

Information and Communication Technologies (ICT)

Programme

Project N°: H2020-ICT-2016-1-732105



D6.10: Smart Travelling Demonstrator (Ver. I)

Lead Beneficiary: 10 – TNO

Work Package: 6

Date: 26-June-18

Distribution - Confidentiality: [Public]

Abstract:

This document contains the description of the M18 Smart Travelling demonstrator. The description is based on the skeleton defined in D6.7 and includes the scope and purpose of the demonstrator, the description of the demonstrator itself and the accomplished results till now.

Disclaimer

This document may contain material that is copyright of certain CERBERO beneficiaries, and may not be reproduced or copied without permission. All CERBERO consortium partners have agreed to the full publication of this document. The commercial use of any information contained in this document may require a license from the proprietor of that information.

The CERBERO Consortium is the following:

Num.	Beneficiary name	Acronym	Country
1 (Coord.)	IBM Israel – Science and Technology LTD	IBM	IL
2	Università degli Studi di Sassari	UniSS	IT
3	Thales Alenia Space Espana, SA	TASE	ES
4	Università degli Studi di Cagliari	UniCA	IT
5	Institut National des Sciences Appliquées de Rennes	INSA	FR
6	Universidad Politecnica de Madrid	UPM	ES
7	Università della Svizzera italiana	USI	CH
8	Abinsula SRL	AI	IT
9	Ambiesense LTD	AS	UK
10	Nederlandse Organisatie Voor Toegepast Natuurwetenschappelijk Onderzoek TNO	TNO	NL
11	Science and Technology	S&T	NL
12	Centro Ricerche FIAT	CRF	IT

For the CERBERO Consortium, please see the <http://cerbero-h2020.eu> web-site.

Except as otherwise expressly provided, the information in this document is provided by CERBERO to members "as is" without warranty of any kind, expressed, implied or

WP6 – D6.10: Smart Travelling Demonstrator

statutory, including but not limited to any implied warranties of merchantability, fitness for a particular purpose and non-infringement of third party's rights.

CERBERO shall not be liable for any direct, indirect, incidental, special or consequential damages of any kind or nature whatsoever (including, without limitation, any damages arising from loss of use or lost business, revenue, profits, data or goodwill) arising in connection with any infringement claims by third parties or the specification, whether in an action in contract, tort, strict liability, negligence, or any other theory, even if advised of the possibility of such damages.

The technology disclosed herein may be protected by one or more patents, copyrights, trademarks and/or trade secrets owned by or licensed to CERBERO Partners. The partners reserve all rights with respect to such technology and related materials. Any use of the protected technology and related material beyond the terms of the License without the prior written consent of CERBERO is prohibited.

Document Authors

The following list of authors reflects the major contribution to the writing of the document.

Name(s)	Organization Acronym
Joost Adriaanse	TNO
Pablo Muñoz	S&T
Giovanni Turi	CRF

The list of authors does not imply any claim of ownership on the Intellectual Properties described in this document. The authors and the publishers make no expressed or implied warranty of any kind and assume no responsibilities for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information contained in this document.

Document Revision History

WP6 – D6.10: Smart Travelling Demonstrator

Date	Ver.	Contributor (Beneficiary)	Summary of main changes
4-June-18	0.1	TNO	1 st Draft
5-June-18	0.2	TNO, UNISS	Second draft after review UNISS
12-June-18	0.3	TNO	Third draft after internal review
15-June-18	1.0	TNO, S&T, CRF	First version for final review
15-June-18	1.1	TNO, CRF	Incorporated review comments CRF
26-June-18	1.2	TNO, UNISS, INSA, IBM	Incorporated review comments UNISS, INSA and IBM

Table of contents

1. Executive Summary.....	6
1.1. Structure of Document.....	6
1.2. Related Documents.....	6
2. Scope and purpose	8
3. Description of the Smart Travelling demonstrator.....	10
3.1. Functionalities.....	12
3.1.1. Real time system-in-the-loop simulation.....	13
3.1.2. Self-Adaptation.....	14
3.1.3. Development of the data fusion application.....	24
3.1.4. Prepare CERBERO intermediate format interfaces	24
3.2. Integrated and Evolved Tools	25
3.3. Development and deployment environment	25
4. Tests, Results and Feedback.....	27
4.1. Tests	27
4.2. Test Results	28
4.3. Feedback.....	29
5. Conclusion	30
6. References	31

1. Executive Summary

This document describes the M18 Smart Travelling demonstrator. The document is a preliminary version of D6.4, due by M36, which will describe the final Smart Travelling demonstrator as one of the CERBERO use-case.

