Deliverable D 2.1
Intraday Operational Planning

Project acronym:	PLASA 2
Starting date:	01/09/2018
Duration (in months):	26
Call (part) identifier:	H2020-S2RJU-CFM-2018
Grant agreement no:	826151
Due date of deliverable:	8
Actual submission date:	24-04-2019
Responsible/Author:	Maurice Becker (DB), Michael Geilke (DB)
Dissemination level:	PU
Status:	Issued

Reviewed: (yes)

Document history		
Revision	*Date*	*Description*
1	24-04-2019	First issue
2	06-03-2020	Update

Report contributors		
Name	Beneficiary Short Name	Details of contribution
Maurice Becker Torsten Betz Michael Geilke Adam Kaluza Carla Terschlüsen Markus Zinser	DB	Gathered information from potential stake-holders; gave input for user stories; wrote deliverable; reviewed document
Emma Solinen, TRV Magnus Wahlborg, TRV Jennifer Warg, KTH Ingrid Johansson, KTH	TRV, KTH	Gave input for user stories and connecting EU projects; reviewed document
Rolf Gooßmann Mahnam Saeednia	HAC	Gave input for use cases and connecting EU projects; reviewed document

Table of Contents

1. Executive Summary

Nowadays, operational planning is performed by several groups, each of which is responsible for one or several parts of the planning process. These groups work together in close collaboration to come up with an initial rail operation plan. Every time changes need to be made, a subset of these groups adapts the existing solution accordingly. If these changes arrive early enough, the collaboration time will not necessarily carry weight. But if they arrive on short notice, a solution will possibly be not found on time, so that rail operations are affected negatively. As this could trigger further changes that again cannot be applied on time, these negative effects could be propagated throughout the rail network.

With today's technology, it is imaginable that operational planning is conducted by computer systems that respond to changes within short reaction times – either as a fully integrated system or several subsystems assisting humans during the planning process. One potential component of such a (sub-)system are simulation tools that evaluate the currently available operation plan or parts thereof.

In this document, we describe several use cases in which a simulation tool could be useful as part of an intraday operation planning system, derive necessary requirements, and assess which modifications are required to meet these requirements in PRISM. For many of these requirements, it turns out that PRISM either supports them already or could be modified accordingly.

2. Abbreviations and acronyms

Abbreviation / Acronyms	Description
RU	Railway undertaking
CCA	Cross Cutting Activities
IM	Infrastructure Manager
OPS	Operational Planning System
TMS	Traffic Management System
WP	Work Package
CA	Consortium Agreement
IP	Innovation Programme

3. Background

The present document constitutes Deliverable D2.1 "Possibilities for application of the PLASA approach in operational planning" in the framework of the WP 2, Task 2.3 of CCA.

4. Objective/Aim

In this document, we (1) describe potential use cases for employing macroscopic rail simulations as part of an (intraday) operational planning system, (2) discuss which requirements a corresponding simulation must meet, and (3) assess whether PRISM meets these requirements already or requires further enhancements.

To specify use cases and requirements as precisely as possible, we first describe the setting of operational planning, in Section 5, and a formal framework to define relevant terms, in Section 6. Subsequently, we define offline, online, and intraday operational planning and explain how operational planning relates to traffic management systems, in Sections 7 and 8. Based on the formal framework, we discuss seven use cases in which a macroscopic rail simulation could add value to planning processes, in Section 9, and assess whether PRISM is suitable, in Section 10.

5. Stakeholders, Resources, and Constraints

Operational planning systems for rail operation, also denoted as OPS in the following, are shaped by three types of factors: the stakeholders that are interested in the planning system or its output, the resources that are part of rail operations, and external parameters that need to be respected. In this section, we provide examples for each type and define which of the factors are relevant for this deliverable.

5.1. Stakeholders

We distinguish between stakeholders that are interested in rail operations in general, stakeholders that are interested in the current state of rail operations, and stakeholders that actively participate in rail operations:

1. Stakeholders that are interested in rail operations in general:
   a. Governments and other rail regulation authorities regulate rail operations on behalf of customers, tax payers, and competitors. For example, by Directive 2012/34/EU of the European Parliament, railway undertaking companies should have equal access to public rail infrastructure.
   b. Customers (passengers and freight traffic companies) are interested in new connections, shorter travel times, and higher service frequencies.

2. Stakeholders that are interested in the current state of rail operations:
   a. Other transportation companies, e.g., airline companies or bus companies, are possibly interested in current delays or major disruptions, as this could have a direct impact on their own operations.
   b. Customers, i.e., passengers and freight traffic companies, want to know which connections are available to reach their destination or whether they reach their destination on-time.
   c. Freight intermodal transport companies as container traffic companies on rail and road.

3. Stakeholders that actively participate in rail operations:
   a. Regional and local passenger railway undertaking company
   b. Long distance passenger railway undertaking company
   c. Infrastructure manager (timetable planning and dispatching railway network)
   d. Freight undertaking company, yard manager and terminal operator
   e. Infrastructure maintenance
   f. Rolling stock maintenance

Each of the stakeholders poses certain requirements either concerning the planning system itself or the resulting output.

5.2. Resources

Resources are entities that are directly or indirectly controlled as part of rail operations. The interface between rail operations and its resources can be either a human-human interface, a human-computer interface, or a computer-computer interface. For rail operation planning, the following types of resources are, in our opinion, the most important ones:

- Infrastructure
- Trains
- Rolling stock
- Infrastructure maintenance
- Rolling stock maintenance
- Staff

Other types of resources are typically not the focus of rail operations and can be modelled similarly to the resource types mentioned above. For example, food and beverages provided to passengers can be modelled in the same way as staff. They move inside and outside of the rail infrastructure but only follow a fixed schedule when moving on the rail infrastructure (at least from the perspective of railway undertaking companies).

5.3. Constraints

The operation plan produced by an OPS needs to respect external parameters that cannot be controlled (extrinsic constraints) and external parameters that can be controlled directly or indirectly by rail operations (intrinsic constraints):

- Extrinsic constraints
  - Weather
  - Disruptions
  - Strikes
  - Demand
  - Laws
  - External partners

- Intrinsic constraints
  - Quality of rail operations in terms of affected resources (e.g., punctuality of trains)
  - Customers reach the destination not later than a certain number of hours (if a connecting train waits for a delayed train, the overall punctuality of trains suffers but many passengers might benefit)
  - Energy consumption
  - Costs
  - Attrition

An operational planning system should find an operation plan that is optimized with respect to the intrinsic constraints but also anticipates modifications if extrinsic constraints change. For example, an initially optimal train path requires re-planning because of a major interlocking failure.

6. A framework for operational planning

As planning processes are typically very complex and differ from company to company, we present a simplified framework for operational planning in this section. Its main objective is to enable precise definitions of topics discussed in this deliverable, including the difference between online and offline planning, the definition of intraday operational planning, and the relation to other subsystems (e.g., traffic management systems). We present the framework as a sequence of definitions, each of which is provided with a brief explanation and an explanation how real-world planning process can be mapped to it.

We start with the definition of a rail network, which is represented as a mathematical graph:

Definition 1: A rail network is a tuple *(N, S),* where *N* is a set of nodes *N* (e.g., stations or stopping points) and *S* is a set of route sections $S \subseteq N \times N$. Nodes and route sections are also called infrastructure elements.

In real-world networks, nodes (e.g., a station) sometimes stretch across large areas so that more than one route section lies between them. For example, assume that there are two nodes n_i and n_j, which are connected by two train routes. In this situation, this rail network can be easily transformed to comply with Definition 1 by replacing n_i and n_j by new nodes $n_{i,1}$, $n_{i,2}$, $n_{j,1}$, $n_{j,2}$, and adding route sections $(n_{i,1}, n_{j,1})$ and $(n_{i,2}, n_{j,2})$.

Based on the rail network, the movements of resources such as trains, rolling stock, and staff can be easily described as a sequence of nodes, thereby providing location information of resources. In addition to the location information, each sequence element is also associated with planned arrival and departure times, where the arrival and departure time is set to the same time if a resource does not dwell at a node.

