in the simulation loop. This could be done by extending the Ansaldo HMI developed in WP4 and allow there the direct modification of the RTTP by the traffic controller.

7.3.4 Deliverables and results

The main results of this part of the project are summarized in the form of reports available publically on the website www.ON-TIME-project.eu:

- D4.1: Functional and technical specification of perturbation management module;
- D4.2: Tools for real-time perturbation management including human machine interface;
- D4.3 Benchmark analysis for algorithms, methods, human machine interfaces using simulator tests.

The following software modules were developed or extended in the project:

- Traffic State Monitoring by TU Dresden;
- Traffic State Prediction: modules by TU Dresden and Transrail;
- Conflict Detection and Resolution: DEJRM (University of Birmingham), RECIFE (IFSTTAR), ROMA (TU Delft), University of Bologna;
- Automatic Execution of real-time traffic plan: TU Dresden;
- HMI: Ansaldo.

7.4 Innovation 4: Improved decision support – handling major perturbations

7.4.1 Implementation into practice

Recovering from a disrupted situation to a feasible state in the network requires railway operators to perform changes in the timetable such as cancelling, rerouting or re-timing trains, changing the order of departures at stations, maintaining or dropping connections between trains, and also to perform reallocation of rolling stock and changes in crew schedules. This recovery problem is very complex and needs to be solved in real-time; it is therefore often heuristically solved manually by the railway operators or by using fast combinatorial optimization algorithms.

Furthermore, the problem is usually split up into three main phases that may be defined as timetable rescheduling, rolling stock rescheduling and crew rescheduling.

Most current solutions deal with a single rescheduling phase. There are just a few approaches that integrate two phases, namely either timetable and rolling stock rescheduling, or timetable and crew rescheduling.

The ON-TIME recovery approach aimed at further integration of the three main rescheduling phases (timetable, rolling stock and crew rescheduling).

The WP developed of a tool with a framework that consists of a closed loop in which each rescheduling phase (i.e. timetable, rolling stock and crew rescheduling) is solved by an efficient algorithm to find a good feasible solution and gets feedback from the other phases in order to obtain a good feasible solution for the whole system.

Experimental results with a large set of scenarios show that the tool computes feasible resource schedules in a couple of minutes. Such solution times are acceptable in practice. This shows that the iterative algo-rithm can be applied in a practical setting for the disruption man-agement process.

The algorithms of the proposed framework use standard RailML as input and ouput timetable. The adoption this standart may facillitate interchangebilty and flexible intergations of the ON-TIME framework for the recovery process.

7.4.2 Technology Readiness Level analysis

The improved decision support handling major perturbation (Innovation 4) has realized a step change in terms of Technology Readiness Level. At the beginning of the project the TRL for decision support handling major perturbation was assessed at TRL3 (Analytical and experimental critical function and/or characteristic proof of concept). The expected step was to reach TRL6 (System/subsystem simulation or prototype demonstration in a railway environment).

The state-of-the-art of decision support handling major perturbation considered algorithms for train timetabling rescheduling, algorithms for rolling stock rescheduling, algorithms for crew rescheduling. Most of the papers deal with a single rescheduling phase and few integrate two phases. This shows that active research is initiated that laboratory studies validate the aims of separate elements of the innovation and that not all components are yet integrated. This state-of-the-art confirms the assessment of a TRL3.

The design a framework that consists of a closed loop in which each rescheduling phase is solved by an efficient algorithm to find a good feasible solution and gets feedback from the other phases in order to obtain a good feasible solution for the whole system establish that the modules of theses phases can work together. As set of tests of the framework has been carried out in a laboratory environment, theses framework tests can correspond to an assessment as a TRL4.

The rolling stock and the crew rescheduling should consider the entire country. This large extension of the problem can be considered as a reasonably realistic environment for testing the integration of the modules. Therefore this validation in a laboratory environment led to TRL5.

