IN2DREAMS

Contract No. 777596

INtelligent solutions 2ward the Development of Railway Energy and Asset Management Systems in Europe

D5.4: Rule-Based and Visual Analytics Knowledge Extraction Demonstrator

DUE DATE OF DELIVERABLE: 31/08/2019

ACTUAL SUBMISSION DATE: 28/08/2019

Leader/Responsible of this Deliverable: Luca Oneto - UNIGE

Reviewed: Y

Document status		
Revision	Date	Description
0.1	02/04/2019	UNIGE completed the first issue of the TOC
0.2	28/05/2019	UNIGE completed the Executive Summary
0.3	29/05/2019	UNIGE completed all its parts in Sections 1, 2, 3, 4, and 5
0.4	30/05/2019	Cefriel inserted the contributes related to section 3 and 4
0.5	20/06/2019	Cefriel, UKON and UNIGE inserted the joint contributes related to section 4 and 5
0.6	15/07/2019	Cefriel and UKON revised and finalized contributes 4 and 5
0.6	15/07/2019	UNIGE revised the finalized document
0.7	26/07/2019	UNIGE, Cefriel, and UKON finalized the document
1.0	27/08/2019	Final Version afer TMT approval and Quality Check

Project funded from the European Union's Horizon 2020 research and innovation programme		
Dissemination Level		
PU	Public	X
CO	Confidential, restricted under conditions set out in Model Grant Agreement	
CI	Classified, information as referred to in Commission Decision 2001/844/EC	

Start date of project: 01/09/2017 - Duration: 26 Months

Executive Summary

This deliverable will describe the demonstrators that have been developed to show the potentialities of the results of Tasks 5.2, 5.3, and 5.4.
Tasks 5.2 focused on the study, design and development of metrics and tools for the assessment of data analytics algorithms in the railway context, especially for infrastructure asset management. Task 5.3 focused on developing and testing data analytics approaches for extracting information and knowledge from railway asset data in a visual or rule-based manner. Task 5.4 exploited state-of-the-art rule-based and visual analytics approaches to enhance the capability of understanding information content included in data by either visualizing it directly, or rendering in a pictorial form the structure and consequently the output of data-driven models.
During the project, as one can retrieve from D5.1, D5.2, and D5.3, we have developed seven scenarios:

- Cross-Scenario 1: Visualizations in the Control Center;
- Cross-Scenario 2: Marketplace of Data and Data Monetization;
- Specific-Scenario 1: Track Circuits;
- Specific-Scenario 2: Train Delays and Penalties;
- Specific-Scenario 3: Restoration Time;
- Specific-Scenario 4: Switches;
- Specific-Scenario 5: Train Energy Consumption.

The scenarios focus on relevant railway assets whose malfunction and maintenance policies have an impact on the KPIs targeted by the SHIFT2RAIL program. The cross-scenarios cover many aspects of the railway ecosystem while the five specific-scenarios focus on a single particular aspect.
In D5.2, Specific-Scenario 3 has been selected as the subject of the final proof of concept since it allows to connect the two WPs (WP4 and WP5) of the WS2. In particular, this deliverable will describe the demonstrators that have been developed for the scenario on Restoration Time which exploits three modules:

- the proof-of-concept developed on WP4 for handling the maintenance process as reliable data source thanks to the Distributed Ledger Technologies;
- the data analytics models developed in WP5 for Specific-Scenario 3 for predicting the restoration time from each maintenance as important information to provide to the train operators and the infrastructure managers;
- the visual analytics models developed in WP5 for Cross-Scenario 1 for displaying the information to the operators coming both from the Distributed Ledger Technologies and the data analytics models.

Abbreviations and Acronyms

Abbreviation	Description
AMS	Asset Management System
CEFRIEL	Cefriel (IN2DREAMS WS2 Partner)
DLT	Distributed Ledger Technology
EU	European Union
EVOLUTION ENERGIE	Evolution Energie (IN2DREAMS Partner)
IM	Infrastructure Manager
IP	Innovation Programme
KPI	Key Performance Indicator
KUL	Katholieke Universiteit Leuven (IN2DREAMS WS2 Partner)
POC	Proof of Concept
RFI	Rete Ferroviaria Italiana (IN2DREAMS WS2 Partner)
TMS	Traffic Management System
TO	Train Operator
TRL	Technology readiness level
UKON	University of Konstanz (IN2DREAMS WS2 Partner)
UNIGE	University of Genoa (IN2DREAMS WS2 Partner)
VA	Visual Analytics
WP	Work Package
WS	Work Stream

Table of Contents

List of Figures

List of Tables

1 Introduction

Data and Visual Analytics Technologies (DVATs) and Distributed Ledger Technologies (DLTs) can bring disruptive innovation in the way we handle, store, and process data to gain knowledge. In this work, we describe the architecture of the final WS2 Proof Of Concept (POC) that will be a system that leverages on both these technologies to better manage maintenance actions in the railways context. On one side we employ a permissioned DLT to ensure the complete transparency and auditability of the process, the integrity and availability of the inserted data and, most of all, the non-repudiation of the actions performed by each participant in the maintenance management process (see the deliverables of WP4). On the other side, exploiting the availability of the data in a single repository (the ledger) and with a standardised format, thanks to the utilisation of a DLT, we adopt DVATs to leverage on the features of each maintenance job, together with external factors, to estimate the maintenance restoration time (see the deliverable of WP5, with particular reference to Specific-Scenario 3: Restoration Time). All these crucial information will be displayed to the different operators, exploiting the tools developed in the Cross-Scenario 1: Visualizations in the Control Center.
The deliverable is organized as follows. Section 2 recalls the scope, the purpose, the organization and the description of the POC (more details can be found in previous deliverables of WP5 and WP4). Section 3 describes the POC software in details with the procedure for installing it. Section 4 shows a demo of the POC thanks to the use of a series of screenshots. A video of the demo is uploaded together with the deliverable. Section 5 concludes the deliverable.

2 The Demonstrator

Railway Infrastructure Managers (IMs) are responsible of operating the existing rail infrastructures. Maintenance is surely one of the main tasks of this job [3, 6]: not properly maintained infrastructures are in fact more prone to failures, that in turn translates into disruptions of the normal execution of railway operations. Maintenance operations are usually demanded to external contractors through framework contracts that guarantee the availability of specialized workers whenever there is a need, planned in advance or unexpected. Considering in particular the planned maintenance operations, the actual work is scheduled on specific time slots, where the train circulations can be modified or suspended without causing major disruptions. An empirical estimation of the time needed to perform the jobs is used to plan the scheduling of all the operations on the available time slots.
Moreover, to ensure that each maintenance job is performed in the correct way, that all safety measures are put in place, and all responsibilities are clearly identified, IMs employ standardised procedures to guarantee that each action is executed in a proper order by the responsible actor, leaving a trace of that execution. To fulfill such requirements, each step in these procedures must be performed leaving a legally valid record of which actions were performed, by whom, and with which authorisations; thus leading to a lot of signed works sent between the various actors via registered letters, and to recorded phone calls. A process involving work documents can be inefficient, leading to increased waiting times between each step in the workflow and the introduction of a lot of wastes due to duplication of information and verification checks. Also having a direct access to maintenance data, to assess the current status of a specific job or to perform data analysis [7, 10] may not be straightforward.
In this context, DVATs and DLTs may bring a great benefit to the current management of maintenance jobs. From one side, the adoption of DLTs and smart contracts could enable the digitalization of the process currently employed maintaining all the required features, like a tamper-proof record for the tracking of all decisions and executed actions [2], and potentially allowing the automated enforcement of contractual clauses. From the other side, the analysis of historical data about previous maintenance operations could enable the

development of a prediction algorithm able to accurately estimate the restoration time for each maintenance job, thus leading to a better planning of the operations. Moreover, the DLT enacts the gathering of all the data on a single repository (the ledger) and with a standardised format, allowing the periodic retraining of the prediction engine: such operation could hardly be done with data stored in isolated silos with a interoperable format.

For this reason in this work we propose an architecture able to merge DLTs, to automate the railways maintenance workflows, and DVATs for improving the decision of IMs in executing railway operations. For the scope of the project, DVATs and DLTs are applied to processes and data as they are and the objectives are to make it possible to introduce advances without rethinking processes themselves. Anyway in future, the impact of DVATs and DLTs, as proved and resulted in the project, will impact the ecosystem and allow a complete and deep digital transformation of the asset maintenance.

2.1 DLT-based Maintenance Management

The described system comprises two major components: a DLT system composed by a peer-to-peer network and a smart contract infrastructure and a prediction engine (see WP4 deliverables). The DLT network is implemented using private blockchain Hyperledger Fabric version 1.3[1]: a open source permissioned DLT engine, where only authorised peers may join the network [5]. The selection was conducted comparing the currently available solutions using a scoring model derived from the requirements of IMs; in the selection, parameters were defined in order to take into account both technical and business aspects. For the technical aspects, parameters included the data model, the governance, the privacy, the scalability and the overall maturity of the blockchain frameworks. For the business, in particular we referred to the Italian Railway Network handled by Rete Ferroviaria Italiana (RFI) which defined a list of requirements. For details about the selection of the platform, it is possibile to see the Deliverable 4.2 D4.2 A smart contract reference for Railways Ecosystems.

Figure 1 shows the logical architecture of the network, based on Hyperledger Fabric components [1]. Each organization participating in the maintenance operations management scenario has its own peers and a Certification Authority (CA). Each CA acts as a Membership Service Provider (MSP) for its own organization, and provides digital certificates to its related peers. The network is globally administrated by the IMs in its role as ecosystem leader. All the organizations instead share the membership service of the dedicated logical channel where all the peers are connected, allowing each of them to add their own peers to the channel. Both the ledger and the chaincodes (smart contracts in Hyperledger Fabric) are replicated on every peer connected to the channel, providing redundancy of the data. The ordering service will be provided by a single orderer on a first testing phase, to be later developed to a crash tolerant Kafka cluster [1] on a later stage.