Definition 2: Given a rail network (*N, S*) and a resource *R*. A resource plan element for *R* is a tuple (n, $t_{arrival}$, $t_{departure}$), where $n \in N$, $t_{arrival}$ is the time at which the resource arrives at *n*, and $t_{departure}$ is the time at which the resource departs from *n*. The components of a resource plan element *rpe* can be accessed using dot notation: *rpe.node*, *rpe.arrival*, *rpe.departure*. A resource plan for *R* is a sequence of resource plan elements (rpe_1, rpe_2, …, rpe_k).

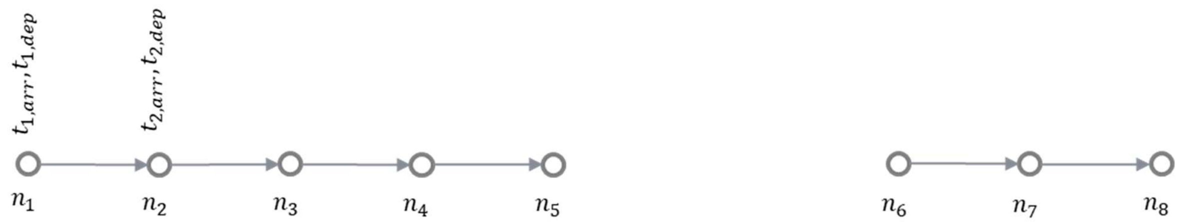

Figure 1: An example for a resource plan. The resource travels from node n_1 to n_5 on the rail network, uses another means of transport between node n_5 and n_6, and travels from node n_6 to n_8 on the rail network again. $t_{i,arr}$ and $t_{i,dep}$ are arrival and departure times, for $\mathbf{1} \leq \mathbf{i} \leq \mathbf{8}$.

A resource plan describes movements on a given rail network (*N, S*). As resources, such as staff, can use means of transport that are outside (*N, S*), a resource plan permits leaving and re-entering the rail network at arbitrary positions (see Figure 1 for an illustration). For other resources, such as trains, rolling stock, and train journeys, this is usually not desired. Therefore, for these resouces, an additional condition enforces that all movements take place on the rail network:

Condition 1: Let (*N, S*) be a rail network, *R* be a resource, and (rpe_1, rpe_2, …, rpe_k) be a resource plan for trains, rolling stock, or train journeys. The following condition must hold for (rpe_1, rpe_2, …, rpe_k): $(rpe_i.node, rpe_j node) \in S$ for all $1 \leq i, j \leq k, j = i + 1$.

Definition 1 and 2 only provide the most basic information of a resource plan. Other information such as construction sites, infrastructure-related disruptions, or train-related disruptions are summarized by so-called entity-related constraints that are either attached to the infrastructure elements or to the resource plans. They allow us to describe the desired rail operations more precisely by restricting the operational planning in terms of degrees of freedom.

Definition 3: An entity-related constraint is a triple (*e, a, v*), which associates an entity *e* with an attribute-value pair (*a, v*), where *e* is either an infrastructure element or a resource.

Example 1: Entity-related constraints provide a flexible means to describe the current and future states of infrastructure elements, resources, and resource plans. The level of detail can be chosen as desired:

- Route section (n_1, n_2)
    - ((n_1, n_2), length, 2000m)
    - ((n_1, n_2), protection system, ETCS2)
    - ((n_1, n_2), maximally allowed speed, 250 km/h)
    - ((n_1, n_2), infrastructure available, true)
    - ((n_1, n_2), number of tracks, 1)
    - ((n_1, n_2), electrified, true)

- Train journey tj_1 (could be a single resource plan element or a sequence thereof)
  - (tj_1, train type, 407)
  - (tj_1, number of seats, 460)
  - (tj_1, number of passengers, 315)

- Train journey tj_2 (a single resource plan element)
  - (tj_2, wait at node n_i, at least 5 minutes)

- Staff *s* (a resource plan)
  - (*s*, staff id, 3458910)
  - (*s*, job title, engine driver)
  - (*s*, allocated working hours, 6)

As illustrated with the examples above, entity-related constraints provide means to define static properties such as the length of an object as well as dynamic properties such as the availability of route sections. For example, if an infrastructure element was currently unavailable due to a tree lying on the tracks, we would either add a constraint ((n_1, n_2), number of tracks, 0) or ((n_1, n_2), infrastructure available, false). Which of these two options are added is mostly a matter of taste but also depends on the tools that process the constraints.

More complex planning processes such as rolling stock maintenance are either described by entity-related constraints or by modifying existing resource plans. For example, one could extend an existing resource plan for rolling stock *rs* by adding a sequence of resource plan elements that start at the latest location of *rs* and end in the required maintenance depot. The stopping time of the last stop could be set to the required maintenance time. To ensure that a sufficient time is allocated for maintenance, one could also add an entity-related constraint (*rs*, required maintenance time, 8:00). This way, a delayed arrival at the maintenance depot does not shorten the maintenance time.

So, in other words, schedules that are typically outside the scope of operational planning such as maintenance or construction sites, etc. can be described by modifying the availability of the considered entity for specified periods – either by modifying the resource plan and/or adding entity-related constraints.

Dependencies between resources such as "rolling stock *rs* is carried out by engine driver *driver_3458910*" could in principle also be described by entity-related constraints. To make these dependencies more visible, however, we introduce the notion of resource associations:

Definition 4: Let R_1 and R_2 be two resources. A resource association is a tuple (R_1, R_2, s_1, s_2, t_1, t_2), where $s_1, s_2 \in N$ are the start and end locations of the association, $t_1, t_2 \in N$ are the start end times of the association, and R_1 depends on R_2.

Example 2: If an engine driver *driver_3458910* gets sick on short notice, we remove the association (*rs*, driver_3458910, s_1, s_2, t_1, t_2) and add a new association (*rs*, driver_2768910, s_1, s_2, t_1, t_2).

With Definition 1-4, we can finally define the output of an operational planning system:

Definition 5: A rail operation plan *ROP* is a tuple consisting of a rail network *(N, S)*, a set of resource plans *RP*, a set of entity-related constraints *C*, and a set of resource associations *RA*. It can be modified as follows:

1. adding / modify / removing a resource plan
2. adding / modify / removing a resource plan element
3. adding / modify / removing an entity-related constraint
4. adding / removing a resource association

The elements of *ROP* can be accessed using dot notation: *ROP.nodes = N, ROP.routeSections = S, ROP. plans = RP, ROP. associations = RA, ROP.attributes =* $\{a \mid (e, a, v) \in C\}$*, ROP.values =* $\{v \mid (e, a, v) \in C\}$.

Modifications to a rail operation plan are typically not applied directly but must be accepted by the affected resources and the responsible authorities (e.g., a traffic manager). Therefore, we provide a modification request protocol along-side a rail operation plan (see Figure 2 for an illustration):

Definition 5: Given a rail operation plan *ROP*. A modification request M_{ROP} for *ROP* is processed according to the following protocol:

1. M_{ROP} is submitted to the affected resources and the responsible authorities. The required time for submitting M_{ROP} to all affected resources and responsible authorities is denoted as $t_{submission}$.
2. A response to the request is expected within a pre-defined period $t_{response}$. The response can either be a confirmation of M_{ROP} or a rejection accompanied by an appropriate explanation (e.g., the rolling stock cannot go to n_k, because it requires maintenances within the next 500 km).
3. If every party accepts the request, M_{ROP} is applied to *ROP*. The required time for applying M_{ROP} to *ROP* is denoted as $t_{processing}$.

The submission time, the response time and the processing time of M_{ROP} can be accessed via dot notation: M_{ROP}*.submission,* M_{ROP}*.response* and M_{ROP}*.processing*.