1.1. Structure of Document

In Section 2 the scope and purpose of the demonstrator is described. Section 3 includes the description of the developed demonstrator. In Section 4 the tests, results and feedback are outlined. Section 5 summarizes our conclusion so far and Section 6 provides the references.

1.2. Related Documents

The CERBERO deliverables related to this document are:

- D2.7 – Technical Requirements (Ver. II)
 - The development of the demonstrator will contribute to satisfy and validate the requirements listed in D2.7
- D2.4 – Description of Scenarios (Ver. II)
 - The smart travelling demonstrator will be based on the use case scenario as defined in D2.4
- D3.4 – Modeling of KPI (Ver. I)
 - The addressed KPIs are based on the generic list of KPIs as defined in D3.4
- D3.6 - Cross Modelling Methodology for CPS (Ver. I) –
 - The methodology applied for cross layer modelling of the CPS of the smart travelling demonstrators is described in D3.6
- D5.6 – Framework Components (Ver. I)
 - The intermediate format for interfacing CERBERO tools in the (M36) demonstrator is described in D5.6
- D6.7 – Demonstration Skeleton (Ver 1)

WP6 – D6.10: Smart Travelling Demonstrator

- The generic skeleton used to build the smart travelling demonstrators is described in D6.7

Another related document is the internal test scenario definition document (see [ST-TEST-SCENARIO]), in which a sample test scenario for smart traveling use case is defined, based on the scenarios defined in D2.4. This document is used for the initial development of the (M18) demonstrator.

2. Scope and purpose

In this document the M18 demonstrator of the Smart Traveling for Electric Vehicles use case is described in order to demonstrate and validate the CERBERO tool-chain for deployment in a cross-layered and adaptive CPS. The CRF driving simulator can be seen as a real time simulator of an electric vehicle, which requires adaptation and multi KPI optimization via added functionality for driver support.

CRF would like to extend its current driving simulator functionality in a flexible way, where CRF is not completely dependent on Oktal (the vendor of the used SCANer software) but can quickly include completely new functionality Oktal cannot or does not yet provide. This will be especially valuable for future electric and autonomous cars.

The M18 demonstrator provides the first iteration of the planned demonstrator, where the development was focused on connecting the CERBERO tools to the CRF simulator (based on the SCANer tool, see [SCANer]) and the development of the required interfaces between all tools.

In the next (M36) iteration the functionality will be further extended using additional tools from the CERBERO toolchain and interfaces between the CERBERO tools will be transferred to interfaces based on the CERBERO intermediate format (see D3.6).

The drivers of demonstration activities have been defined in D2.7. In the following table an excerpt of Table 4 of D2.7 is provided.

Table 2-1 –Requirement validation table

User Requirements	Technical Requirement(s)	Validation within the ST demonstrator	Planned Month
ST1. Develop reconfigurable extendable modular simulation environment for smart travelling driver interfaces.	5, 7 and 8 (see section 4.5 and table 4 in 4.6 in D2.7)	Integrate CERBERO tools (e.g. DynAA and MECA) in the car simulator and show environment can be configured to support	M18 integration of tools, M36 execution of use case scenarios.

WP6 – D6.10: Smart Travelling Demonstrator

		defined use case scenarios.	
ST2. Develop integrated “open” toolchain environment for development of simulation modules and their integration with focus on modular integration with existing virtual environment.	2 and 4 (see section 4.5 and table 4 in 4.6 in D2.7)	<p>Integrate DynAA and a number of simulation modules into the car simulator in such a way that simulation modules can easily be added.</p> <p>In case of smart travelling demonstrator, the DynAA, MECA and SCANeR tools are not open source tools. The partners are however allowed to use the tools within the project.</p>	In M18 the battery and motor models are integrated using DynAA.
ST3. Development of a self-adaptation methodology with supporting tools.	20 (CERBERO-0020: “CERBERO framework SHALL provide methodology and tools for development of adaptive applications”, see section 4.5 in D2.7)	Using the DynAA and MECA tools efficient support of functional adaptation is shown, according to system (car), human (driver) and environmental triggers.	<p>In M18 the adaptation mechanism is developed.</p> <p>In M36 usage of different adaptation scenarios will be added and validated.</p>

In Table 4 of D2.7 we mapped user requirements with technical requirements, while we are discussing here how (Validation within the ST demonstrator) and when (Planned Month) the validation is going to take place. As you can see, the coverage of the various needs is distributed between M18 and M36.