In our scenario, confidentiality of the data is not a critical requirement as the IM has a strong interest that all the process is transparent within all the authorized network participant. Nevertheless, Hyperledger Fabric v.1.3 enables the definition of Private Data within the same channel, to ensure that confidential data between two parties are not shared with the other participants: this is done using private databases separated from the main ledger, only broadcasting hashes of the private transactions on the main ledger. In addition, the network could easily support other scenarios and applications via the creation of different channels between the organizations, ensuring the complete separation of the respective ledgers.

The Data and Visual Analytics module engine is built as a separate component deployed outside the DLT network as it is not possible to implement it as a chaincode inside Fabric: the Data and Visual Analytics module needs to be able to automatically retrain (and therefore modify) the prediction algorithm, but chaincodes can only be updated manually. The interconnection between the two components is developed through two REST APIs, as depicted in Figure 2.

[1] www.hyperledger.org/projects/fabric

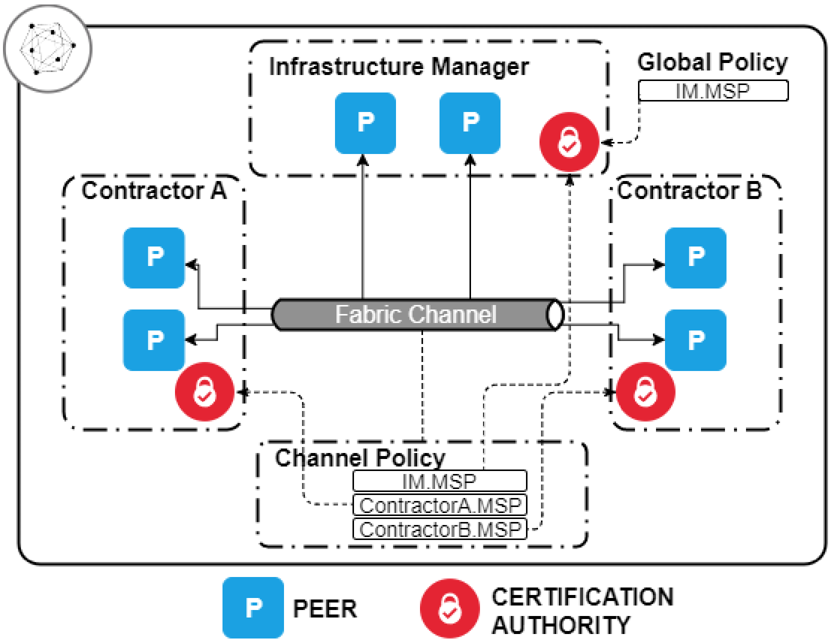

Figure 1: A logical view of the Hyperledger Fabric network: each organization has its own peers authorized by the respective Certification Authority and connected to a single Fabric channel. Please note that the number of peers is not relevant in this view.

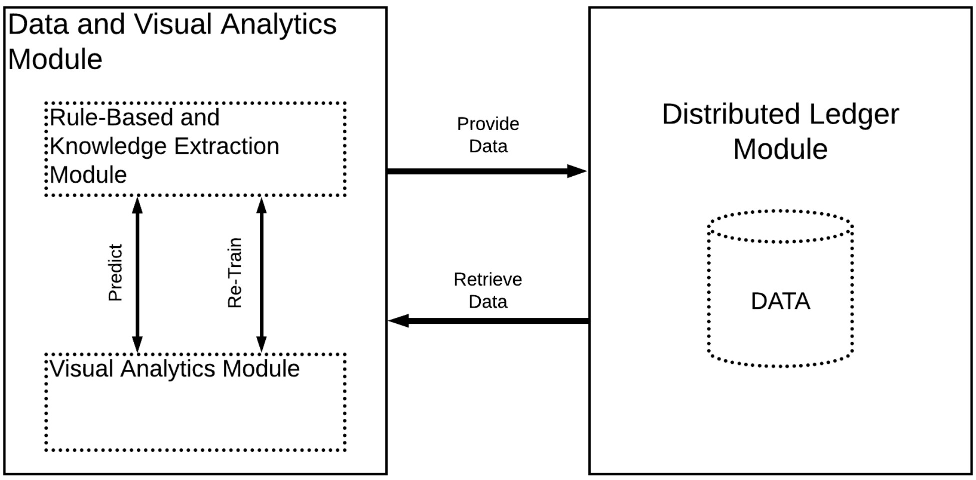

Figure 2: Interconnection between the Hyperledger Fabric network and the Prediction and Visualization Engine.

Table 1: Quality of the models.

Int.	MAE	MAPE	PCOR
RFI			
Maint.	30.5	31.5	0.75
Our Proposal			
All	11.3 ± 1.1	10.7 ± 0.9	0.93 ± 0.03
Maint.	8.1 ± 1.0	7.8 ± 0.7	0.97 ± 0.03
Fail.	15.2 ± 1.3	14.3 ± 1.1	0.88 ± 0.04

Once a maintenance operation is about to start, an operator has to officially state it by committing a transaction to the DLT, where he/she inserts all the exogenous data, like the current weather condition, needed to the Data and Visual Analytics module. The application running on the blockhani (that is the chaincode) therefore calls *Retrieve Data*, where the data are the ones required for the prediction, getting in return the estimated restoration time and the tree model used to estimate it with the function *Provide Data*. It is important to note that the prediction algorithm does not need to retrieve additional data through external sources, since it gets everything from the smart contract (chaincode). This is required to avoid non-determinism. Indeed, considering that the chaincode is executed by all the endorsing peers independently at potentially different times, retrieving data from external sources could change the results of each execution, since there is no control on external data, preventing the consensus from being achieved.

2.2 Data and Visual Analytics module

The Data and Visual Analytics module is in charge of estimating the time to restoration for different assets and different failures and malfunctions (see Specific-Scenario 3: Restoration Time in WP5 deliverables) and to extract an interpretable model to be visualized to the operators in the most intelligible way together with the state of the maintenance activities. The Data and Visual Analytics needs to take into account the knowledge enclosed into maintenance reports, exogenous information such as the weather conditions and the experience of the operators in order to predict the time needed to complete a maintenance action over an asset and to restore its functional status. Moreover, the model should be interpretable enough to give insights to the operators on which are the main factors influencing the restoration time, to better plan the maintenance activities. This information will help IMs to assess the availability of the network, by estimating the time at which a section block including a malfunctioning asset will become available again, and properly reschedule the train circulation.
For this purpose we have build a rule-based model which is able to exploit real maintenance historical data provided by RFI, the historical data about weather conditions and forecasts, which is publicly available from the Italian weather services, and the experience-based model currently exploited by the train operators for predicting the restoration time of planned maintenance. Then we implemented the prediction models described in previous WP5 deliverables using the R[2] framework. Table 1 reports:

- the error of the RFI model measured with the Mean Average Error (MAE), the Mean Average Percentage Error (MAPE), and the Pearson Correlation (PCOR) on the maintenance since no model for the failures is available to RFI;
- the error of the data-driven model measured with the MAE, MAPE, and PCOR on all the intervention, on the maintenance, and on the failures.

[2] https://www.r-project.org/

From Table 1 it is possible to note that the quality of our model is remarkably higher than the one of the RFI model.

2.3 Discussion

The system described in this deliverable, built upon a permissioned DLT with smart contracts and a prediction engine, permits the automated management of the highly regulated administrative workflow that each maintenance job has to deal with, while enriching it with the possibility to estimate the restoration time of each job, leading to a better planning of train disruptions. The main achievements of such system are twofold. The first one is to bring forward the digitisation of the workflow currently employed ensuring integrity and non-repudiation of every action performed inside the workflow thanks to the native features of DLTs; permitting, as a consequence, the instant retrieval of the status of each maintenance job and the complete audibility of the maintenance process (which, as explained in other previous deliverables of the work, it is related to possible different actors and some times in a context where the trust of the different actors each other is not guaranteed). The second one is to allow to better plan the maintenance operations thanks to the availability of an estimated restoration time for each job, a clear visualization of the state of the maintenance, and an understanding of the factors which influence the maintenance processes. Additionally, the system could be extended to enable the enforcement of contractual clauses (i.e. penalties for delays) via automatic execution of disputation procedures backed by evidence stored in the audit-proof ledger. In any case for a complete discussion about the effective advantages of such scenario, it is possible to read Deliverable D4.1 - The Data Transaction model in railways ecosystem.

3 Software Description

The SW related to the demonstrator is composed by the following two main modules (see Figure 2):

- the Distributed Ledger Technology Module (Section 3.1).
- the Data and Visual Analytics Module (Section 3.2) which is composed of two submodules
  - the Rule-Based and Knowledge Extraction Module (Section 3.2.1);
  - the Visual Analytics Module (Section 3.2.2).

The first, the DLT, has been described in detail in a dedicated deliverable (the guidelines and SW release are included in D4.3 PoC for a Smart Contract). For this reason, in the following it is given a short recall of the main elements, in order to make more comprehensive the present document. An integrated demo of the two modules integrated with a track path and instruction will be included too.

3.1 Distributed Ledger Technology Module

The DLT module represent the ”decentralized infrastructure onto which an asset maintenance workflow has been deployed” and consists of a complete stack made up by the network and the application layer. The core of the system is the ”ledger”, which is the database of all the asset maintenance jobs and related data: each asset maintenance is registered and identified. Each action made up by a participants in the network (the IM’s roles) is tracked and signed. The system implements a typical workflow of programmed asset maintenance, as referenced by i RFI procedures, roles and rules.

The SW is composed by the following components:

- The Application layer: the SW logic that allow the set of running self executing logic (smart contracts, called in the system ”transactions”) acting on the set of assets (the digitised maintenance jobs and reports);
- The Core Ledger, which is the decentralized database that tracks all the ”transactions” executed by the running application (the smart contracts);
- The Network, which is the peer-to-peer network of ”nodes ”which are able to maintain updated, synchronized, and secured the ledger of transactions.

The system requires also Ancillary components, necessary to make the system running, are:

- The identity system,
- The Data Filtering,
- The Maintenance Ops Chaincode,
- The Front End APIs.

The ASSET ”Work specification document” is composed by the following:

- ID WSD,
- Description,
- Location,
- StartDate,
- EndDate,
- SafetyCheck,
- Status of the document.