The laboratory testing of the framework done within HERMES environment that is near a railway environment has been carried out with one scenario. The few number of scenarios tested and the open issues regarding the HERMES logging function are significant obstacles to led to TRL6.
To conclude this analysis of the steps of TRL, it must be noted that the TRL6 has not been reached.

7.4.3 Deliverables and results

The following deliverables and project results are publicly available. Please visit www.ON-TIME-project.eu.

- D5.1: Functional and technical requirements specification for large scale perturbation management;
- D5.2: Decision support tools for the optimal human supervisory control of the recovery processes;
- D5.3: Analysis of the benchmarking.

7.5 Innovation 5: Centrally Guided Train Operation (CGTO)

7.5.1 Technology Readiness Level (TRL) analysis

TRL of CGTO prior the ON-TIME project has been specified as TRL3 in the DOW. The ON-TIME project has raised this to TRL4 and TRL5, depending on the different components. As described in chapter 4.4.4 different issues in combination with work on the other work packages and the given time line impeded the validation of CGTO in the closed loop with WP4 and the simulation. For this reason the planned TRL6 could not be achieved within the project.

Component	TRL at project start	TRL at project end
Trajectory computation including time targets set by control centre decisions	3	5
Advice generation and display (HMI)	3	4
Standardized communication interface between IMs and RUs	3	5
CGTO as integrated system	3	4*

**TRL4: validation of components only*

Table 8 - TRL levels at project start and at project end

7.5.2 Implementation into practice and future tasks

In order to apply Centrally Guided Train Operation in real operations these steps should be followed after the end of the ON-TIME project:

- A demonstrator for CGTO working in closed loop with conflict detection/resolution and a simulated railway environment should be finished in order to finally prove the concept and to be able to generate and evaluate KPIs. This will allow IMs and RUs to quantify the potential benefit of CGTO and build the basis for necessary investments.
- Standardisation of the specified interface should be taken forward.
- Infrastructure managers have to upgrade their Traffic Management Systems to support driving advice (e.g. to collect the operational data required and generate target points) and set up a communication server with the specified interface.
- Train operators and industry have to include the specified interface in existing or newly developed DAS and enhance internal algorithms to consider targets set by control centre decisions in their calculation. Optionally, enabling the communication of current train data back to the control centre could further increase the quality of

control centre decisions.

7.5.3 Research outputs

State of the art analysis of 22 DAS (most of them not operational), of which only 8 are control-centre-connected, leading to the identification of key functions.

Based on existing experiences, three design alternatives distributing these functions differently between control centre and on-board components have been analysed and described as system architectures:

- DAS-C: mainly central intelligence;
- DAS-I: distributed intelligence;
- DAS-O: mainly on-board intelligence.

Specification and implementation of an XML-interface data format supporting all three system architectures mentioned above and enabling bidirectional communication between central and on-board components.

Enhancement of existing algorithms for optimising train speed profiles (trajectory computation) to match the time constraints set by control centre decisions.

Evaluation of different methods to present advice to drivers (e.g. speed vs. time targets) based on simulator studies and expert interviews, examination of useful contextual advice (e.g. icons displaying the reason of advice) and experimental implementation of the proposed HMI design.

Implementation of a demonstrator for all components, calculating and displaying driving advice based on control centre decisions.

7.5.4 Deliverables and results

The following deliverables and project results are publicly available. Please visit www.ON-TIME-project.eu.

- D6.1 Specification of a Driving Advisory Systems (DAS) data format;
- D6.2 Sample Human Machine Interface.

7.6 Innovation 6: Standardised ICT architecture supporting interoperability of operational data between industry stakeholders

7.6.1 State of the art

The increasing availability of distributed system architecture paradigms and messaging protocols has given information system designers the freedom to develop feature-rich software systems. The management of railway operations is an ideal candidate application domain for the use of these architectures, because of the geographical distribution and multi-stakeholder nature of most railway networks.