3. Description of the Smart Travelling demonstrator

For the use case Smart Travelling the goal is to enrich the CRF driving simulator with functionalities required by the electric vehicle use cases (see D2.4). For this a number of CERBERO tools will be used. To integrate these tools with the CRF driving simulator (based on the SCANeR software tool), a number of interfaces need to be developed.

As depicted in the following picture, the interfaces developed for the M18 demonstrator involve both interfaces between CERBERO tools (DynAA and MECA) and interfaces with the use case specific CRF tool (SCANeR).

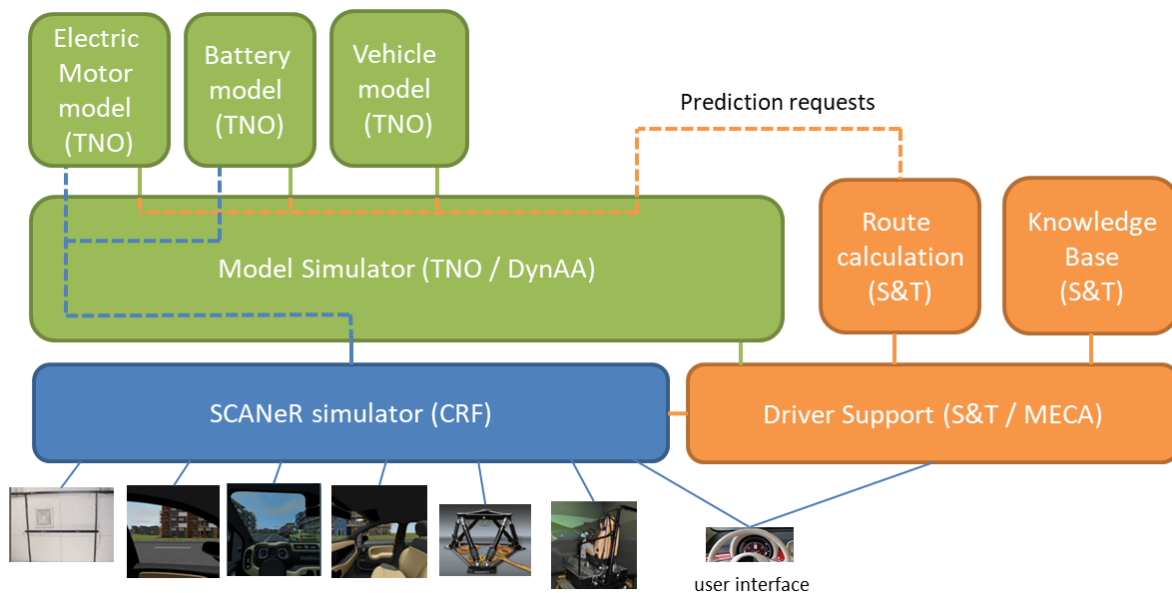


Figure 3-1: Schematic overview of the M18 ST demonstrator components and their interfaces.

The AOW optimization tool of IBM will not yet be integrated in the M18 demonstrator as first the basic driver support mechanism and integration of DynAA and MECA with SCANeR will need to be completed.

The high level CERBERO requirements related to the Smart Travelling use case are (see D2.7 section 4.6):

1. (ST1) Development of **parametric, modular and extendable** cyber-physical co-simulation environment;

WP6 – D6.10: Smart Travelling Demonstrator

2. (ST2) Development of an integrated open-source or commercially available toolchain for design space exploration and co-simulation, with **system-in-the-loop capabilities**;
3. (ST3) Development of **a self-adaptation methodology** with supporting tools.

Derived from these requirements, two main goals define what we want to accomplish during the development of the M18 demonstrator (see D2.4 section 2.2):

1. Verify the condition awareness capabilities of the CERBERO framework for **real-time system-in-the-loop simulation**.
2. Continuous system monitoring for **self-adaptation**.

These goals mainly cover the ST2 and ST3 requirements. To comply with the ST1 requirement, the development of the demonstrator has to be done in such a way that solution can be parameterised as much as possible (avoid hard coded algorithms), be modular (so split up in logical modules) and extendable (thus can easily be extended with new functionalities).

As the CERBERO use cases should be developed according to the defined CERBERO demonstration skeleton in D6.7, it is important to be able to identify skeleton parts in the use case, which could help in future reuse of the developed CERBERO tools. In the figure below a graphical mapping is done between the use case architecture and the CERBERO skeleton (see also Section 4.1 of D6.7).