The document status admittedd fields are:

- Open, if the ocument is created by the IM but not assigned to any worker;
- Assigned, if the document is assigned to workers;
- Closed, if and only if Project Manager (or who is responsible) after evaluation check and select the closure;
- Reported: if a problem occurred during maintenance and a worker or any other participants reports the issue.

The ASSET ”ReportsofWorkIsDelivery” is the Report of works delivery with all of the details and the required signature and goto of the document that they will be able to start the maintenance job:

- ID ROWD
- Description
- Location
- DateTime StartDate
- DateTime EndDate
- Status documentStatus
- SignedByTD
- SignedByWD
- AssignedToWorker

The PoC for a Smart Contract is a decentralized peer to peer architecture which allows the IMs to manage the asset maintenance workflow from the first issuing (done typically by the Project Manager) to the creation of the reports (done by a Work Manager) and finally the updating of the reports all along the process, included the double check on safety done by the CCC operator. The operators might not be necessarily belonging to the same entity (namely RFI). At each stage of the process, all the roles in the IM are able to retrieve the

```
Prov,Type,Cond,BegStatInc,EndStatInc,Track,PredictedTime,Day,Mon,Hour,Rain,Temp,Sun,Wind,BegStatLat,BegStatLong,EndStatLat,EndStatLong,ActualTime
IM,S,C,E,E,D,0,1,6,21,0,25.4,0,1.8,4.441309e+01,8.887278,4.44169e+01,8.921409,11
GE,S,C,E,E,D,0,2,6,9,0,24.8,42.4,1.2,4.441309e+01,8.887278,4.44169e+01,8.921409,55
...
```

Figure 3: Snipping of the file *DataClean.csv*.

information on the ledger related to their competences. Everything is tracked and potentially transparent for everybody. The process has been taken and translated AS IS into the DLT, since the objective was to prove a technical feasibility. A second further step would obvously be, taken the advtange of the DLT, ehnance the process and its digitalization.
At each stage of the process it is possible to list and see the status of the document or we can filter them based on some features (date, location (partially done), who issued them etc), thus having a complete tracking of the process. Such an approach implies also the "forced" digitization of the asset maintenance. Moreover, a possible further implementation of the asset maintenance over DLT may also include the introduction of physical asset description like digital twins thus improving the capability of managing the infrastructure. For detailed explanation and description of components it is possible to refer to Chapter 3 and 4 of D4.3 (where it is possible to see the architecture, the description of the smart contracts and the running deployment view).

3.2 Data and Visual Analytics Module

3.2.1 Rule-Based and Knowledge Extraction Module

Rule-Based and Knowledge Extraction Module is composed of two R[3] scripts. In order to be able to run them it is necessary to install the R environment which is cross platform.
The two R scripts (*Learn.R* and *Forward.R*) work as follows

- *Learn.R*, which is the result of the selection of the optimal procedure and algorithms developed in the project and described in D5.2 and has the following input and outputs
  - INPUT: a file *DataClean.csv* as the one snipped in Figure 3 where the first line is an header and its meaning and format is described in Table 2 (for more details refer to the previous WP5 deliverables);
  - OUTPUT: a file *M.model* for storing the created model in a proprietary R format and used internally and file *M.txt* to be exploited by the Visual Analytics module of Section 3.2.2;
- *Forward.R*, which exploits the models stored in the file *M.model* to predict the restoration time of an accident and has the following input and outputs
  - INPUT: a file Input.csv in the same format of *DataClean.csv* but just with the header and one line of data without the field *ActualTime* since it needs to be predicted;
  - OUTPUT: a file Output.csv with one line and 2 numbers (restoration time prediction and uncertainty).

All the files need to be in the same folder and in order to run the scripts a system call to the R executable *Rscript* is needed, followed by the name of the script that needs to be run (*Learn.R* or *Forward.R*). In case of problems the script generates a file in the same folder names *Error.log* where a diagnosis of the problem is reported.

[3] https://www.r-project.org/

Field	Meaning	Format
Prov	Province	Text (categorical)
Type	P= Planned, S= Extraordinary Maintenance	Text (categorical)
Cond	C= Allowed, NC= Not Allowed, Rn= Renounced, R= Requested	Text (categorical)
BegStatInc	I=Included, E=Excluded	Text (categorical)
EndStatInc	I=Included, E=Excluded	Text (categorical)
Track	D= Left Track, P= Right Track, L= Entire Line	Text (categorical)
PredictedTime	Predicted time of the operators	Number
Day	Day of the week	Number
Mon	Month of the year	Number
Hour	Hour of the day	Number
Rain	millimeter of rain	Number
Temp	temperature in Celsius degree	Number
Sun	solar radiation in watt per square metre	Number
Wind	wind intensity in meter per second	Number
BegStatLat	Latitude of the beginning station	Number
BegStatLong	Longitude of the beginning station	Number
EndStatLat	Latitude of the end station	Number
EndStatLong	Longitude of the end station	Number
ActualTime	Actual Restoration time	Number

Table 2: Meaning of the header and data format of the file *DataClean.csv*.

3.2.2 Visual Analytics Module

As outlined in D5.1[4], Visual Analytics interfaces are provided for three different aspects. First, a decision support system for the resolution of rail schedule conflicts aids the operator in this important task illustrated in Section 3.2.3. Additionally, the presented visual interface allows TMS operators to explore maintenance duration predictions by integrating the rule-based knowledge extraction module as described in Section 3.2.1. Second, a component for situational awareness for maintenance management is explained in Section 3.2.4, allowing for both exploring events in an overview and hierarchically organized detailed view. The third component targets the issue of attention steering using adaptive highlighting solutions for TMS operators in large and complex overview situations shown in section 3.2.5.
Train Management Systems are not only operationally critical, but also with respect to passenger and personnel safety. Consequently, improvements to these systems can only be implemented in an incremental way, so as not to disrupt the mental model of operators with experience on a current system. As well, established systems incorporate safety-critical best practices from decades of operations and can and should not be switched for an untested, fundamentally different visual interface. We employed this rationale for all improvements and additions to the interfaces of Task 1-3 by keeping the visual representations and paradigms as far as possible while selectively adding or improving lacking functionality and optimizing visual readability and situational awareness capabilities.
During this process, we follow the principles of the *Visual Analytics*[8] approach to data analysis and visualization, which we have described in more detail already in D5.1 [4]. In summary, Visual Analytics is a process which integrates machine learning and data mining approaches with the intuition, experience and knowledge generation capabilities of human expert users. Given the sensitive nature of the tasks at hand, we employ Visual Analytics principles to identify potential for improvement and to research suitable solutions.

The Data and Visual Analytics Module is provided as a containerized-ready version. The used containerization software is docker. Therefore, docker has to be installed.[4] When the source-code is unzipped, the environment can be started with:

```
$ docker-compose up
```

Ports 4200 and 3000 must be available. After some time, the front-end can be accessed via `http://localhost:4200`. The containerized environment contains a REST API that executes the R-Scripts that are described in Section 3.2.1. A Node.js [5] server with the framework express [6] handles the routing and execution of the R-Scripts. The visualizations are embedded into an angular-based front-end [7]. The interactive visualizations are created with the help of D3.js [8].

The Data and Visual Analytics Module is deployed on a password protected server. It can be accessed via `https://in2dreams.dbvis.de/` [9].

3.2.3 Task 1: Decision Support System for Rail-Conflict Resolution

In train management scenarios, two main focuses exist: First, organizing the train traffic schedule, and second, implementing the schedule in the physical track network. While we concentrate on the second aspect in Section 3.2.5, the view described in Figure 4 is developed to improve the functionality of train schedule displays. Following the conservative approach towards improvements of safety-critical interfaces as described in Section 3.2.2, the general paradigm of the visual representation of the train schedule is retained, representing time on the horizontal axis and the sequence of train stations on the vertical axis. As improvements over the original view, we have added the following features:

Zooming & Panning Formerly being a fixed display with the current time locked in the middle of the view and a fixed amount of time to view into the past and future, we provide intuitive zooming and panning capabilities to the operator. By interactively pulling the schedule by click to the left or the right, the operator can extend his view arbitrarily into the past or future and is still able to quickly return to the current time. Also, train schedules for large train stations and track sections can get visually complex easily, being worse if large time intervals are display (meaning more trains in the display). As a solution for alleviating these issues, the operator can as well change the displayed time interval, thus zooming in on the temporal axis.

Interactive Highlighting In the former system, no direct interactions with the train representations was possible. We have implemented several visual cues which help an operator to maintain his focus even in complex schedules. To assist with the train tracking in complex displays, a train is highlighted when hovering over its representation. Also, an unobtrusive dashed vertical helper line in blue provides a reference to the exact time at the chosen position. Blue dots indicate intersections of all trains with the helper line, helping to identify the rough position of a train at any given point in time. As well, a hovering click on a train provides a link back to the physical track network, indicating which of the physical tracks the train is scheduled to occupy. If a train occupies a non-default track, red dots indicate this condition.

[4] `https://docs.docker.com/install/`, accessed 18. July 2019
[5] `https://nodejs.org/en/`, accessed 18. July 2019
[6] `https://expressjs.com/`, accessed 18. July 2019
[7] `https://angular.io/`, accessed 18. July 2019
[8] `https://d3js.org/`, accessed 18. July 2019
[9] Username: dreams ; Password: $dreams31$

Track Network Topology While the schedule displays are typically arranged as linear sequence of train stations, the physical track network can be more complex. For example, the could be shortcuts or alternative routes between two or more stations in the network which are not reflected in the linear layout. We add this additional information as connections to the left of the stations to enable the operator to easily identify alternative routes for scheduling purposes.

Station Detail View When rescheduling trains, for example due to delays or maintenance works, it is not only important to find a time when the track is free to be used, but also to identify, whether the scheduled stops feature enough tracks to actually accommodate all trains scheduled to be in a specific station at a given time. While this was not possible before, now, a click on a station expands the available physical tracks and their occupancy over time. For each train stopped in a station, a label indicates its assigned track. For quick rescheduling, trains can be quickly dragged from one track to another interactively if necessary.