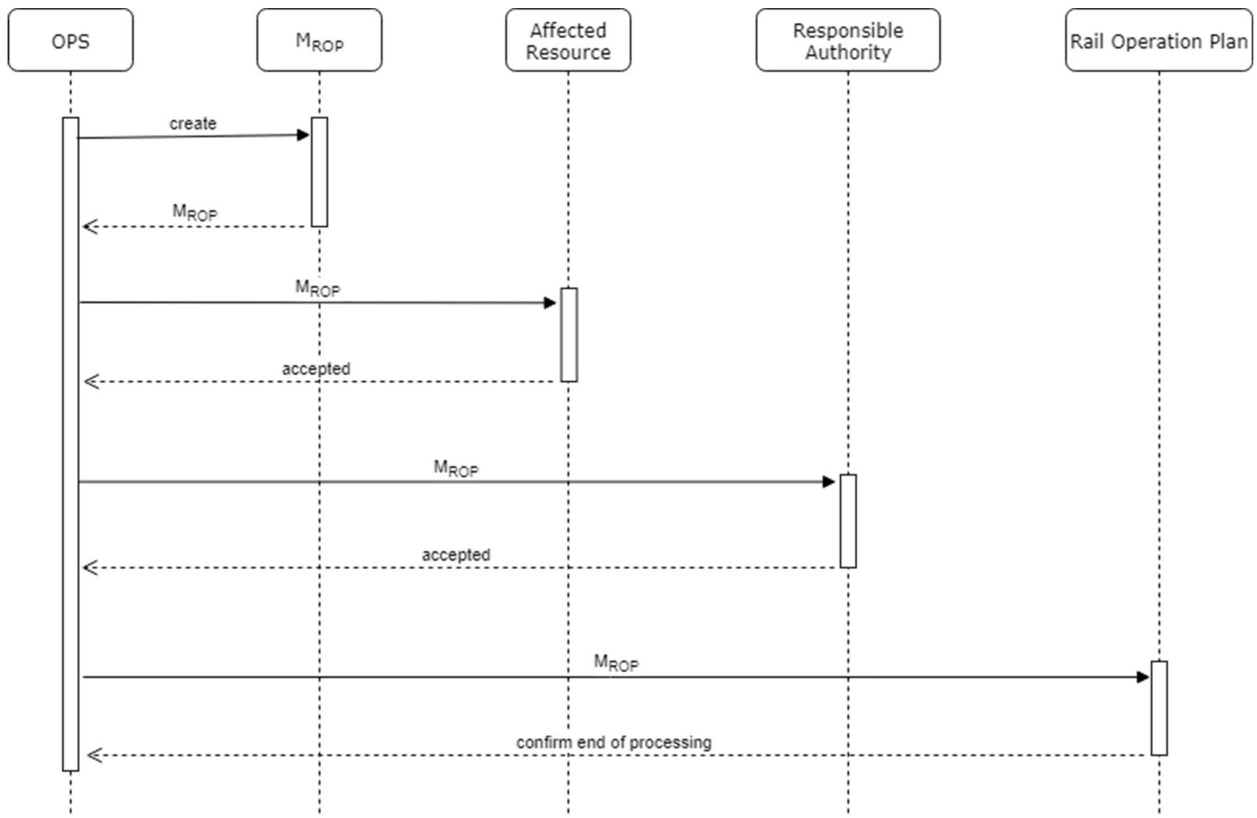

Figure 2: A description of the modification request protocol with one responsible authority, one affected resource, and one modification request, $\boldsymbol{M_{ROP}}$. In this case, the request is accepted and applied to the rail operation plan.

Applying modifications to a rail operation plan requires a certain amount of time. For example, if the engine driver A is sick and engine driver B should fill in for A, we need to schedule some time for B to get to the assigned rolling stock. Dependent on the current location of B, the required time could range from minutes to hours. Other modifications such as modification to the train journey, on the other hand, are potentially performed immediately.

Remark 1: Modifications to a rail operation plan will be typically described as a set of modification requests together with a set of attachments providing explanations for each request. As these modifications are often dependent on each other, the OPS will process the complete set of modification requests at once.

7. (Intraday) operational offline and online planning

Nowadays, operational planning is performed by several groups, each of which is responsible for one or several parts of the planning process. These groups work together in close collaboration to come up with an initial rail operation plan. Every time changes need to be made, a subset of these groups adapts the existing solution accordingly. If these changes arrive early enough, the collaboration time will not necessarily carry weight. But if they arrive on short notice, a solution will potentially be not found on time, so that rail operations are affected negatively. As this could trigger further changes that again cannot be applied on time, these negative effects could be propagated throughout the rail network.

With today's technology, it is imaginable that operational planning is conducted by computer systems that respond to changes within short reaction times. Such a system could still be divided into several sub-systems, given that the communication time among those sub-systems does not increase the reaction time above a certain threshold. If the overall time for creating, submitting, and communicating the modification request is below this threshold, we denote such an operational planning system as an online system.

Definition 6: Let *P* be an operational planning system, *ROP* be a rail operation plan, *R* be all resources that are controlled by *P*, and $t_{reaction}$ be a predefined and acceptable reaction time (e.g., 3 seconds). Then *P* is an online operational planning system, if

1. $R = ROP.nodes \cup ROP.routeSections \cup ROP.trainRoutes \cup ROP.attributes$,
2. the maximal time to access the current state of elements in R is below $t_{reaction}$, and
3. *M.submission + M.response* is below $t_{reaction}$ for any modification request *M*.

If *P* does not fulfill these requirements, it is considered offline.

Remark 2: Online operational planning has the potential to substantially improve rail operations in case of short-term changes. Nowadays, however, operational planning is mostly offline for several reasons:

1. For safety reasons, responsible authorities are in most cases humans, so that short reaction times cannot be guaranteed in general.
2. The current state of resources cannot be accessed within short reaction times, since the resources are humans, interfaces to resources are only available to humans, or sub-systems are isolated (i.e., direct data communication is not supported).
3. There are no centralized means to control resources.
4. Regulations prohibit a fully integrated system that knows and controls all relevant resources. For example, staff must not be tracked and managed centrally, because this data could be misused for surveillance purposes.
5. The resources are owned by different companies, which are possibly competitors.

In this deliverable, the focus lies on intraday operational planning, that means planning processes that happen shortly before or within the operation day. In general, a detailed rail operation plan is already available and intraday planning only leads to minor modifications or extensions to include additional demand for train journeys or to modify existing train journeys due to larger disruptions or the like.

Definition 7: A modification request M_{ROP} is called intraday, if M_{ROP}*.submission* + M_{ROP}*.response* + M_{ROP}*.processing* preceeds the start of the operation day. Intraday operation planning addresses all modification requests that are intraday.

8. Relation to Traffic Management Systems

Operational planning as described in the previous sections is strongly related to traffic management systems, which are commonly understood as systems that detect potential train conflicts and resolve them in real-time. Nowadays, traffic management systems usually require human input and only focus on train routes and infrastructure elements. Projects developing future traffic management systems, on the other hand, have online systems in mind that assume a broader view on rail operations, where information on other resources such as rolling stock, staff, or yard and terminal operation is also included. Research in future traffic management systems and operational planning systems in this area with published results are IP2 projects In2Rail and X2Rail-2 and IP5 projects ARCC and FR8Hub[1] [1].

A clear distinction between planning systems and traffic managements is difficult, probably also depending on the corresponding company or research community. Whereas operational planning happening before the operation day could be easily assigned to an operational planning system, there is no clear distinction between an intraday operational planning system and a TMS. In parts, this is only a wording problem, but restrictions imposed by laws could also enforce a clear separation. For example, resources such as rolling stock and staff is typically the responsibility of railway undertaking companies and therefore outside the scope of a TMS. For cases like this, it is imaginable that the TMS collaborates with an intraday operational planning system, where the TMS provides the current state and awaits a solution by the planning system. This way, the system can figure out solutions for emerging problems minutes or even hours in advance (e.g., in case of bad weather and other external influences).

[1] Shift2Rail projects such as In2Rail (http://www.in2rail.eu/), X2Rail-2 (https://shift2rail.org/project/x2rail-2/), ARCC (https://shift2rail.org/project/ARCC), or FR8Hub (FR8Hub https://shift2rail.org/project/FR8Hub).

9. Use Cases

This section describes possible use cases for the interplay between an operational planning system and macroscopic simulation tool, where the focus lies on intraday operational planning.

9.1. UC 1: Manage real time traffic

The main purpose of intraday railway operation is monitoring and modifying train movements. Due to unpredictable events (e.g., disruptions such as signal failures), deviations from the plan are inevitable and ad-hoc modifications to the affected resource plan become necessary to maintain stable operations (dispatching). In principal, dispatching decisions could target all types of resources, but, in this section, we mainly focus on train journeys. Moreover, we additionally distinguish between low- and high-level dispatching: low-level dispatching aims at solving train conflict among trains that use the same infrastructure at the same time, whereas high-level dispatching aims at ensuring the quality of the system. In the following we describe how dispatching decisions are performed for both levels today and how an OPS can be used to improve dispatching decisions.