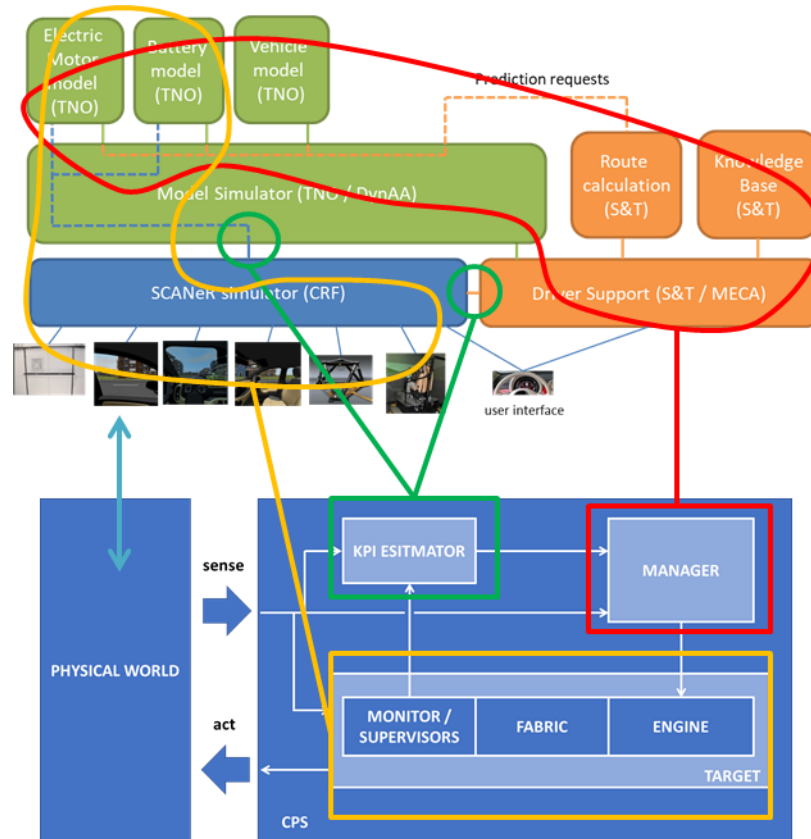


Figure 3-2: Smart Travelling use case - CERBERO skeleton mapping

Here we can see that the SCANer car simulator is their main target for adaptation, while the relevant KPI (sensor) data is gathered from the car, the driver and the external environment by the manager implemented using MECA and DynAA. MECA is responsible for giving advice and triggering adaptation based on internal or external signals. In the model DynAA performs two roles, where one is the simulation of parts of the car (battery and motor), functioning as part of the target and one as support to MECA in calculating and simulating possible routes to take (see D6.7 for a more elaborate explanation of the skeleton and the mapping).

3.1. Functionalities

The implementation of the M18 Smart Travelling demonstrator splits functionalities (according to set goals and requirements in D2.7) into the following four parts:

1. Real time system-in-the-loop simulation (see 3.1.1);

WP6 – D6.10: Smart Travelling Demonstrator

2. Self-adaptation (see 3.1.2);
3. Development of the data fusion application for CRF logfiles (see 3.1.3);
4. Preparation of CERBERO intermediate format interfaces (see 3.1.4).

In the following paragraphs the developed functionalities of different parts of the demonstrator are described in more details.

3.1.1. Real time system-in-the-loop simulation

In the real time system-in-the-loop simulation DynAA and the TNO battery and electric motor models are used to adapt the existing (combustion car) simulator into an electric vehicle simulator, including an electric motor and a battery (see also vehicle simulation mode in Section 2.3 of D2.4).

In order to facilitate a cyber-physical co-simulation environment for system-in-the-loop simulations, the existing DynAA simulation framework was extended with a real-time co-simulation module. This module consists of a set of interfaces that describe how real-time sensors or actuators can hook into the DynAA simulation environment. Without making any decisions on how to implement such sensors/actuators, the interfaces ensure that future physical environment or co-simulated systems can be added in a modular way to the framework.

In the M18 demonstrator the implementations of these interfaces for the SCANeR simulator are shown, by running a time-synchronized co-simulation of SCANeR and DynAA. The information flow is limited to a few parameters, the position of the vehicle and the battery state-of-charge. However, in the M36 we target a more tightly coupled experiment, in which both simulations influence one another.

The implementation is performed in the following steps:

- Extend DynAA with real time co-simulation module (see explanation above).
- Update TNO battery and motor models to support real time simulation and integration in DynAA. As DynAA is Java based and the original models were coded using C, some adaptation was needed to execute them within the DynAA environment.