The ON-TIME Project proposes a distributed architecture to integrate different algorithms to solve typical problems in railway operations, such as timetable planning, dynamic re-planning of services at the macroscopic and microscopic levels following disturbance, resource management and rostering. Since it would be unfeasible to create a system capable of substituting current Train Management Systems, the purpose of the ON-TIME architecture is to complement Train Control Systems, extending their functionalities with a new layer of algorithms and real-time solutions to cope with the usually static planning of Train Control Systems.

The key-purpose of the architecture definition is to define a distributed, configurable and flexible infrastructure to exchange data and messages between different modules. The advantage of using a distributed architecture in this context is the ability to collect and exchange data on systems that are by their own nature loosely coupled and create a coherent, dynamic communication context in which this information can be exchanged.

Data and technical standards must be implemented in order to easily integrate a collection of systems and to represent data in a way such that regional differences between neighbouring systems will have a small impact on the communication semantic. Since most European Countries have different processes and data standards, a common data representation is needed. In line with EU open data policy, common standards must be used to encourage the adoption of the platform and to implement a uniform data representation that will supersede specific regional requirements.

Another key aspect is to treat modules as services that can be queried and interacted by other systems and users. This is a very important aspect of distributed systems: each functional module should be black-boxed and self-sufficient, to be easily replaced by another implementation using the same modular framework.
Since for real-time operational systems it is paramount to have data consistency and to avoid synchronization issues, seeing systems as services opens the possibility to abstract their data as services as well.
Furthermore, in order to facilitate the integration with legacy and brand-new systems, the architecture must use open communication standards and integration frameworks suitable for scaling from small-scale up to large-scale distributed systems without compromising performance.

7.6.2 Research aims and objectives

From an ICT perspective, the ON-TIME project aimed to deliver a modular, service-based, distributed processing architecture for use in the rail domain. The architecture would facilitate the integration of diverse ICT systems from around the industry by exposing a range of data integration algorithms via a standard communications interface.
The basic architectural principles defined by the ON-TIME team were:

- Modularity – Every element of the ON-TIME system should be a black box encapsulating a single function. Module abstraction would allow easy substitution of differing implementations of each functional unit, encouraging competition between module providers and facilitating user choice.
- Extensibility – Open and extensible software paradigms would be adopted, freeing developers from dependencies on particular programming languages and allowing

the ON-TIME architecture to evolve alongside the railway in future years.
- Distributed functionality – State-of-the-art protocols would be employed to allow the system to operate in distributed physical environments; a vital feature in the multi-stakeholder setting of the rail industry.
- Responsivity – Communication between different modules would be facilitated by the passing of event messages via the architecture, removing dependencies and allowing individual processing units to react to the changing environment of the operational railway.

7.6.3 Research outputs

The ON-TIME data dictionary was developed to help the project team identify the types and content of messages that may be required in the context of railway operations. It contains over 300 concepts relevant to the domain, along with definitions and mappings to the RailML data model as appropriate. In order to facilitate easy multi-site access to the resource the dictionary was implemented as a wiki; this had the added benefit of enabling cross-references between entries to be embedded as hyperlinks, helping users to easily navigate between related entries.

The ON-TIME architecture can be seen in Figure 38 and consists of a collection of loosely coupled data providers and data processing modules that communicate via the passing of event messages. The architecture utilises a publish-subscribe communication pattern, allowing modules to join or leave the system at any time without interfering with the communications to other elements of the framework. The well-known open source RabbitMQ messaging platform has been used to provide this functionality. RabbitMQ runs on all major operation systems and clients can be written using a wide range of programming languages, including Java, .NET, Ruby, Python, Erlang, PHP and C/C++. This flexibility is vital to the wide-scale adoption of the ON-TIME architecture.

Real-time data is passed as events in near-real-time from the live rail network / rail network simulator to the traffic management modules; when disruptions occur re-planning takes place and new events informing stakeholders of updated traffic management plans/crew rostas etc. are raised within the system.

Non-real-time data used by ON-TIME (for example infrastructure data and the current timetable) are cached by the platform and are provided via an on-demand, read-only public interface allowing it to be accessed as required by any system. By providing a single-source-of-truth for these key, largely static, data resources, the ON-TIME architecture reduces the risk of inconsistencies arising between stakeholder systems.