9.1.1 Low-level dispatching

Low-level dispatching can be divided into two parts: conflict detection and conflict resolution. Nowadays, conflict detection is based on forecasting train journeys up to 30 minutes into the future. The conflict resolution is performed by staff members of the infrastructure manager. They perform dispatching decisions such as increasing the dwell time of a train (e.g., waiting for an overtaking of a delayed following train) or changing the train route and instructing an overtaking.

For large railway networks, a dispatcher controls a small subset of the network to make sure that the number of conflicts and the amount of information is manageable. The information available to the dispatcher, besides the forecast of the train routes (i.e., parts of a train journey), is a list of current disturbances. On this basis, the dispatcher has a short time window for decisions, which are mostly based on the dispatchers knowledge and experience, e.g., about connecting trains and knowing local features of the monitored region. Since the information given to the dispatcher are specific to a local region and resulting secondary (tertiary etc.) effects are not easy to estimate, it can easily happen that some dispatching decisions are beneficial on a local scale but lead to problems on a larger scale.

There are already systems[2] that try to optimize the dispatching process (local dispatching optimizer) in real-time. An implementation of such a system was build in the research project Co2Reopt, where a real-time decision support demonstrator was developed. The demonstrator was evaluated by the traffic control center in Narvik[3]. The main problem of these optimization systems is that the (mathematical) parameter space grows with longer look-ahead times and the size of the monitored region. Thus, it is computationally very expensive to make optimization on large-scale microscopic infrastructures. The parameter space is further increased when follow-up conflicts of

[2] https://www.thalesgroup.com/en/countries/europe/germany/transportation/traffic-management-systems

[3] https://www.co2reopt.eu

affected trains are considered. At the moment, the size of the networks on which these systems operate corresponds to the size of the monitored network. The look-ahead time equals that of a dispatcher but is planned to be further increased in the future. Furthermore, secondary and tertiary follow-up conflicts are included in the optimization. The output are suggestions for train routes shown to the dispatcher.

Currently, one problem is the definition of the target value of the optimization. For example, the system could propose conflict solutions, so that the overall punctuality is optimized or that the CO2 emission is lowest. Today, these decisions are implicit at best. However, if implemented as part of an OPS, corresponding preferences need to be formulated.

In the following, different variants of relevant parts of an OPS are discussed that use local dispatching optimizers:

Variant 1: The local dispatching optimizer detects a conflict and calculates one or more conflict solutions. The solutions (dispatching suggestions) are sent to a tool, capable to estimate the effect of different dispatching decisions on the quality of a large-scale railway network. Here, we consider that this part is done with PRISM, but, in principle, other tools capable of predicting the quality for an operation day can be used. The output of PRISM, together with the suggestions of the local dispatching optimizer, creates a modification request, which is further processed as described in Section 6. Thereby, the responsible authority is the dispatcher. Figure 3 shows an exemplary sequence diagram for such a system.

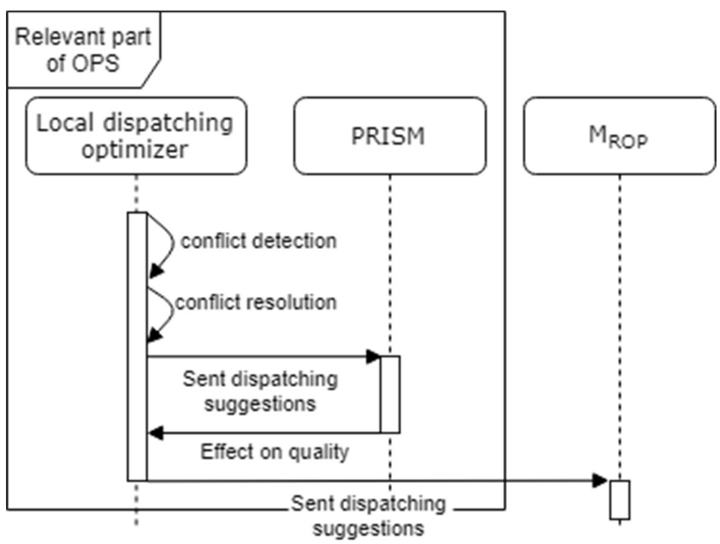

Figure 3: An exemplary sequence diagram for the relevant part of an OPS to process low-level dispatching decisions. In this variant PRISM is used once to enrich the suggested dispatching decisions with the effect on the quality of the railway system before modification requests are sent.

When there are, for example, three conflict resolutions that lead to similar results according to the dispatching optimizer, PRISM could estimate the impact on the quality (e.g., punctuality) of all three decisions on a netwide scale for the rest of the day of operation. This can then also be used by the dispatcher to decide which of the three solutions to use. A consequence of such a layout is that the suggestions of the local dispatching optimizer could be different from the output of PRISM, see variant 2.

Variant 2: This variant is very similar to the first variant with the difference that the output of PRISM is directly used as in input for the dispatching optimizer, which calculates new optimized dispatching solutions that are again sent to PRISM. This process is repeated until the benefit in the quality is below a certain threshold between two iterations or the time constrains are met. An exemplary sequence diagram of this variant is shown in Figure 4.

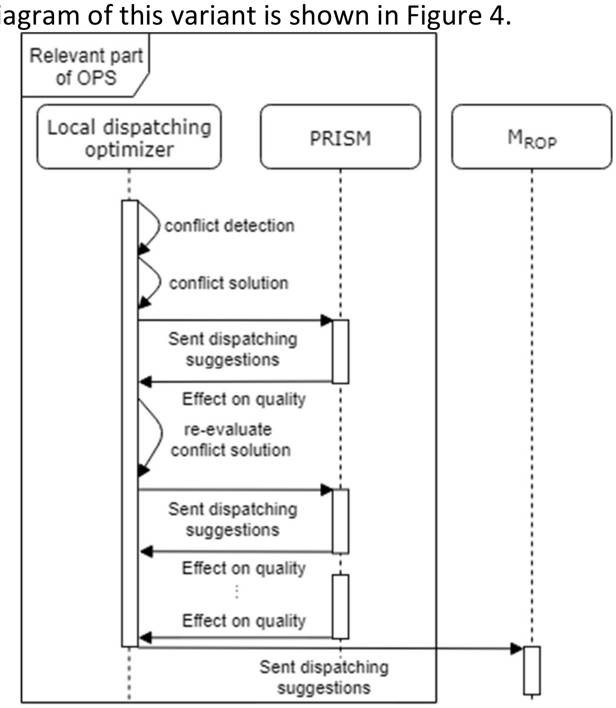

Figure 4: An exemplary sequence diagram for the relevant part of an OPS to process low-level dispatching decisions. In this variant an iterative approach is used to optimize the initially suggested dispatching decisions of the local dispatching optimizer in terms of quality of the railway system before modification requests are sent.

9.1.2 High-level dispatching

It is also possible to make dispatching decisions that have larger impact on the operation. These are:

- Cancelling/adding of complete or partial train journeys
- New train path arrival and/or departure time for a train before starting the trip

- Joining/splitting of trains
- Large-scale re-routings so that planned stops are removed/added.

Such larger scale dispatching decisions are mostly initiated by the corresponding RU and confirmed or declined by the IM.

The following example can be considered for such high-level dispatching: A train gets a (large) amount of delay due to a disturbance, so that it falls back to the time slot of the following train of the same line or a train with the same (or similar) remaining train journey. A high-level dispatching decision could be cancelling the train journey of the delayed train. The passengers must switch to other trains with the same or similar route. The benefit is that the delayed train can then be used for other journeys and the following trains are not influenced by the delayed one.

Generally, these decisions should be done at least one or two hours before the effect of the decision takes place, depending on the affected trains and journeys. This large time window is needed to inform the passengers and/or the connection trains to wait, if necessary. Due to this larger time window and the corresponding larger network to monitor, these decisions are often done by persons of the RU, monitoring the train movements of the corresponding trains. The larger time and location window makes it difficult for a local dispatching optimizer to propose optimized conflict solutions even if the reaction time to propose these decisions can be up to 3 minutes.

In an OPS variant including PRSIM, it is possible to see the effect of a possible dispatching decision on the netwide quality before it is realized. Figure 5 shows an exemplary sequence diagram for this variant of an OPS.