WP6 – D6.10: Smart Travelling Demonstrator

- Integration test (SCANeR and DynAA) in TNO test platform.
- Execute real time simulation test scenario in demonstrator (and possibly record execution in video for later presentation).

3.1.2. Self-Adaptation

An important part of the self-adaptation functionality is the ability to calculate possible routes to the destination and estimate feasibility, costs, travel time and expected battery load for the given routes. Based on the calculation the user will be provided with a selected subset of alternatives. In the scenario definition document (see [ST-TEST-SCENARIO]) a sample adaptation scenarios is described which contains all relevant steps of the Smart Travelling scenarios described in D2.4.

At the highest level two phases were identified:

- I. **Planning phase** (where driver provides MECA with starting point and destination as well as desired departure and arrival times and the system returns a limited set of calculated itineraries from which user can select the preferred one);
- II. **Adaptation to off-nominal event** (where system will respond to an internal or external trigger and initiate some form of adaptation).

One of the most important activities in the development of self-adaptation was the definition of a format for routes and the design of the process to calculate routes, including the route characteristics and the charging capabilities.

When designing a mechanism and interfaces, special attention was paid to the KPIs (defined in D3.4) as well as existing open standards, specifically the Open Source Routing Machine (OSRM) standard, in order to maximize user acceptance, portability and reusability of our tools.

For the Smart Travelling use case initially the following KPIs were listed:

- *Response time to triggers* (especially important for the real time simulator mode where the TNO battery and motor models are executed and need to provide input at least 100 times a second);
- *Energy* (needed to perform the trip from origin towards destination);

WP6 – D6.10: Smart Travelling Demonstrator

- *Cost* (mainly the cost of energy obtained from the charging points);
- *Travel time* (of the route(s) towards the destination).

When a driver requests the best route to destination, he/she will require the system to reason about almost all of the KPIs listed above. An optimal solution has to be provided, which will often be a compromise between travel time, costs and energy. While calculating the best routes, the system will have to ensure that sufficient energy is available during the drive and that the answer is given within a reasonable timeframe (which will result in requirement for limiting the search space as much as possible).

The generic process of retrieving input from the user, generating and calculating the itineraries and providing advice to the driver is depicted in Figure 3-3..