ON-TIME module integration specification allows generic processing modules to be added to the ON-TIME framework through the implementation of a standard set of interfaces. In particular the specification covers subscription to event queues, the production/consumption of event messages, and access to static data resources.

For the representation of messages to be exchanged within the system, the ON-TIME team needed to make use of an open, extensible set of data models that are well aligned with the terminology set identified in the data dictionary. A mapping was created between the dictionary terms and the RailML format, which is rapidly gaining traction within the industry, and the degree of overlap between the two is becoming established. The coverage of the ON-TIME data dictionary by the RailML model was good, with only a few gaps found. These included information on Interlockings, ETCS-type train control concepts, disruption information, crew duty assignments and resource conflicts. Where gaps existed project-specific data model extensions were created, and this work has included contributions back to the RailML community in the form of the candidate RailML interlocking model.

7.6.4 Future tasks

- Integration of in-the-field traffic control devices, and traffic management systems, with the ON-TIME architecture;
- Formalisation of the ON-TIME extensions to the RailML format. This is particularly applicable in the domain of dynamic data (signal state changes, vehicle movements etc.);
- Development of architecture extensions for user-focused tasks (advice to passengers via mobile apps, digital signage, electronic ticketing support etc.);
- Automatic extraction of RailML representations of network layouts from CAD packages or similar to feed the ON-TIME algorithms.

7.6.5 Deliverables and results

The following deliverables and project results are publicly available. Please visit www.ON-TIME-project.eu.

- D7.1 Library of data and communication models;
- D7.2 Architecture specification & integration requirements.

7.7 Demonstration of Iron Ore Line Scenarios

7.7.1 The Iron Ore Line, Sweden

The Iron Ore Line demonstrator illustrates the traffic on a single track line with a border crossing. The purpose is to evaluate ON-TIME systems for optimal re-planning and driver advisory systems, in case of minor perturbations. The traffic between Kiruna in Sweden and Narvik in Norway is simulated. Characteristics of the traffic are very heavy iron ore trains (8000 tons), long trains (750 m) and mixed traffic. The mixed traffic and special requirements for the iron ore trains make the optimality of planning and handling of perturbations extremely important. Delayed or cancelled trains are associated with very high costs.

Figure 42 - An Iron Ore Line train (Source: http://www.bahnbilder.ch)

The traffic control centre (TCC) in Boden, controlling the Iron Ore line, is a centre for development, testing and evaluation of future systems for traffic control in Sweden. A new system, STEG [1] has been deployed, which supports traffic controllers to re-plan traffic in an interactive time-distance graph. The continuously updated real time traffic plan (RTTP) is automatically executed to the traffic control system. For communication with the train drivers another test DAS system, CATO, has been deployed.

7.7.2 Objectives

The objective of this demonstrator is to implement the ON-TIME systems for perturbation handling in a model of the Iron Ore Line, and to evaluate how the developed optimization algorithms can solve perturbation scenarios.

7.7.3 Scenarios

Some perturbation scenarios for the Iron Ore Line have been identified and used for simulations. The scenarios have been specified based on an investigation of most common perturbations in real traffic situations. The scenarios are:

- One fully loaded iron ore train delayed from the original station;
- Long distance freight train, entering the Iron Ore line, delayed;
- Speed restriction due to maintenance work between two stations;
- Infrastructure problem. Point out of order at one station;

The results concerning how the re-planning algorithms, developed in the ON-TIME project, can solve problems in connection with traffic perturbations are especially interesting. Such algorithms will in the future be a part of new traffic control systems in Sweden.

7.7.4 Implementation of test system

The design of the ON-TIME system for perturbation management is visualized in the following figure.