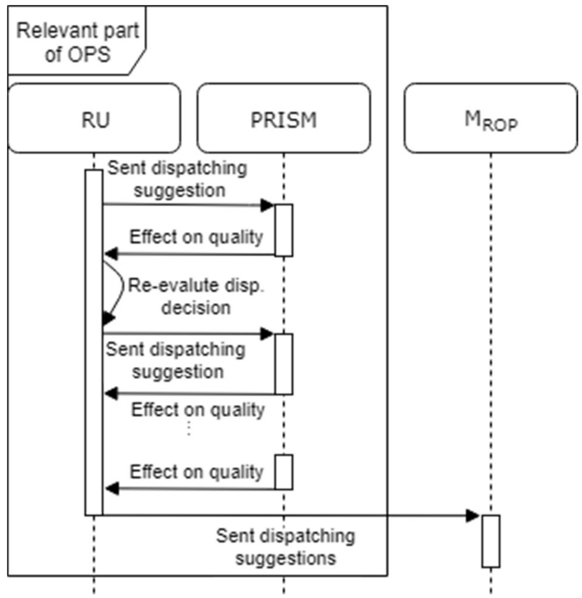

Figure 5: An exemplary sequence diagram for the relevant part of an OPS to process high-level dispatching decisions. In this variant an iterative approach is used to optimize the initially suggested dispatching decisions of the RU in terms of quality of the railway system before modification requests are sent.

This way, the dispatcher of the RU could weigh different conflict solutions against each other before deciding. Authorities that are involved in the process of the modification requests are the dispatcher of the IM.

Requirements on PRISM, that need to be fulfilled for the before described variants of an OPS are explained in Section 10.

9.2. UC 2: Manage temporary traffic restrictions

There are situations where parts of the railway network or trains are disrupted. Smaller disturbances can be controlled as described in the use case before. Therefore, here we focus on larger disruptions, which could happen due to various reasons. Examples for those temporary restrictions are:

- Closing of tracks due to damages (e.g., tree falling on a track, fire on embankments, etc.)
- Failing of a signal tower or overhead of catenary
- Weather conditions leading to a reduced velocity of trains for safety reasons
- Unplanned maintenance that require track closure/reduced speed

In case of such events, it is often not possible to stick to the timetable and one has to switch to a different operation mode (e.g. disruption management). In this operation mode, the focus lies on finding a way of getting the passengers around the traffic restriction or out of the “affected part” of the network.

Today, disruption management is completely done by people of the IM and RU. Dependent on the scale of the disruption many dispatching decisions (low-level and high-level) might be necessary which cannot be handled by persons in the needed time and therefore delay can pile up. An OPS that includes all real-time temporary traffic restriction can be used to speed up the process of dispatching, on the one hand, and, on the other hand, it can be used to find more optimized dispatching decisions. A possible variant of the relevant part of the OPS is in principle the same as described in Section 9.1.2.

In the following, an example scenario is described where a track between two stations gets damaged. As soon as it is clear that the track is damaged the IM and RUs are informed and the local dispatching optimizer together with PRISM must include the restriction. The local dispatching optimizer finds optimal dispatching decisions for affected trains in the neighboring stations to get as much trains as possible through the affected route section (e.g., using the opposite track). PRISM can be involved in this process as shown in Figure 3. At the same time, the RU can start to find high-level dispatching decisions (e.g., large-scale re-routings) for trains that are planned to drive over the affected track using the process shown in Figure 4.

The benefits of an OPS are, that the local dispatching optimizer can handle the dispatching decisions faster than people and PRISM gives feedback about the impact of the disruption on a netwide scale and shows effects of different dispatching suggestions.

9.3. UC 3: Manage resource plan for rolling stock and staff

The planning of rolling stock and staff is done by the corresponding RUs separately. Often it is a process that starts long before the day of operation with different levels of detail, depending on the available information at the planning time. There are optimization tools, as described in [2] and [3], which can be used in the planning process but those do not include any infrastructure maintenance or disturbance/disruption information. Often there are short-term changes to the rolling stock plan (e.g., through delays in train maintenance) and/or the staff plan (e.g., due to sickness) that must be accounted for, shortly before the day of operation.

Today, all short-term requests are incorporated into the resource plan of the operation day by persons of the corresponding RU without knowing the impact of the changes on the quality of the system. This process can be improved using following setups of an OPS:

Variant 1: The optimization tools incorporate the current state of the infrastructure (e.g., speed restrictions due to maintenance) and trains (e.g., speed restriction on a certain vehicle due to a technical defect). Then it builds an optimized plan of the resource for the day. Short-term change requests are then processed by using the optimization tools with changed input. The outcome is then processed according to the protocol defined in Section 6.

Variant 2: The optimization tool builds one or more plans that are similar with respect to the utilization of the of the resource for the day. The plans are then evaluated by PRISM and the outcome in terms of quality (e.g., punctuality) is then used to decide which resource plan to use. Short-term change requests are processed by using the optimization tools with changed input from which the output is sent to PRISM for further processing. In Figure 6 an exemplary sequence diagram is shown for this variant. The benefits of using automated and optimization processes as done in an OPS, are better utilization of the resources and faster processing times of changes.

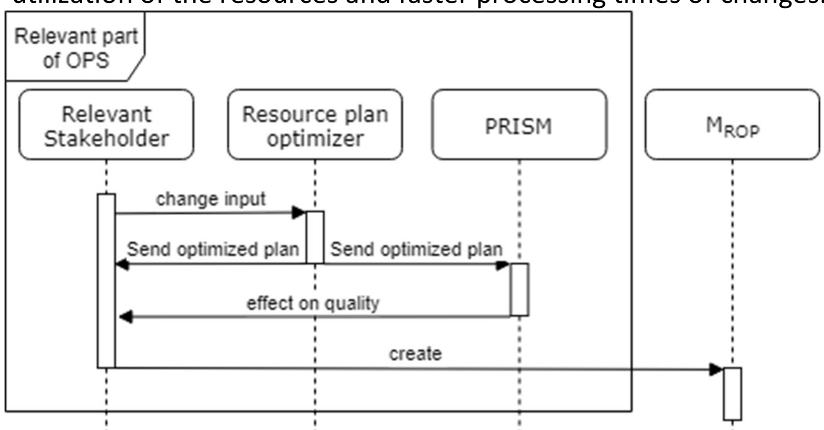

Figure 6: An exemplary sequence diagram for the relevant part of an OPS to process short term changes to resource plans. The relevant stakeholder uses a tool to get an optimized resource plan with changed input. This optimized plan is used by PRISM to estimate the effect of the changes on the quality of the system. Based on this information the stakeholder can send modification requests.

9.4. UC 4: Manage short-term request

It is possible that very short-term requests for modifications of train journeys are done in the beginning of an operation day. The modifications are like the ones listed in Section 9.1.2. Today those requests are either done manually by staff members of the RU or automatically by tools that find optimal routes for train journeys.

The methods to handle these requests are very similar to the handling of high-level dispatching decisions described before. The only difference is that the operations are not done online but offline, affecting the rail operation plan of the day. Therefore, the layout of an OPS is very similar and not discussed in more detail, here. In summary, PRISM can be used to estimate the condition of the railway system over time for the day of operation before a modification request is sent.

9.5. UC 5: Manage infrastructure maintenance

Planning of infrastructure maintenance (e.g., track renewal, bridge maintenance, etc.) is a process that expands over several periods of time. Dependent on the size, location and duration of a maintenance work, it is necessary to change the timetable. For larger construction measures the (half-) annual timetable is changed. For middle sized maintenance work, it is also possible to notify the passengers of timetable changes via posters or warnings on the platforms as well as in media when booking tickets. In addition, there are small maintenance works that are planned rather shortly before they start or requests to modify the maintenance plan for an operation day. Today, these modifications are often accepted or declined based on the experience of the people responsible, without knowing how the maintenance work could affect the rail operation in detail.

An OPS could be used to evaluate the effect on the quality before modifications to the plan are done. In Figure 7, a basic sequence diagram is shown for the relevant part of the OPS which is described in the following.