Maintenance Works Planned and ongoing maintenance works appear as areas dashed in red lines. The direction of the dashes indicates which track direction is affected, and checkered dashes indicate a full closure of the respective track section. A click on the planned maintenance window opens the maintenance prediction screen, which assists operators in estimating the planned maintenance works depending on parametric based inputs such as location, conditions, time of day or weather. A decision tree visualizes the prediction model outcome and allows the operator to judge the quality of the results based on the decisions made by the model. A more detailed description of the prediction model and its parameters, for which we provide the visual interface here, can be taken from Section 3.2.1.

Conflict Resolution Modal Train conflicts are further represented by a circled cross and indicate points in space and time where trains would run into a conflict if the conflict is not resolved. To do so, an operator has to decide between two options, either, train A can be allowed to go first from its last station before the conflict, or train B. This decision has measurable impacts on different parameters. For example, one can measure, which train would need to wait longer in a station if the other was allowed to go first or how the train running time develops for each train (dwell time & running time). Other measures are the total train running time or the economical cost resulting from a decision. When trying to resolve a conflict, our demonstrator provides a hover-window first with the most relevant information about the conflict such as affected trains or specific track. Second, to resolve the issue, we provide a conflict resolution view opening on click: A prediction model predicts the consequences for each option (letting train A or B pass before the other). The results are expressed in a bar chart for each decision a highlighting of the “winner” for each predicted measure helps operators to understand the consequences of their decision without further need for investigation. As well, the impact on the schedule are automatically presented for each decision as a preview on how the schedule would look like, including a difference visualization for the shift in the schedule. A table with recorded decisions of operators and solutions suggested by the model in the past helps the operator to incorporate previous experiences. This view and the underlying prediction and its visualization are explained in more detail in Schlegel et al. [9].

3.2.4 Task 2: Alert Management and Prioritization System for AMS

While organizational overlaps exist, requirements for train scheduling and maintenance scheduling are different and thus need different visual solutions. In the latter scenario, an operators focus lies on monitoring the current state of the whole track system including thousands of different components which can break down or need maintenance. Given the large amount of components, it is obvious, that hundreds of faults occur every day which need to be attended according to severity of the fault. While current systems are able to provide an overview, they lack in providing an operator with a complete, prioritized situational awareness. Figure 5 shows a novel view where We present improvements to several aspects of the maintenance overview screen, enabling an operator to get a more detailed overview at a glance and to judge the severity of faults by hierarchical encoding.

Spatial overview In the updated view, organizational units can now be represented as unobtrusive colored boxes, enabling to encode information such as human/technical resource affiliation for each represented station. As well, while we incorporate the established color coding scheme of the current system, we reduce the amount of primary colors applied. Consequently, our color model is able to modulate the perceived importance of visual components by reducing or adding contrast to its representation. For example, important components or high-priority faults will be represented in strong primary colors again, while components with guidance functionality such as connecting tracks are featured in dark grey to provide context without catching attention. Consequently, components and tracks in regular working condition will be represented with less contrast than stations with maintenance alerts of high priority. Concerning the planning of maintenance works, the position of trains running in the system are represented by white arrowheads, allowing maintenance operators to judge, whether and how trains will be affected by recorded faults in the near future.

Component Hierarchy & Color Coding Current systems are able to internally encode a component hierarchy (e.g. Station->peripheral equipment->server->file system->hard disk), but fail to incorporate this hierarchy conceptually. For example, if a component far down in the hierarchy fails (in our example, the hard disk), not only one alert is generated, but one for each hierarchy level, confronting operators with far more alerts than actual faults exist in the system. Current systems display the severity of the highest alert for a station as color for the whole station. For example, in case a minor, non-vital component fails, the whole station will be indicated as red (= alarms present). This is not helpful when a maintenance operator wants to assess the system status quickly, since experience shows, that for any given station, usually non-critical faults are present. Our approach helps in two ways: First, our system divides each station representation in four quadrants, where each quadrant represents a main category of components. While freely choosable, based on expert feedback, we identify power systems, interlocking systems, train control equipment and station periphery as main categorization. An operator can now see at a glance, which kinds of systems are affected in a station and thus, can judge better the impact of faults on service quality. A user can interactively follow the fault hierarchy by clicking on a category, which opens new hierarchy levels.

Temporal Pattern View Equipment faults can have many different causes, and sometimes are caused by external influence. For example, if a sensor fails, its power supply can deliver wrong power levels without the power supply itself showing a fault. If the sensor is changed, it can break again quickly. To diagnose such complex faults, our system does not only provide a detailed view about the failure history in each identified hierarchy level, but also a visual temporal overview. Using this overview, a maintenance operator can identify long-term patterns within the alert history, helping to limit possible causes temporally and by their specific nature.

3.2.5 Task 3: Improving the TMS and Directing the Awareness of the Operator

As mentioned in Section 3.2.3, another important aspect of train management is management from the track perspective instead of the schedule perspective. While these views complement each other, they satisfy different requirements. Whereas scheduling is about **when trains go where**, the topological track network view allows to determine *which path through a network is available* for scheduled trains to go to their destination. The goal of our improvements is to improve the readability of these potentially extremely complex displays.

Improved Track Occupancy Representation Current displays are able to show train positions and track segments used or blocked by the same. For operators, it is also important to see, which track segments are blocked for the next train movements. In addition to the conventional red representation of passed track segments in red, we incorporate a visual highlighting of segments blocked next by applying a ribbon-animation, which indicates the direction as well.

Context-Dependent Highlighting A major issue with track displays is their complexity which grows with amounts of stations and especially the amount of tracks and track branches in the network. Currently, existing systems provide overviews of parts or the whole network on large screen walls. As well, no visual distinction is made for parts of the network which are important versus parts with less importance. Given the complexity of the networks, switching contexts is a cognitively costly action for an operator. The context-dependent highlighting we introduce helps to alleviate the cognitive load of operators by visually making them aware of areas in the network which are more important than others. Specifically, an operator can choose from three different highlighting options: First, a circular highlight can be chosen to be shown around running trains, while the rest of the view will be greyed out, allowing operators to easily spot and concentrate on the trains themselves. Second, operators can choose to concentrate on stations instead of trains. The same circular highlight metaphor is applied here as well, with the stations in the center of the highlight instead of the trains. The third highlighting mode allows the operator to focus on peripheral equipment which can be numerous and hard to spot in the complex views otherwise.

Attention Steering Current displays provide an overview over the network status, but do not support the operator in determining important events (for example, when a train does not follow a scheduled track). Given the large physical size of these displays and the complexity of the track networks, even at full attention it is not possible for operators to perceive every development. We aid operators by providing a homing animation which is realized as a green circle closing in over the screen towards an area of interest. Given defined conditions, this method can be used for attention steering, making an operator aware of special events and triggering an appropriate reaction.

4 Demo

In this section it is presented a demo path of the integrated demonstrator, based on a sequence of the most relevant screenshots. This shots have been selected from the whole demo path recorded during testing. To reach the aim of making the testing of the demonstration simple, in particular the demo path might be illustrated with the following steps:

1. Creation of the restoration time predictive models, in order to start the process (Figure 7);
2. Opening of the new maintenance work to allow the operators to start the work and all the related procedures (Figures 8, 9, and 10);
3. Triggers of the data analytic module to make a prediction of the restoration to be displayed to the operators (Figure 11, 12, and 13);
4. The maintenance job steps performed in sequence and the maintenance job is closed (Figure 14 and 15);

A video-recording with explanation of the entire demo is uploaded together with this deliverable and the software on the cooperation tool.

5 Conclusions

This deliverable describes the demonstrators that have been developed to show the potentialities of the results of Tasks 5.2, 5.3, and 5.4. In particular, we described the architecture of the final WS2 Proof Of Concept which leverages on both Data and Visual Analytics Technologies and Distributed Ledger Technologies to better manage maintenance actions in the railways context. The proposed Proof Of Concept counts of three main modules: (i) the proof-of-concept developed on WP4 for handling the maintenance process as reliable data source thanks to the Distributed Ledger Technologies, (ii) the data analytic models developed in WP5 for Specific-Scenario 3 for predicting the restoration time from each maintenance as important information to provide to the train operators and the infrastructure managers, and (iii) the visual analytic models developed in WP5 for Cross-Scenario 1 for displaying the information to the operators coming both from the Distributed Ledger Technologies and the Data Analytic models. On one side we employed a permissioned Distributed Ledger Technology to ensure the complete transparency and auditability of the process, the integrity and availability of the inserted data and, most of all, the non-repudiation of the actions performed by each participant in the maintenance management process (see the deliverable of WP4). On the other side, exploiting the availability of the data in a single repository (the ledger) and with a standardised format, thanks to the utilisation of a Distributed Ledger Technologies, we adopt Data and Visual Analytic to leverage on the features of each maintenance job, together with external factors, to estimate the maintenance restoration time (see the deliverable of WP5, with particular reference to Specific-Scenario 3: Restoration Time). All these crucial information have been displayed to the different operators, exploiting the tools developed in the Cross-Scenario 1: Visualizations in the Control Center.

References

[1] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, and Others. Hyperledger fabric: A distributed operating system for permissioned blockchains. In *EuroSys*, 2018.

[2] R. Benbunan-Fich and A. Castellanos. Digitalization of land records: from paper to blockchain. In *International Conference on Information Systems*, 2018.

[3] G. Budai, D. Huisman, and R. Dekker. Scheduling preventive railway maintenance activities. *Journal of the Operational Research Society*, 57(9):1035–1044, 2006.

[4] Renzo Canepa, Simone Petralli, Luca Oneto, Wolfgang Jentner, Juri Buchmüller, Charlotte Ducuing, Ivo Emanuilov, Marie Swiatek, and Davide Anguita. In2dreams d5.1: Data analytics scenarios. Technical report, 2018.

[5] J. De Kruijff and H. Weigand. Understanding the blokchain using enterprise ontology. In *International Conference on Advanced Information Systems Engineering*, 2017.

[6] T. Farrington-Darby, L. Pickup, and J. R. Wilson. Safety culture in railway maintenance. *Safety Science*, 43(1):39–60, 2005.

[7] E. Fumeo, L. Oneto, and D. Anguita. Condition based maintenance in railway transportation systems based on big data streaming analysis. In *INNS International Conference on Big Data*, 2015.