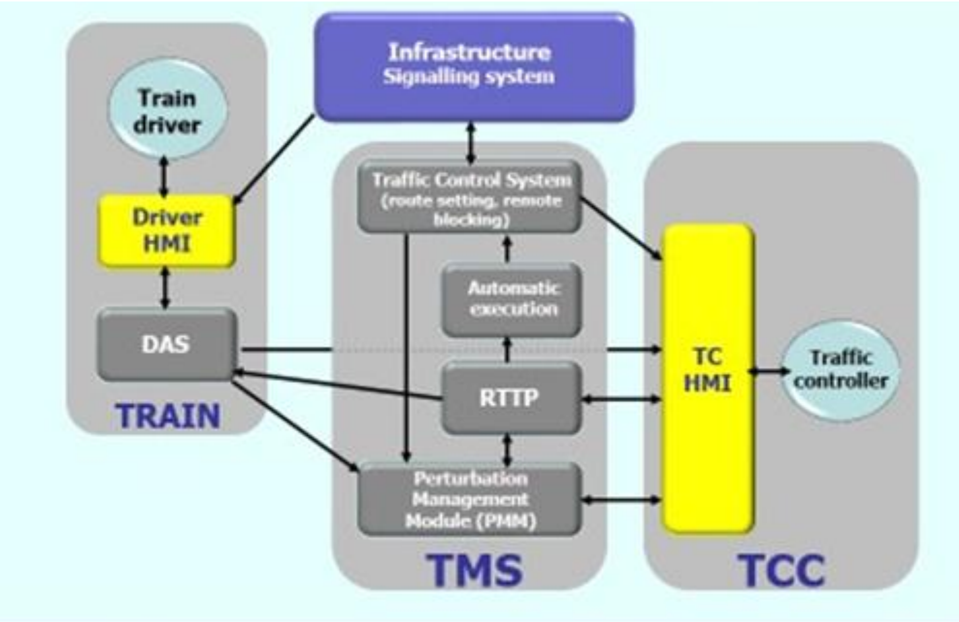

Figure 43 - Structure of the ON-TIME system for perturbation handling

The PMM detects the need for re-planning and calculates a new optimal RTTP (real time traffic plan). The RTTP is automatically executed. When required, the human traffic controller can perform re-planning tasks. The new plan is via a DAS (driver advisory system) sent to the train driver.
In the demonstrator for the Iron Ore Line, the interactive HMI for the traffic controller is not implemented. However, in the STEG implementation in Boden, a fully interactive system is in full operation. Future research will combine these two innovations: the fully interactive system and the supportive optimization algorithms developed in ON-TIME.

7.7.5 Results

Simulations for the different perturbation scenarios have been performed, using the HERMES system, developed in ON-TIME. The results of the simulations have been visualized in time-distance graphs. The solutions to perturbations, calculated and executed by the optimization algorithms have been analysed. The algorithms can solve perturbations and the result is a new RTTP that allows traffic to run with minimized delay times.

7.7.6 Future development

The developed tools for solving perturbations and disruptions will be further evaluated, using the simulator system. The results will be evaluated and compared to how the traffic controllers normally would solve the same scenarios for the Iron Ore Line.
The developed algorithms will also be further tested as integrated parts of the present interactive systems for traffic control (STEG) and driver support (CATO), implemented in Boden.

7.7.7 Deliverables

The main results of the projects are described in the form of reports available publically on the web site www.ON-TIME-project.eu, e.g.:

- D1.2 A framework for developing an objective function for evaluating work package solutions (Cost function);
- D1.3 Best practice, recommendations and standardisation;
- D2.1 Review of capacity restrictions, railway planning, problem description and existing approaches;
- D2.2 Approach and specification of system integration and demonstration;
- D2.3 A strategy for putting methods into practice and a formal evaluation of demonstrators;
- D4.1: Functional and technical specification of Perturbation management module;
- D4.2: Tools for real-time perturbation management including human machine interface;
- D4.3 Benchmark analysis for algorithms, methods, human machine interfaces using simulator tests;
- D7.3 Service-oriented architecture with integrated software artefacts.