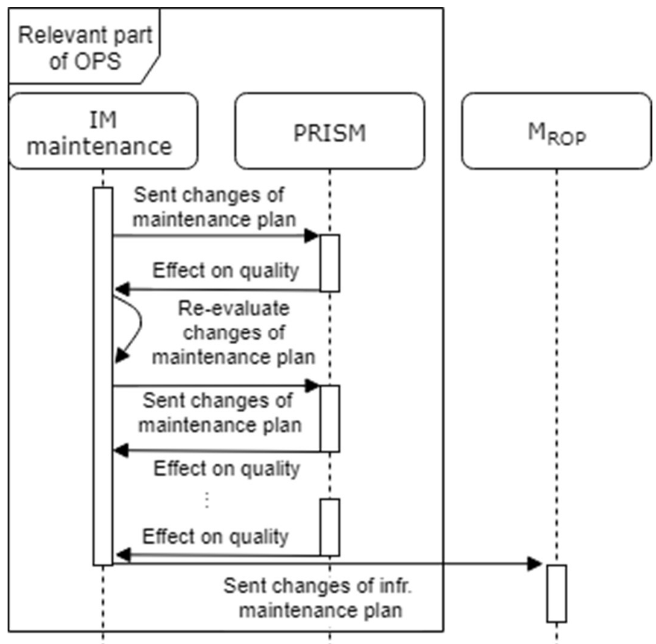

Figure 7: An exemplary sequence diagram for the relevant part of an OPS to process infrastructure maintenance. Here, an iterative approach is used to optimize the initially suggested changes of the plan proposed by the IM maintenance in terms of quality of the railway system before modification requests are sent.

The changes of the infrastructure maintenance plan are sent to PRISM, which returns the impact of the changes in terms of quality (e.g., punctuality) for the operation day. Based on this, the staff member (or tool) can re-evaluate the suggested changes and send it to PRSIM again. This process can be repeated until a satisfying suggestion of changes is reached. This modification request is then sent to the responsible authorities which is in this case the maintenance department of the IM, following the protocol described in Section 6. Then, they decide to accept the request or not. Examples of modification requests are:

- Extending maintenance
- Adding new maintenances
- Cancel maintenance

The OPS should also be able to show warnings when some maintenance work planed for the operation day lead to large impacts on the quality of the system. To achieve this, the OPS must evaluate the situation periodically and send warnings to the IM who then might make request to shift or cancel maintenance work.

9.6. UC 6: Manage rolling stock maintenance plan

Rolling stock maintenance takes mostly place at special train yards where the needed tools, repair equipment and personal is stationed. Most of the vehicle maintenance can be planned before the day of operation, since there are regulations when and how often vehicles must be maintained. But there might be changes to that plan that come in very short before the day of operation or even at the day of operation due to delays during maintenance or other trains needing more urgent repairs.

Today, this intraday planning is mostly done by persons of the RUs fleet management that communicate with the train maintenance regarding the priority which vehicle to process first. Most often, there is only little knowledge about the consequences of these decisions on the quality of the railway operation (e.g., punctuality development over the day of operation). This process can be optimized using an OPS. An exemplary sequence diagram of the relevant parts of the OPS is shown in Figure 8 and explained in the following.

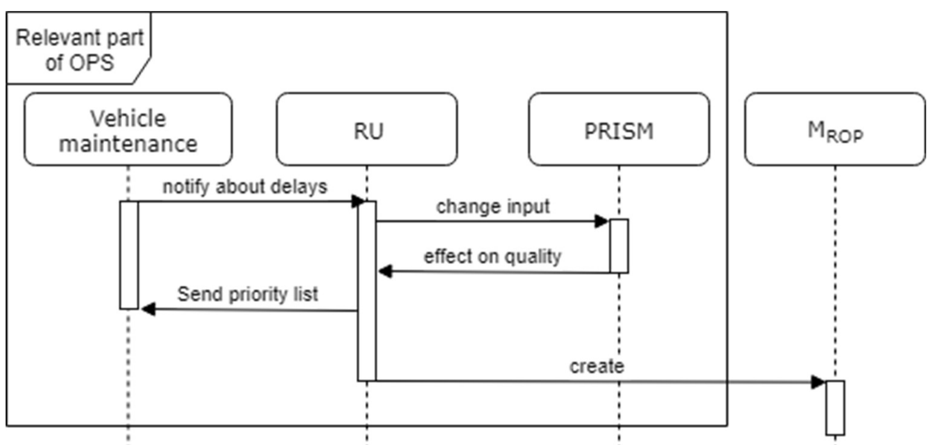

Figure 8: An exemplary sequence diagram for the relevant part of an OPS to process short term changes to the rolling stock maintenance plan. Before a change request ($\boldsymbol{M_{ROP}}$) is sent, the impact of the changes (e.g., delays during vehicle maintenance) on the quality are evaluated with PRISM.

The vehicle maintenance informs the RU about changes in the current maintenance plan (e.g., due to some delays). The RU can then modify available vehicles in the input data for PRISM and receives the effect on the quality for different scenarios. Based on this the RU sends a priority list which train to process first to the vehicle maintenance. In addition, a modification process is created to change the plan of rolling stock following the protocol defined in Section 6.

Another variant of the relevant part of the OPS is shown in Figure 9.

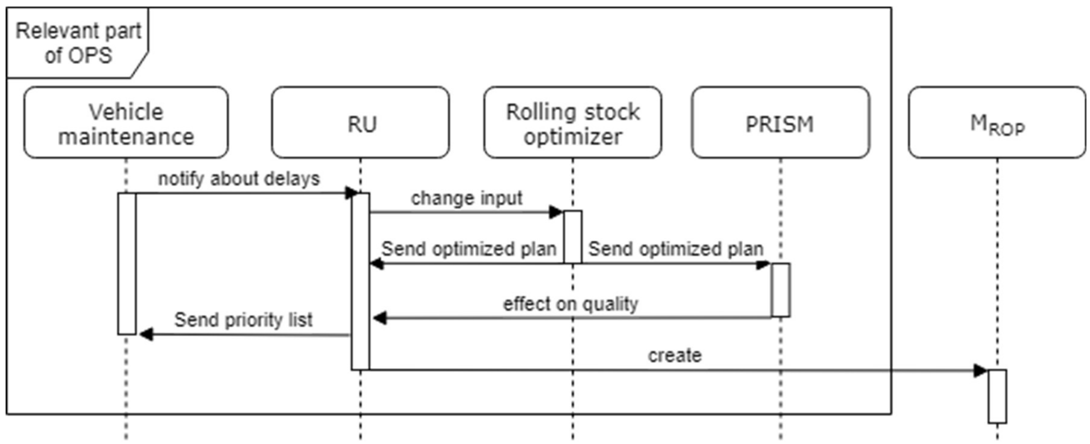

Figure 9: An exemplary sequence diagram for the relevant part of an OPS to process short term changes to the rolling stock maintenance plan. Before a change request ($\boldsymbol{M_{ROP}}$) is sent, the changes (e.g., delays during vehicle maintenance) are evaluated using a rolling stock optimization tool and PRISM.

This variant includes an optimization tool that optimizes the plan of rolling stock, already mentioned in Section 9.3.

9.7. UC 7: Communication of weather-related information

The impact of weather on railway operation can be very big, but there is no system that predicts this impact in terms of quality (e.g., punctuality), although the weather forecast uses to be rather reliable for the next 1-2 days. Even if changes to the plan are applied due to information of the weather forecast, it cannot be evaluated how beneficial the changes will be.

An OPS using PRISM would be capable to give that possibility. In Figure 10 an exemplary sequence diagram of the OPS is shown. PRISM periodically evaluates the quality of the coming days using real-time forecast weather data and sends warning to relevant stakeholders (e.g., IM manager, RUs) when the predicted quality drops below a certain threshold. For example: “Due to a storm in a region XY starting in the middle of the day, the train can only drive with reduced speed. This will reduce the overall punctuality to XY%.” The stakeholders can then modify their resource plan for the critical time period (e.g., a plan with all trains routed around the storm) and re-evaluate the effect on the quality using PRISM. This can be repeated until the needed quality or the time constrain is reached. The resulting change request for the resource plan of the operation day is then processed further using the protocol described in Section 6.