[8] Daniel Keim. Mastering the information age: solving problems with visual analytics. 2010.

[9] Udo Schlegel, Wolfgang Jentner, Juri Buchmüller, Eren Cakmak, Giuliano Castiglia, Renzo Canepa, Simone Petralli, Luca Oneto, Daniel A. Keim, and Davide Anguita. Visual analytics for supporting conflict resolution in large railway networks. In *2019 INNS Big Data and Deep Learning (INNSBDDL 2019)*, Sestri Levante, Genoa, Italy, 2019.

[10] A. Thaduri, D. Galar, and U. Kumar. Railway assets: A potential domain for big data analytics. In *INNS International Conference on Big Data*, 2015.

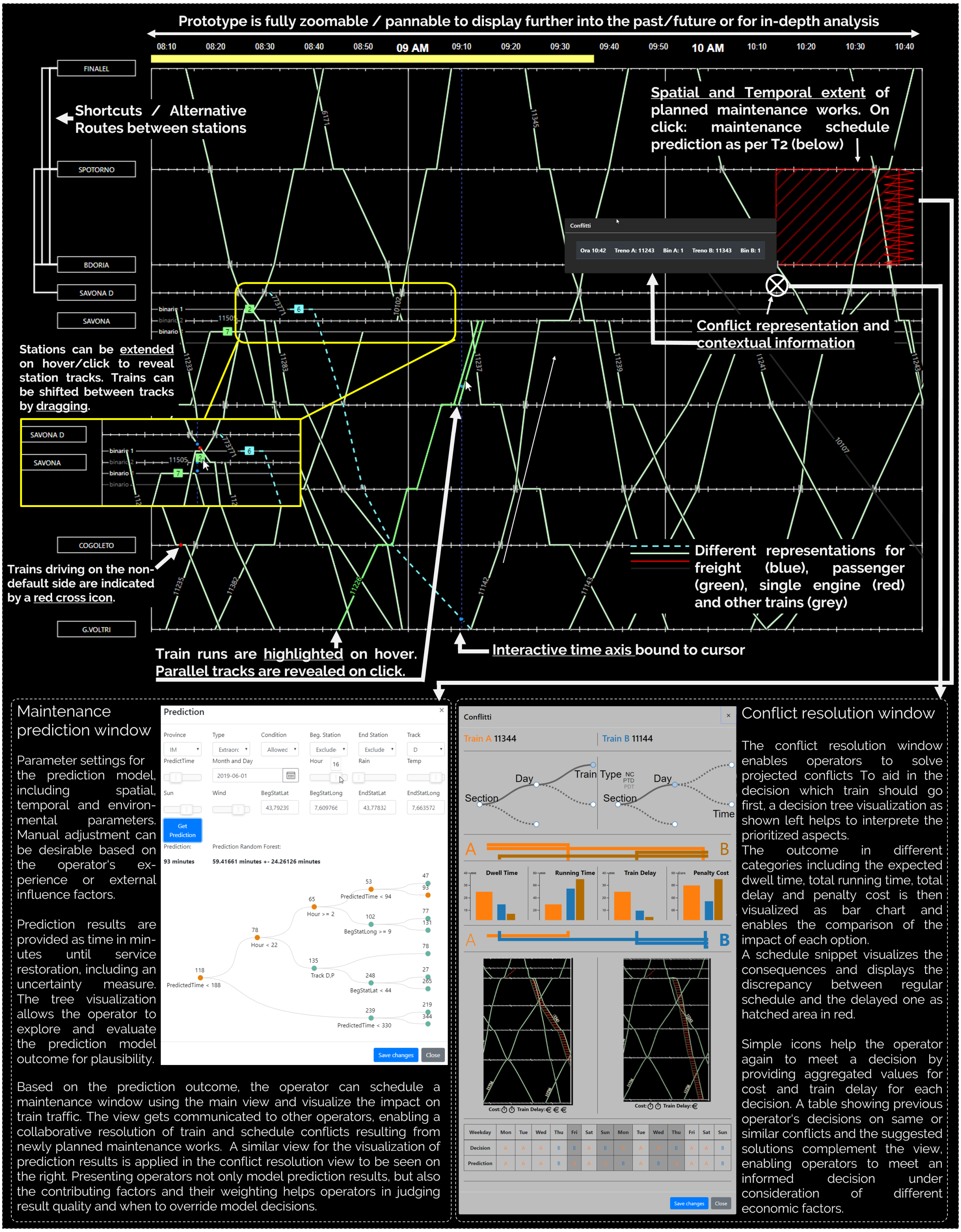