8. REFERENCES

[1] B Sandblad, A W Andersson, A Kauppi, G Isaksson-Lutteman. Development and implementation of new principles and systems for train traffic control in Sweden. Computers in Railways XII, WIT-press, 2010, pp 441-450.
[2] B Sandblad, A W Andersson, A Kauppi, G Isaksson-Lutteman. Development and implementation of new principles and systems for train traffic control in Sweden. Computers in Railways XII, WIT-press, 2010, pp 441-450.
[3] N. Bešinović, E. Quaglietta, R.M.P. Goverde (2013). A simulation-based optimization approach for the calibration of dynamic train speed profiles. Journal of Rail Transport Planning and Management, 3(4), 126–136.
[4] E. Quaglietta, F. Corman, R.M.P. Goverde (2013). Stability analysis of railway dispatching plans in a stochastic and dynamic environment. Journal of Rail Transport Planning and Management, 3(4), 137–149.
[5] K. Panoua, P. Tzieropoulosb, D. Emery (2013). Railway driver advice systems: Evaluation of methods, tools and systems. Journal of Rail Transport Planning and Management, 3(4), 150–162.
[6] B. Jaekel, T. Albrecht (2014). Comparative analysis of algorithms and models for train running simulation. Journal of Rail Transport Planning and Management, 4(1-2), 14–27.
[7] D. Golightly, N. Dadashi, S. Sharples, and M. Dasigi (2014) Disruption management processes during emergencies on the railways. International Journal of Human Factors and Ergonomics, 2(2), 175-195.
[8] V. Cacchiani, D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf, J. Wagenaar. An Overview of Recovery Models and Algorithms for Real-time Railway Rescheduling, Transportation Research Part B, 63, 15-37, 2014.

ON-TIME Deliverables and Documents

- D1.2: A framework for developing an objective function for evaluating work package solutions (Cost function)
- D1.3: Best practice, recommendations and standardisation
- D2.1: Review of capacity restrictions, railway planning, problem description and existing approaches
- D2.2: Approach and specification of system integration and demonstration
- D2.3: A strategy for putting methods into practice and a formal evaluation of demonstrators
- Task 3.1: Assessment of state-of-art of train timetabling, ONT-WP03-I-EPF-008-03
- Task 3.3: Problems_in_the_integration_of_timetabling_and_traffic_control, ONT-WP03-T-UOU-021-01
- Task 3.4: Functional design of robust and resilient timetable models, ONT-WP03-I-UDB-009-03
- D3.1:Methods and algorithms for the development of robust and resilient timetables, ONT-WP03-D-TUT-034-01
- D3.2: Benchmark analysis, test and integration of selected timetable tools, ONT-WP03-D-TUT
- D4.1: Functional and technical specification of perturbation management module
- D4.2: Tools for real-time perturbation management including human machine interface
- D4.3: Benchmark analysis for algorithms, methods, human machine interfaces using simulator tests
- State-of-the-art of Recovery Algorithms for Real-time Railway Optimization, ONT-WP05-I-UDB-013-01.
- D5.1: Functional and technical requirements specification for large scale perturbation management, ONT-WP05-I-IFS-015-01.
- Input to task 5.2 Questionnaire, Events leading to perturbations and actions, ONT-

WP05-I-RFI-0 [17,18,19,20].

- Input to WPs 5.3 and 4.4 Incident selection study, ONT-WP05-T-UON-008-01.
- D5.2: Decision support tools for the optimal human supervisory control of the recovery processes, WP05-D-EUR-026-03.
- T6.1: Assessment of state-of-art of driver advice systems
- D6.1: Specification of a driving advisory systems (DAS) data format
- D6.2: Sample Human Machine Interface
- D7.1: Library of data and communication models
- D7.2: Architecture specification & integration requirements
- D7.3: Service-oriented architecture with integrated software artefacts
- D8.2: Demonstration scenario1 International cross boarder
- D8.3: Demonstration scenario 2 Complex node/line
- D8.4: Demonstration scenario 3 Simulations of Iron Ore line