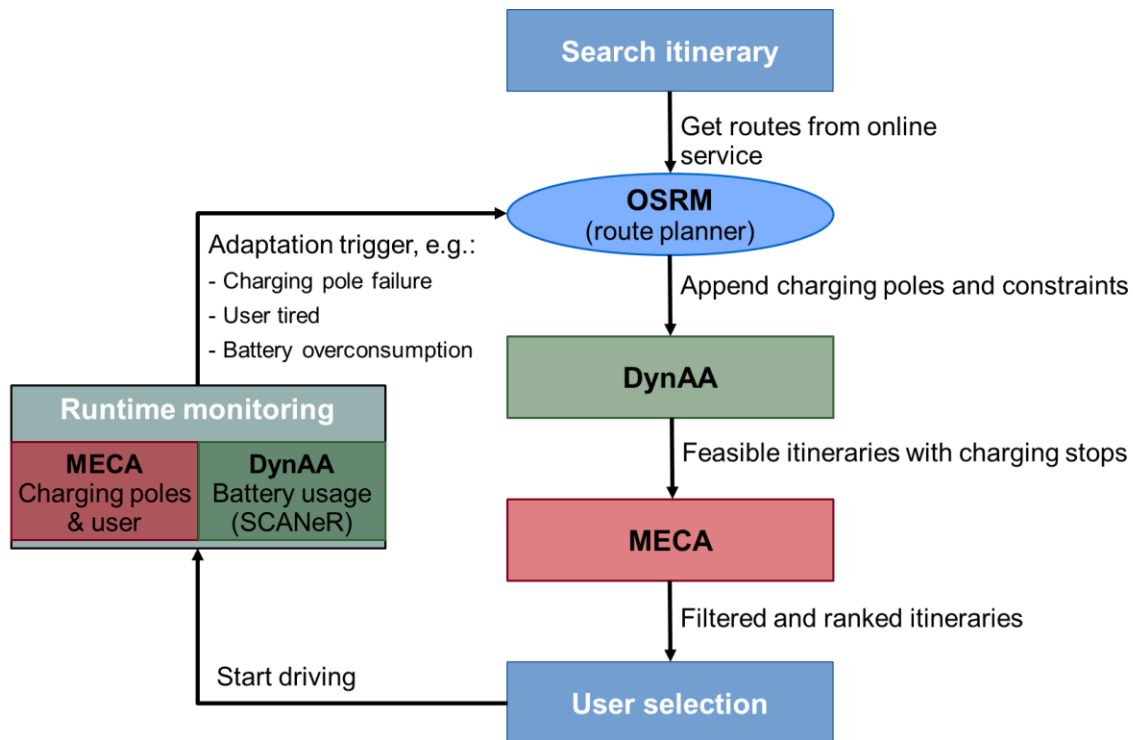


Figure 3-3: Itineraries generation and validation loop

To design the required functionalities and interfaces, it was important to first define the data structure and components of itineraries. In Figure 3-4 route is specified, where to each segment or point specific info can be attached (like declination to a segment and opening

WP6 – D6.10: Smart Travelling Demonstrator

times to a charging point) which can be used in finding the overall best solution (taking into account all relevant KPIs)

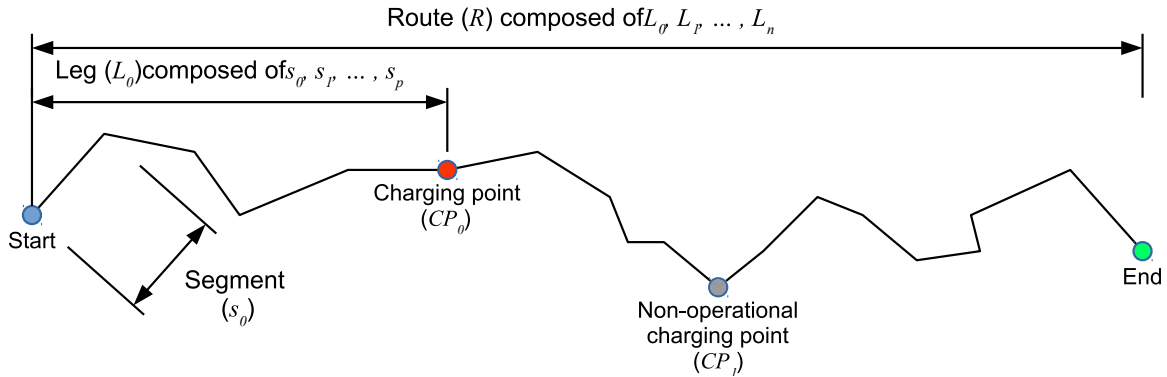


Figure 3-4: Route definition

For some KPIs the system will collect user preferences, which can be used to select or rank specific solutions in order of preference. In the planning phase the flow depicted in Figure 3-5 is executed. Here the interface calls between the different tools are indicated with numbers. An important task for the M18 demonstrator was the development and test of these interfaces between the tools.