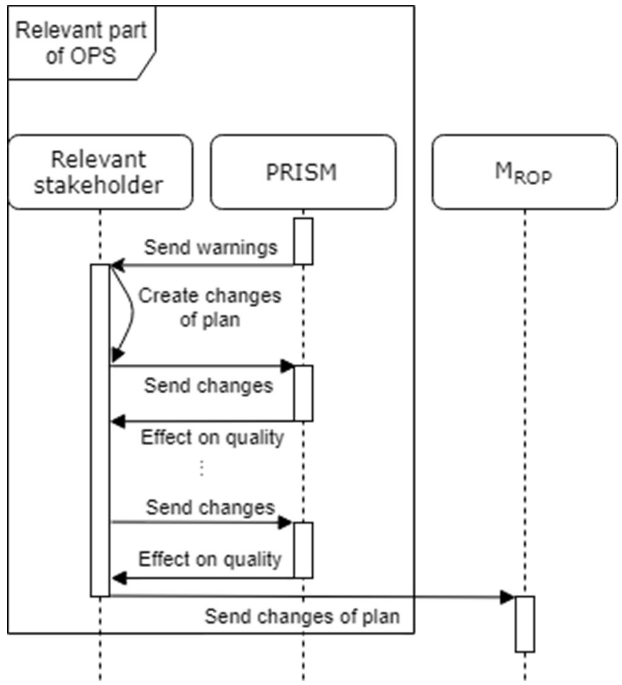

Figure 10: An exemplary sequence diagram for the relevant part of an OPS to re-plan relevant resource plans due to weather information. Before a change request ($\boldsymbol{M_{ROP}}$) is sent, the impact of the changes (e.g., trains re-routed around a storm) on the quality are evaluated with PRISM.

10. Feasibility Study

In the scenarios described by Use Case 1 through 7, rail operations would benefit from an intraday operational planning system that interacts with a macroscopic simulation tool to evaluate a set of potential modification requests. In this section, we assess which requirements such a macroscopic simulation tool needs to fulfill, which of these requirements are already fulfilled by PRISM, and which modifications are necessary.

10.1. Integration into an Operational Planning System

Since modification requests typically need to be applied within a short amount of time, e.g., within a few minutes, and often involve several resources as well as several responsible authorities, parties need direct communication channels to respond as quickly as possible. For any machine-machine communication, this requires interfaces having a low-latency and a common protocol to share information. For PRISM, we would provide such an interface in the form of a REST API (see **Fehler! Verweisquelle konnte nicht gefunden werden.** for an illustration), which facilitates the use in heterogeneous IT environments and is, therefore, a good choice to easily integrate existing tools. Communication channels involving humans, on the other hand, should be provided by systems connected to the operational planning system. This way, it is easier to design user interfaces that meet the individual requirements of the various user groups or facilitates the use of already existing tools, for which users possibly received intensive training or got used to over the years.

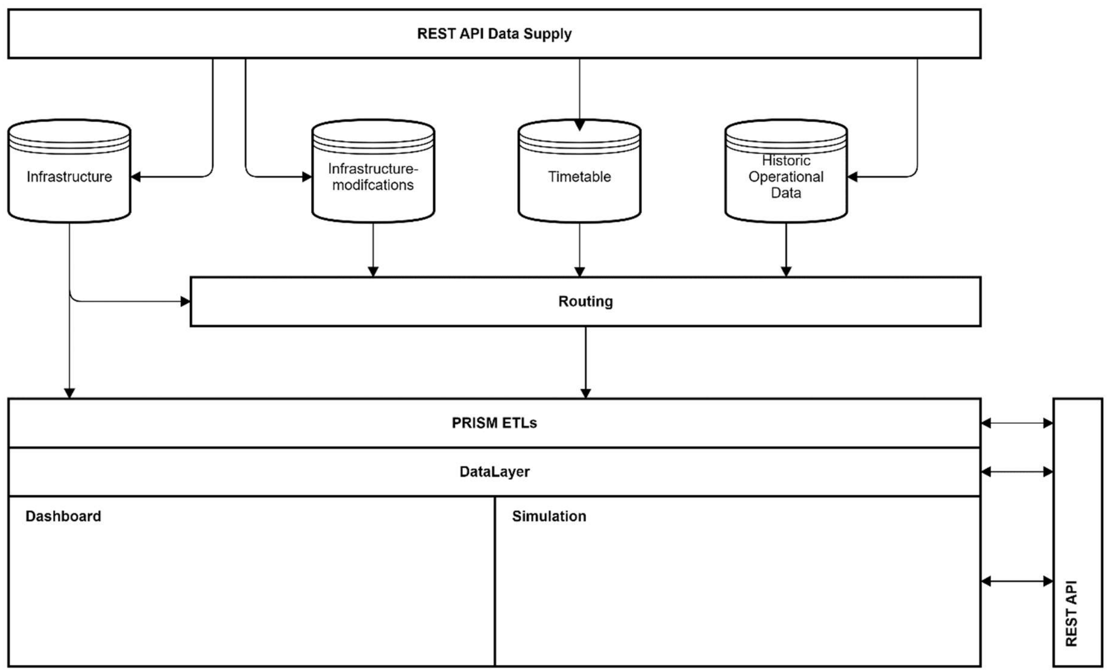

Figure 11 To communicate with operational planning systems, a macroscopic simulation could offer REST APIs to provide information, to trigger simulations, and to access information. This diagram shows a possible architecture for PRISM, which provides a REST API to accept input data and one REST API to trigger simulations and requests its results. The input data is routed to the imported infrastructure (by the component Routing), transformed into PRISM's internal data format (by the component PRISM ETLs), and provided to the simulation core and analysis dashboard by the data abstraction layer (DataLayer).

For existing tools, it could be necessary to write a wrapper that provides an interface to the operational planning system. In any case, the user interfaces should be non-distractive and easy-to-use, so that the overall process is not decelerated substantially by the involvement of humans.

10.1. Input

In its current state, PRISM has an explicit model for the infrastructure and the timetable. Other resources such as rolling stock or staff are merely described by probability distributions from historical data that are either attached to infrastructure elements, to train elements, or to both. Hence, modifications to their resource plans are typically more difficult to assess and limited to frequently occurring situations (for situations occurring rarely, the number of observations is probably insufficient to provide the corresponding probability distribution). Since we cannot assume in general that a sufficient number of observations is available for all relevant situations (i.e., the situations that are considered by an operational planning system), we pursue an explicit modelling approach with PRISM.

provides an overview on which input data (i.e., resource plan or information about resources) is already explicitly modelled by PRISM, which input data is in the scope of PLASA, and which input data is most relevant for the use cases discussed in this deliverable.

All the input data can be provided to PRISM by using the REST API (see Subsection 10.1). In the following, the different input parameters are listed with their respective features and possible risks in collecting them.

Timetable: For all use cases, it is necessary to have the most up-to-date timetable of the current operation day. The construction of a timetable is a process that takes place in several stages with different time scales and various responsible stakeholders. There could be short term changes due to maintenance (see also UC5), for example, or short-term requests to add freight trains to the timetable for the next day. Since most railway companies developed historically and responsibilities are typically distributed among different railway undertakings and the infrastructure manager, there is a risk that modifications to the timetable are not propagated fast enough to the central timetable system. This results in outdated information, which renders intraday operational planning very difficult.

Rolling stock plan: The rolling stock plan must be provided by the respective railway undertaking, which could be problematic, since this information is possibly confidential and thus using them in an OPS might raise concerns due to competitiveness.

Staff plan: The staff stock plan is classified information of the railway undertaking. Using it in a tool severely increases the security conditions, probably even if the personal data is anonymized. Both, rolling stock and staff plan, are not yet supported by PRISM and there is no definition of the data format or interface model. This will be part of Deliverable 3.4 “Case study on resource dependencies”.

Infrastructure maintenance plan: In order to use the infrastructure maintenance plan in PRISM, one also needs the information how the maintenance will affect the operation, e.g., setting up a temporary speed restriction. In addition, the timetable must be in line with the maintenance plan. For example, if a track is planned to be closed, no trains should be planed over this section. Often this is a problem, since there could be different processes and responsible undertakings for planning maintenance and for planning the impact on trains.

Current state of resources: The current state of resources includes, for example, information on disturbances / disruptions caused by malfunctions of the infrastructure or trains. The infrastructure manager who is also responsible for the train operation holds this data. It can be transferred to PRISM using the REST API, but there is currently no data format or interface model defined to get real-time data.