Figure 4: Description of T1 demonstrator interface

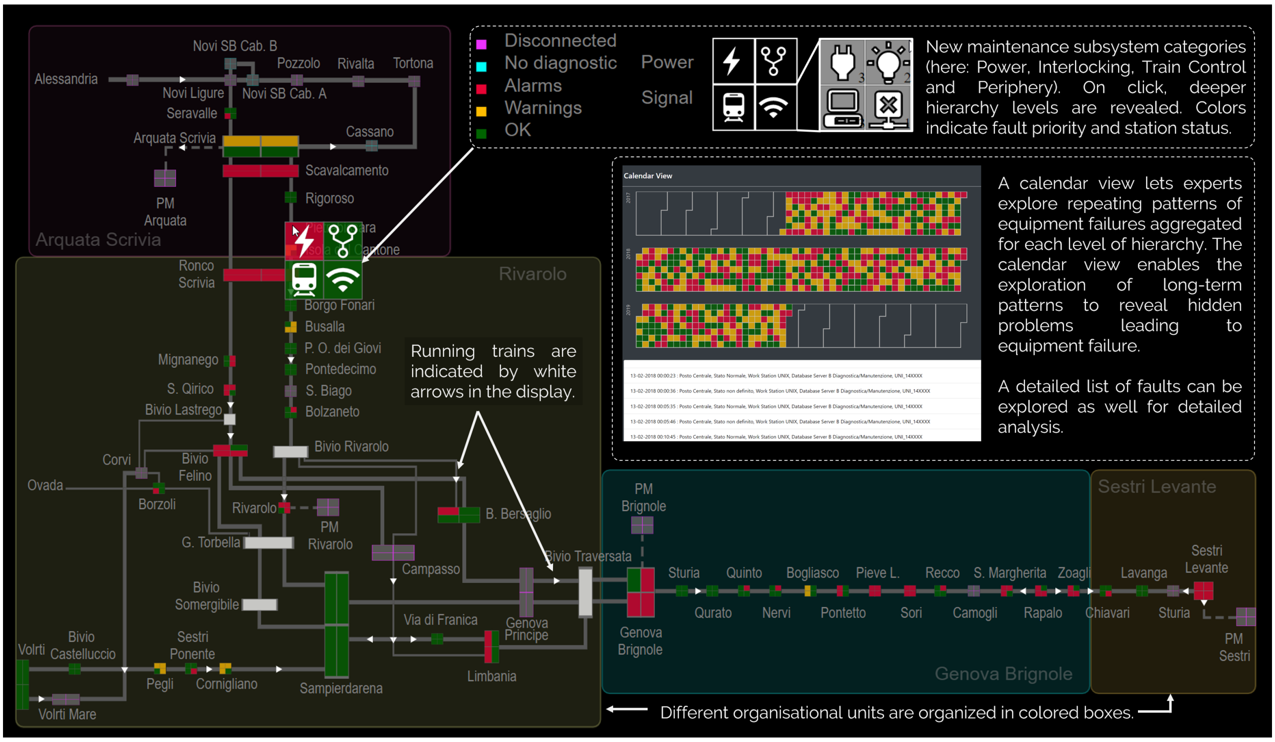

Figure 5: Description of T2 demonstrator interface

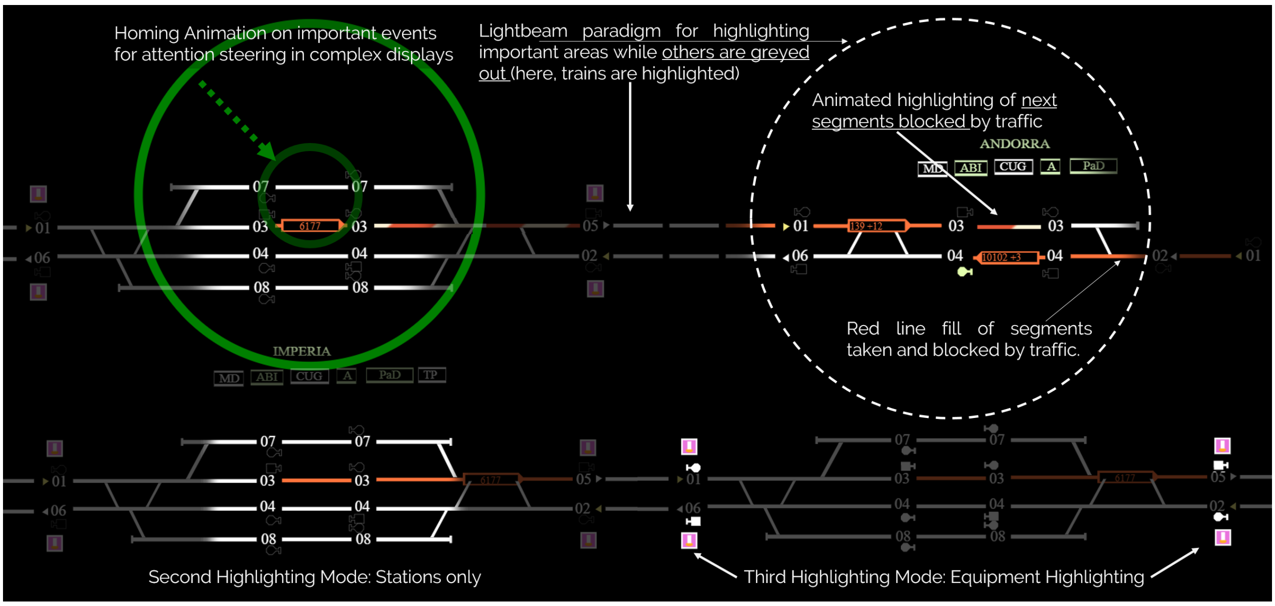

Figure 6: Description of T3 demonstrator interface

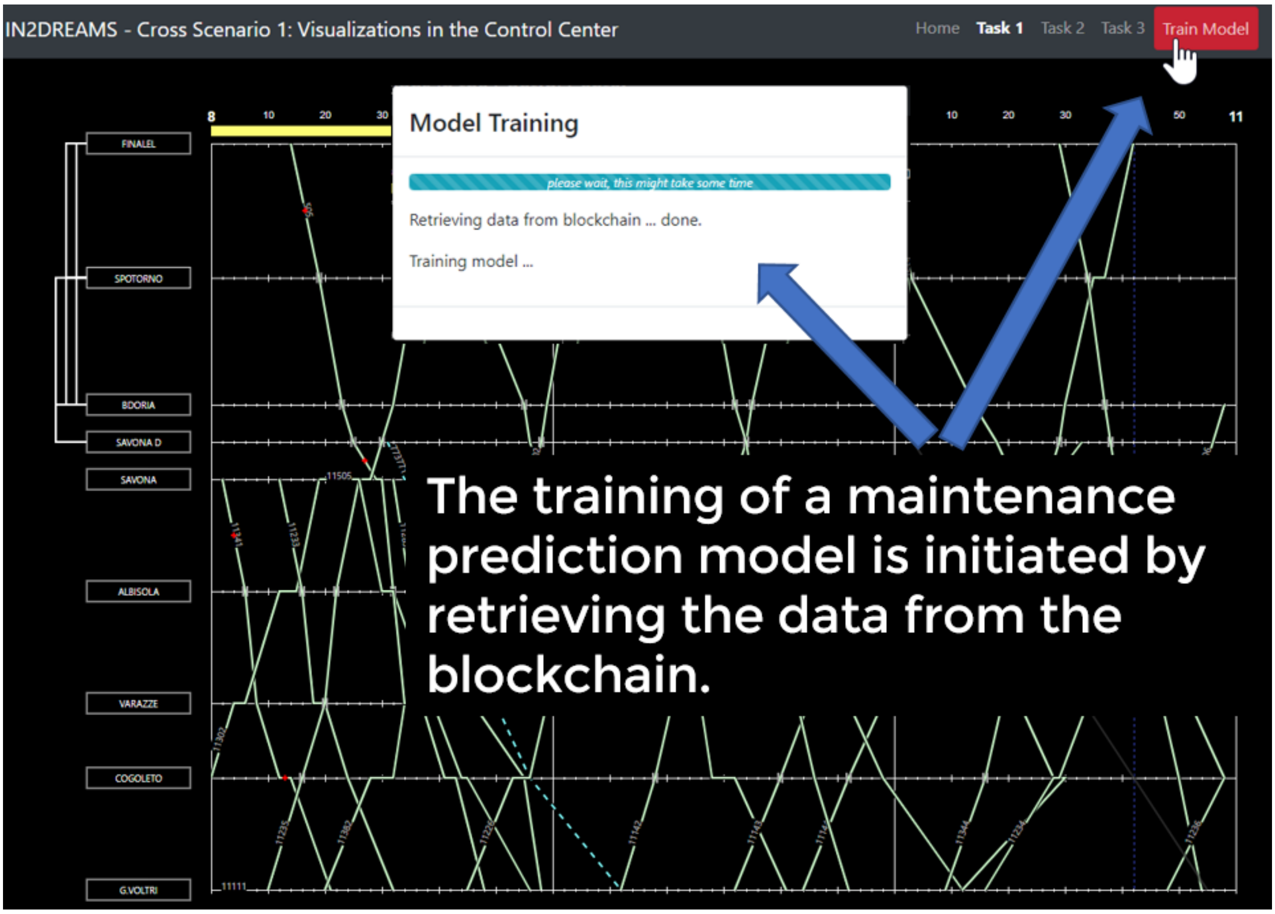

Figure 7: Step 1 of the demo described in Section 4: Training of a maintenance prediction model.

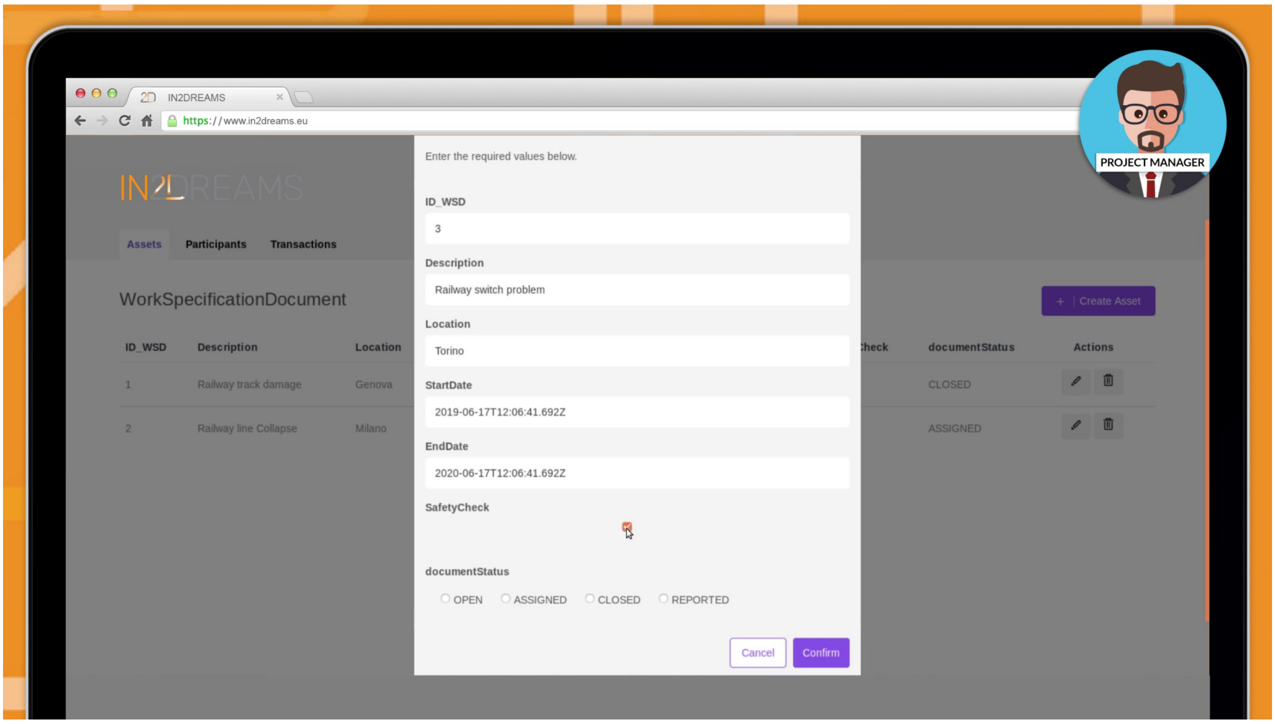

Figure 8: Step 2 of the demo described in Section 4.

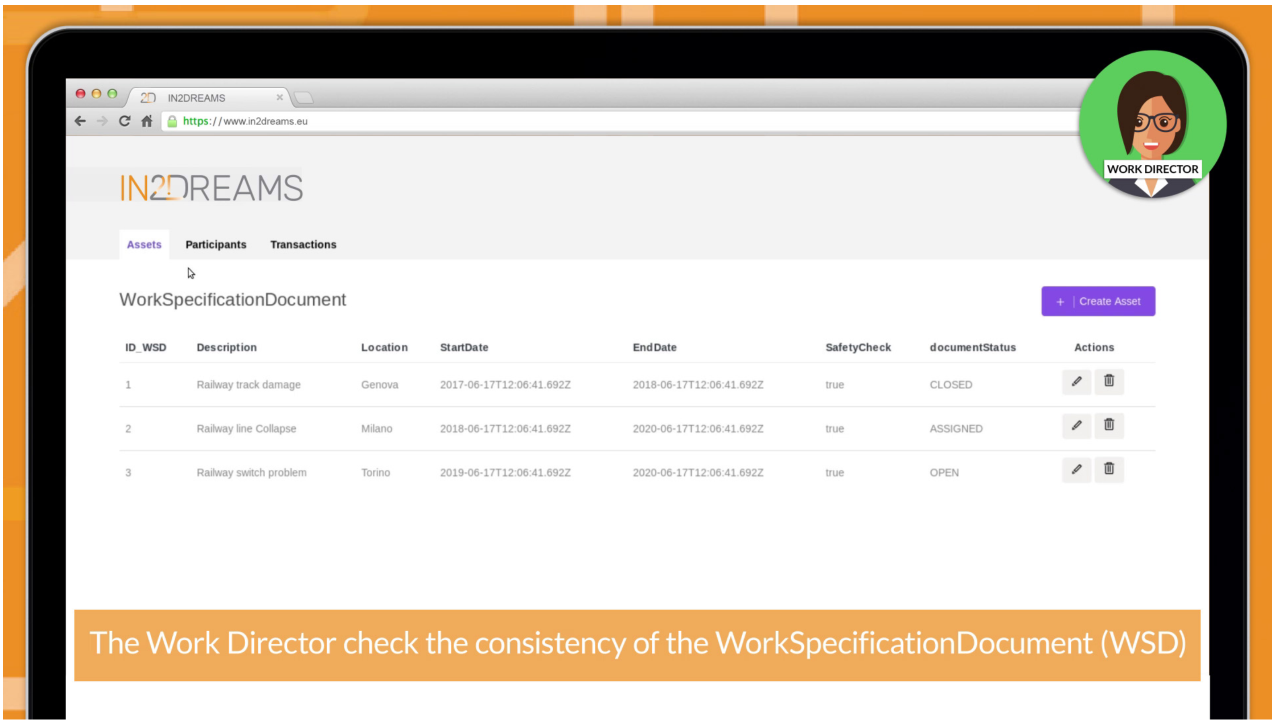

Figure 9: Step 2 of the demo in Section 4. The process includes checks by Work and Technical