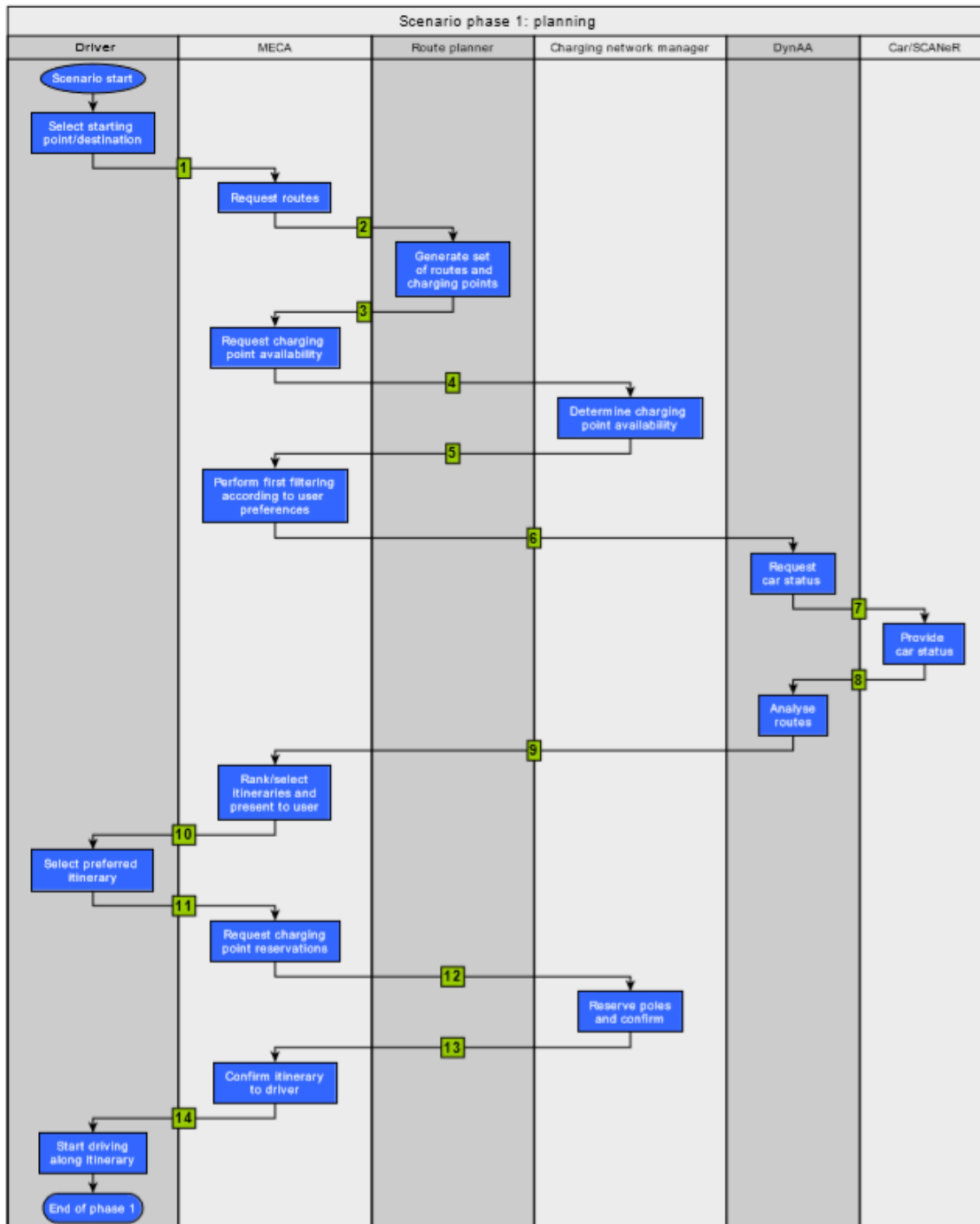


Figure 3-5: Phase 1: Planning

The calculation of the itineraries in the planning phase is performed using following transformation:

WP6 – D6.10: Smart Travelling Demonstrator

- **User input** – user defines starting point A, destination B, preferred departure and arrival times;
- **Route planning** – it calculates the fastest routes (R) from A to B comparing the available options, as an example please consider Figure 3-6;

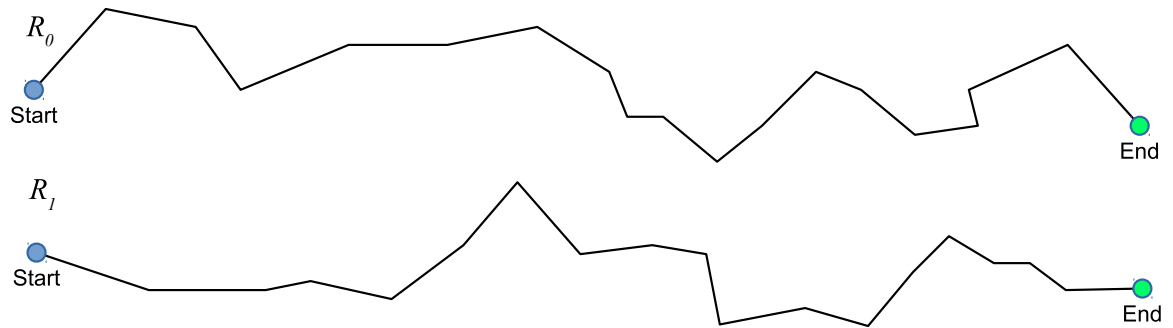


Figure 3-6: Two different routes from A to B

- **Charging point addition** - charging points (CP) along the calculated routes should be connected to points in the route with additional info like availability (opening times) and costs added to the charging points. The calculated routes are called *itineraries* within the use case, as soon as additional info (like charging poles) is added to the route, as an example please consider Figure 3-7