Information from external partners: For UC7, weather information is needed, which can be provided by several weather services. Since weather forecast data is, in general, fluctuating substantially, it is important to have up-to-date data. The data format and the interface model has to be defined in order to include it into PRISM.

The synchronization and bookkeeping of the different data sources are also topics that needs to be addressed. It must be ensured that, for each evaluation performed by the OPS, the most up-to-date data is used, which also means that changes to input data applied after an evaluation needs to be propagated to the OPS as fast possible, e.g., using the modification request protocol (see Figure 2).

Input	Supported by PRISM	In scope of PLASA	UC 1	UC 2	UC 3	UC 4	UC 5	UC 6	UC 7
Timetable	✔	✔	✔	✔	✔	✔	✔	✔	✔
Rolling stock plan		✔			✔			✔	
Staff plan		✔			✔				
Infrastructure maintenance plan	✔	✔					✔		
Current state of resources	(✔)	(✔)	✔	✔	✔	✔	✔	✔	✔
Information from external partners									✔

Table 1: Overview of input data needed for intraday operational planning. A check mark indicates that the input data is already part of PRISM, in the scope of PLASA, or modified as part of the use cases.

10.2. Features

Some inputs that are necessary to enable Use Case 1 through 7 are currently not handled by PRISM (see Section 10.1). In the following, it is shortly described which features must be implemented to support these inputs:

- PRISM models modifications to the infrastructure such as closed tracks (e.g., track gets damaged by a tree) in the same way as it handles the infrastructure maintenance plan. Train malfunctions, on the other hand, are not modelled explicitly and will be a part of Deliverable 3.4 “Case study on resource dependencies”.

- Weather information needs to be included into disruption/disturbance handling to simulate the impact of weather. Currently, this is not supported by PRISM, but additional weather-related parameters could be handled similarly to disruption/disturbances caused by infrastructure or train malfunction.
- The inclusion of rolling stock and staff plans will be discussed in Deliverable 3.4 “Case study on resource dependencies” in detail.

Besides additional inputs, there are a few features that PRISM needs to support to model rail operations as precisely as possible and thereby also improving the overall prediction accuracy. Currently, deviations from the timetable are caused by local events (e.g., disruptions, infrastructure restrictions, or train-type specific driving behavior on edges) and a low-level dispatching algorithm that changes the sequence of trains, if appropriate. A more sophisticated dispatching algorithm (high-level dispatching, re-routing, etc.) that can handle large-scale disruptions such as strikes or that can make dispatching decisions that consider the current situation of other resources such as rolling stock or staff is not yet supported by PRISM. Partially this will be covered by Deliverable “Smart Planning: Dispatching in the context of the PLASA macro-simulation”.

For all of these features, there is a risk of not having appropriate input data to evaluate the implementation (e.g., no reliable weather data, staff data is confidential). For the low-level and high-level dispatching, there is a risk of substantially increasing the running-time due to performing deadlock detection.

10.3. Running time

To use PRISM as a component of an intraday operation planning system, certain running time requirements need to be met. The timing constraints vary for the use cases, but, in general, the time that is available for PRISM can be described as following:

$$t_{total}/n - t_{eval} - t_{buffer},$$

where t_{total} is the total available time to form a modification request, n is the number of re-evaluations (iterations), t_{eval} , the time needed by other tools/people to (re-)evaluate the modifications to the plan, and t_{buffer}, the time needed to send/receive information.

In the following, the timing constraints for the different use cases are listed, using the general description from above.

UC 1:

- Low-level: The complete chain from conflict detection to sending a change request must be very fast ($t_{total} \leq 3$ seconds). It strongly depends on the setup of the OPS how this time window is split among the local dispatching optimizer and PRISM. In case of the second variant, the available time must be divided by the number of re-evaluations, which reduces the time constraint per iteration even more. Considering an even splitting of the time among the local dispatching optimize and PRISM, it results in ($3/n - 3/2n$ – t_{buffer}) seconds available for PRISM for each iteration.

- High-level: The time available to make a high-level dispatching decision is $t_{total} \leq 3$ minutes. In the variant of the OPS presented before, this means that PRISM has a maximal time window of ($3/n$ – t_{eval} $-$ t_{buffer}) minutes per iteration.

UC 2: In case of larger disruptions, the time constraint for decisions is about 30 seconds. As in the case of high-level dispatching, the available time per iteration is ($30/n$ $-$ t_{eval} $-$ t_{buffer}) seconds.

UC 3 – 7: The use cases 3 – 7 consider offline planning and therefore time constraints (t_{total}) are less strict, e.g., up to 5 minutes to form a modification request. How the available time must be split depends on the involved tools and steps of the use case.

Also, the time needed to follow the protocol, described in Section 6, must be taken into account which might lower the total available time further.

Currently, PRISM has four simulation steps. First all necessary input, like timetable and infrastructure information for one day, is read in, which takes about 65 s. Then disturbances are sampled using distributions from historical data, which needs about 5 s. Thirdly, the train movements of the day are simulated which takes about 10 s. In the last step, the output is written, which needs about 25 s. To make statistical significant statements, one operation day should be simulated at least 10 times. This can be done in parallel and thus does not increase the runtime. In total, this results in about 105 s to make a prediction of the quality for one operation day.

Table 2 summarizes for which use case the current runtime performance of PRISM is sufficient. In this table, t_{buffer} is set to 0.5 s. The evaluation of low-level dispatching would currently not be possible. For high-level dispatching decisions, an evaluation time for the RU of 30 s is considered, which results in a maximal number of one iterations to stay within the time constraint of 3 minutes. The time constraints assumed for the second use case show that it is not possible to stay within the 30 seconds considering an evaluation time of 5 seconds. For use cases considering offline planning (UC 3-7), the time constrain of 5 minutes leads to a maximal number of two iterations.

Use case	n_{max}	t_{eval} **[s]**	t_{total} **[s]**
UC 1 low-level	0	3/(2 n_{max})	3
UC 1 high-level	1	30	180
UC 2	0	5	30
UC 3-7	2	30	300

Table 2: Overview of timing information for different use cases. $\boldsymbol{n_{max}}$ indicates how often it is possible to evaluate a scenario (e.g., modified resource plan) with PRISM, given the total available time $\boldsymbol{t_{total}}$, considering the time $\boldsymbol{t_{eval}}$ consumed by other parts (tools/people) involved in the OPS. $\boldsymbol{t_{buffer}}$ is set to 0.5 s.

Please notice that, so far, only little effort was put into optimizing the runtime performance. For instance, reading of the input data could be improved by enhancing the object model. Therefore, it is unclear, whether PRISM could be used for low-level dispatching in the future.

11. Conclusions

In the presented document, we addressed how a smart planning system can be used for intraday operational planning. At first, attributes of an operational planning system were introduced which include involved stakeholders, resources and constraints. Interactions between these attributes and further definitions of such a system were described by introducing a framework for intraday operational planning systems. Use cases were presented, where we described how an operational planning system could be designed and how it could be included in intraday planning. The use cases cover topics like real time traffic management (e.g., dispatching) and short-term re-planning of resource plans, such as plan of rolling stock, staff and maintenance. One use case focused on how an operational planning system could be designed to predict the impact of weather-related information on the quality of the railway system and how this can be used for intraday operational planning.

It was evaluated how PRISM can be used in each of the use cases and requirements on the input data, features and running time of PRISM were set. It was shown that for most use cases PRISM can be a beneficial part of intraday operational planning once resource plans (e.g., the rolling stock plan) are modeled explicitly in the simulation. Furthermore, some initial discussions have been made about how PRISM can be used for online (real-time) planning (e.g., dispatching).

12. References

[1] R. Domberger, L. Frey and T. Hanne, “Single and multiobjective optimization of the train staff planning problem using genetic algorithms,” in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, 2008.

[2] Y.-C. Lai, D.-C. Fan and K.-L. Huang, “Optimizing rolling stock assignment and maintenance plan for passenger railway operations,” *Computers & Industrial Engineering,* vol. 85, pp. 284-295, 2015.

[3] M. Mazzarello and E. Ottaviani, “A traffic management system for real-time traffic optimisation in railways,” *Transportation Research Part B: Methodological,* vol. 41, pp. 246-274, 2007.