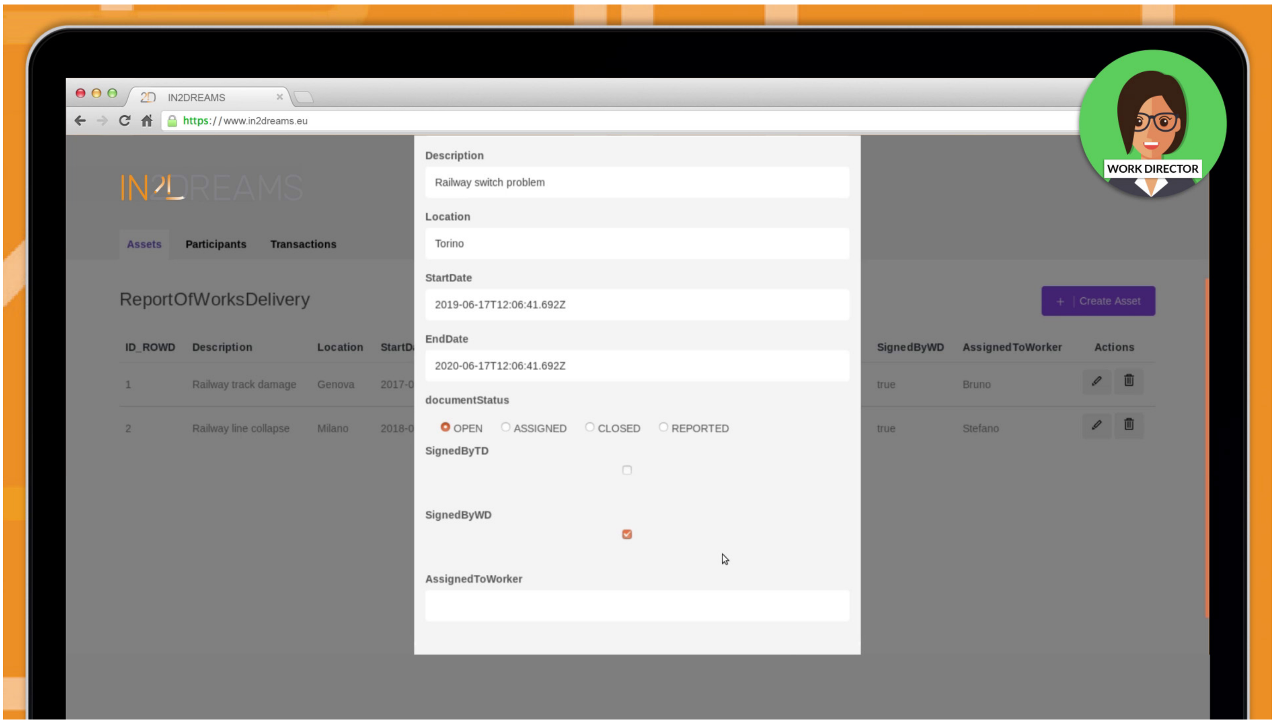

Figure 10: Step 2 Progression of the workflow managed in the system

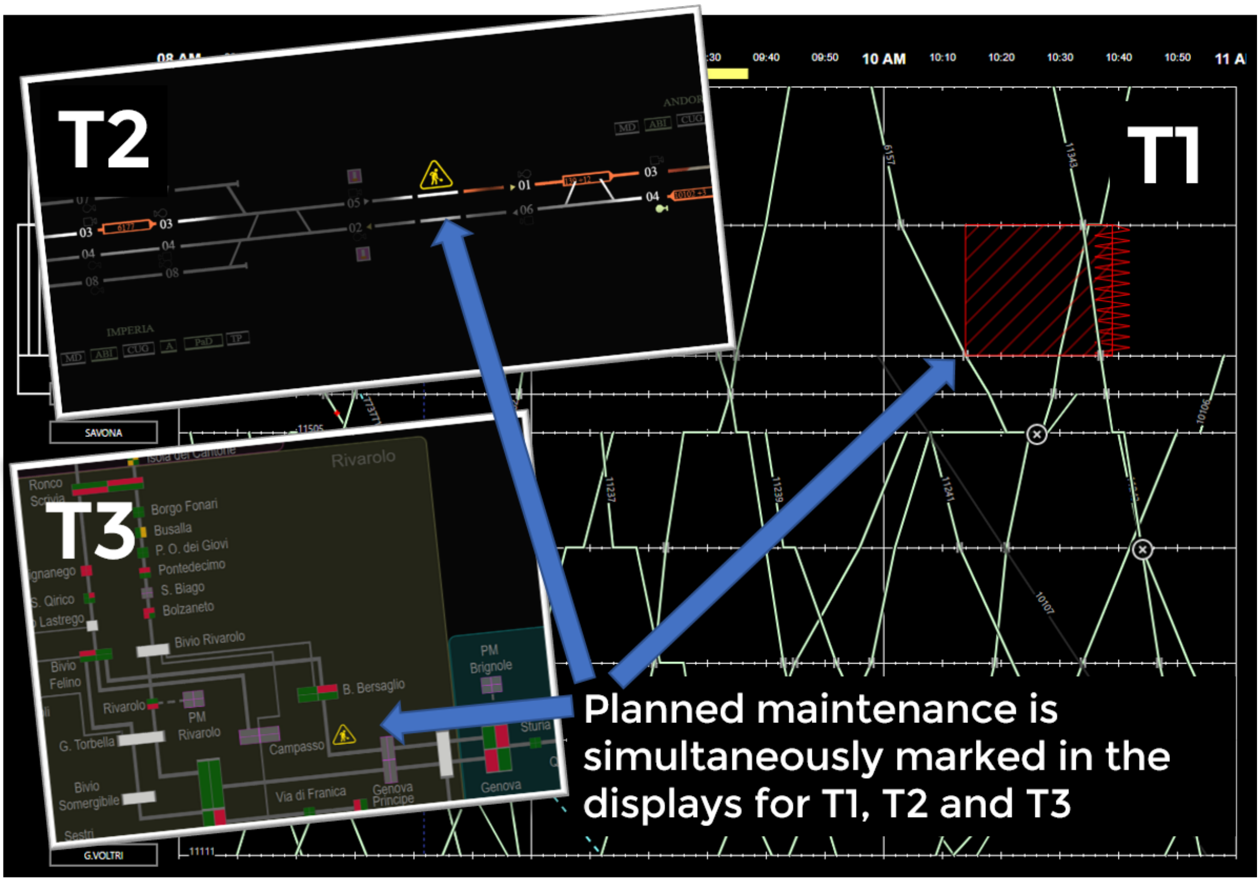

Figure 11: Step 3 of the demo described in Section 4: Train management operators and maintenance managers receive the information about upcoming maintenance and the location on their main views. In T1, the estimated time is displayed additionally.

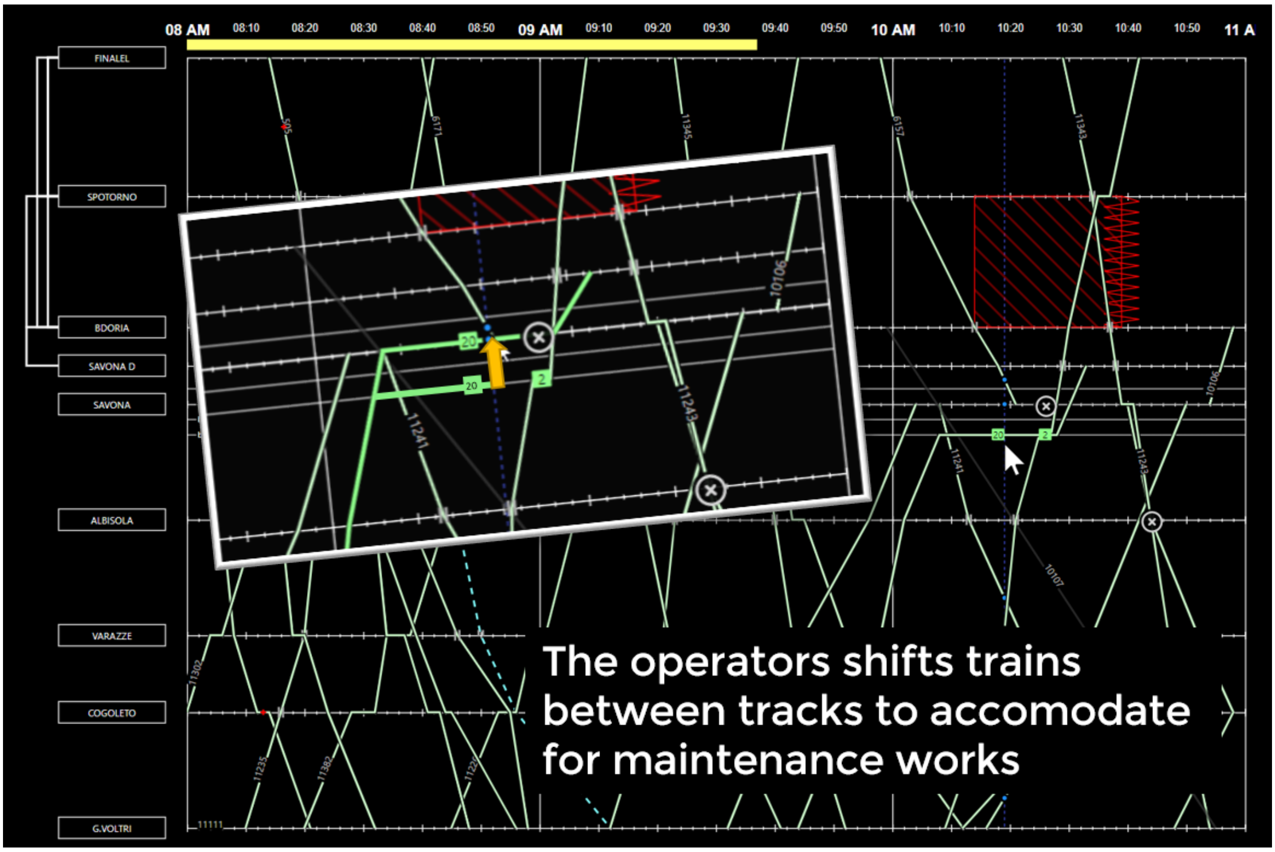

Figure 12: Step 3 of the demo described in Section 4: In reaction to the new planned maintenance works, operators can reschedule trains and change tracks of trains affected by the maintenance works.

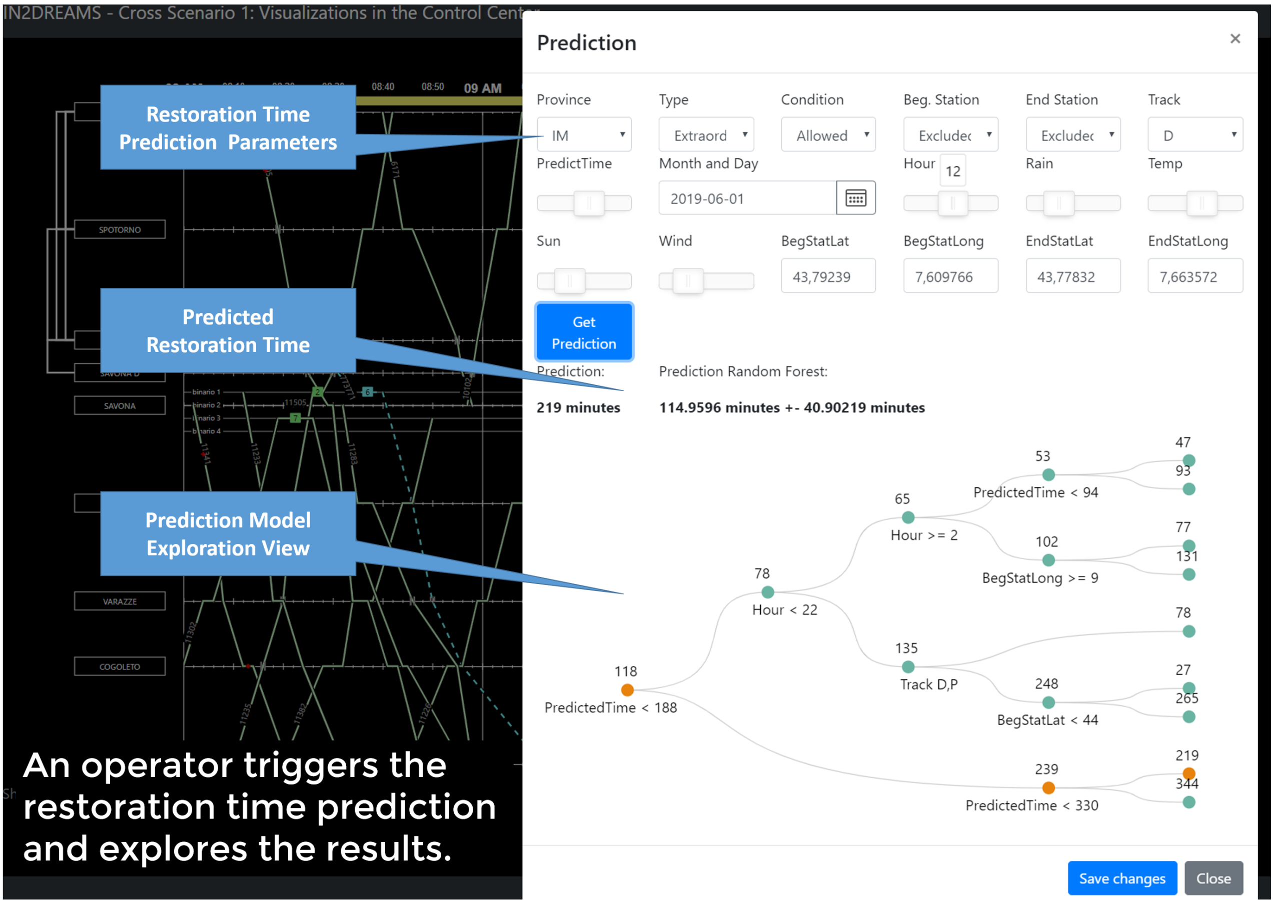

Figure 13: Step 3 of the demo described in Section 4: Environmental factors can influence the duration of the maintenance works. An operator can check and modify the parameters influencing restoration time. Updated times will be reflected in the schedule view immediately.

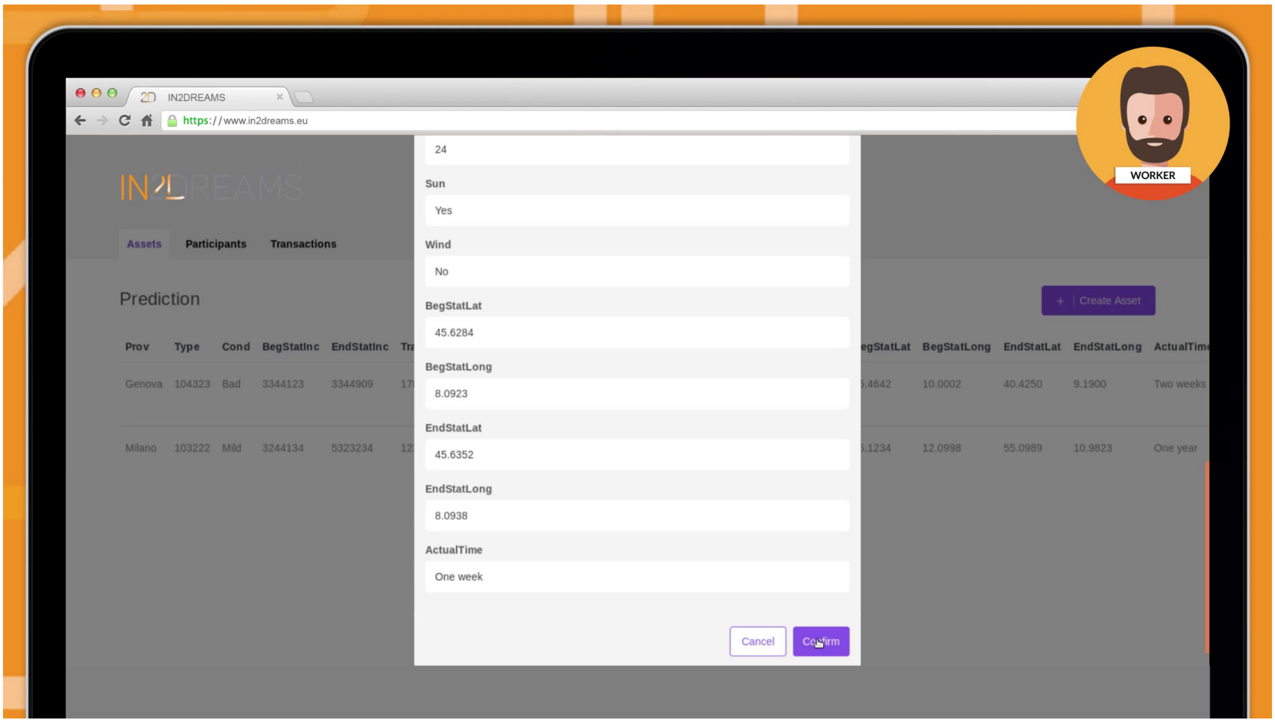

Figure 14: Step 4 of the demo in Section 4. Worker compiles the prediction document and proceed with the maintenance job.

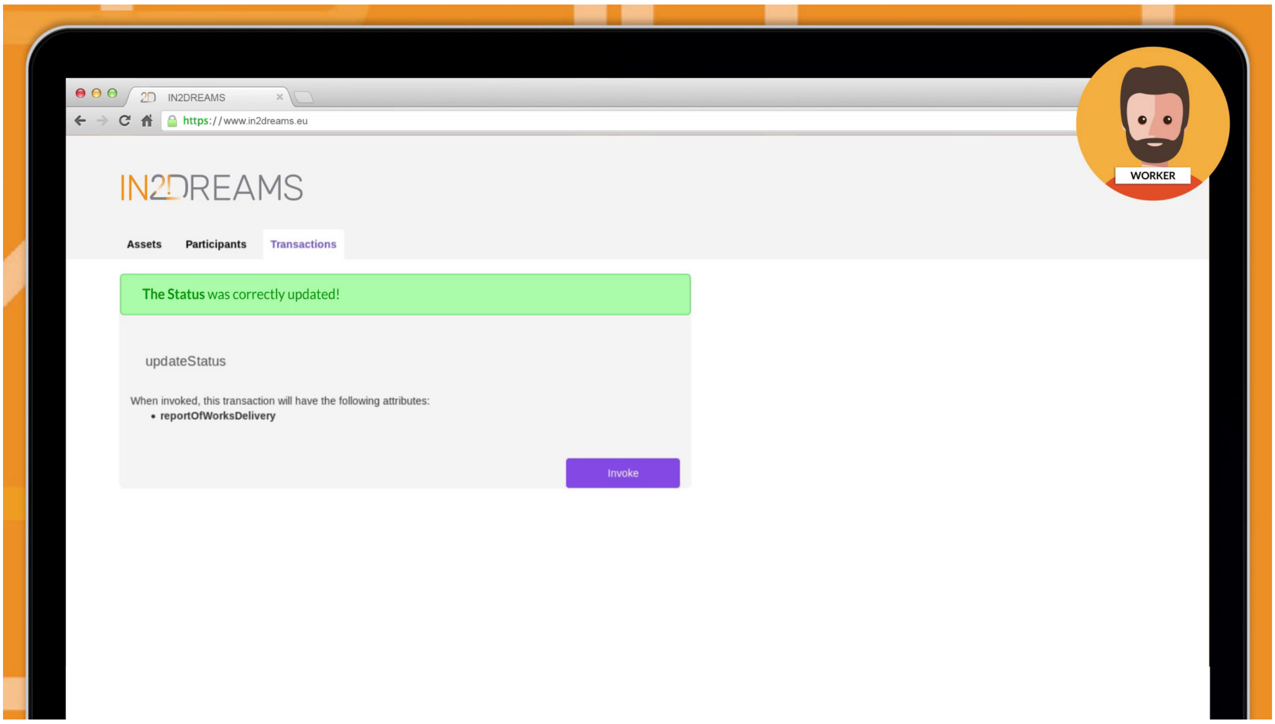

Figure 15: Step 4 of the demo in Section 4. When the workers finish with the job they need to report the status