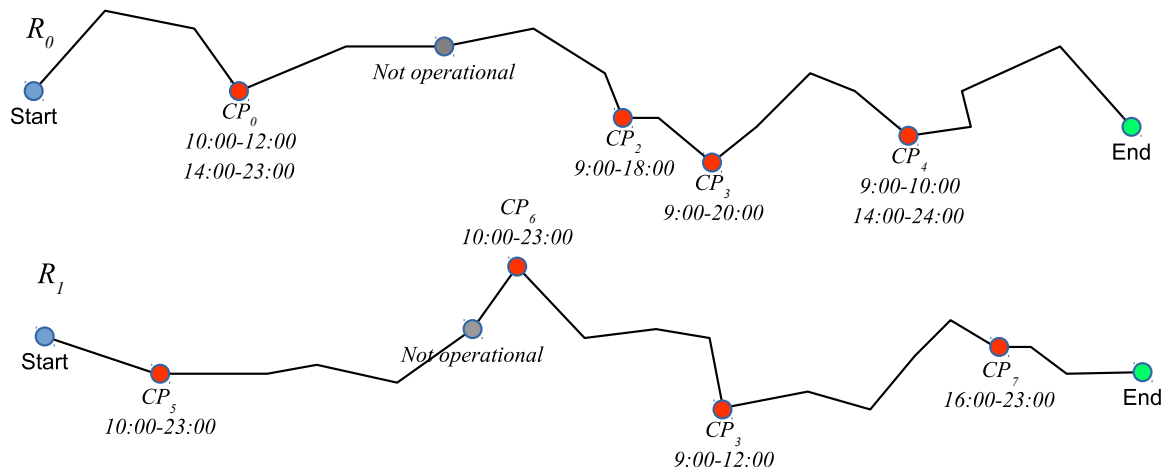
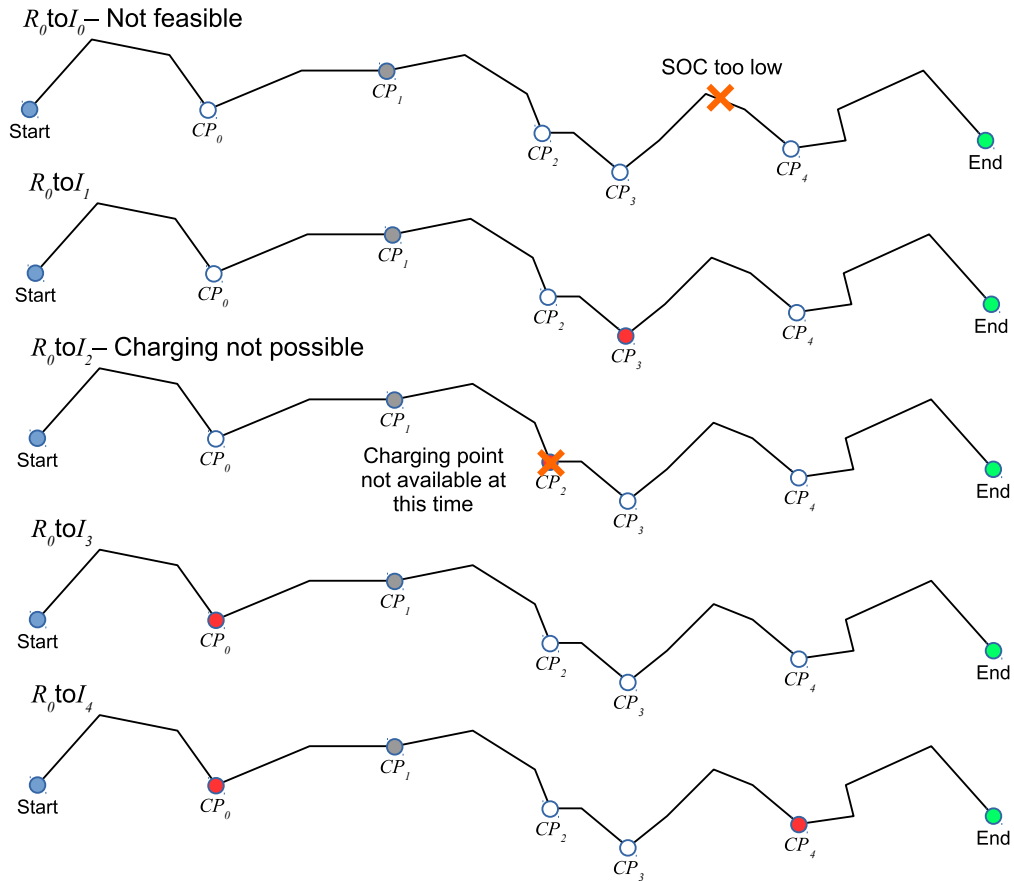


Figure 3-7: Additional of charging points along the itineraries

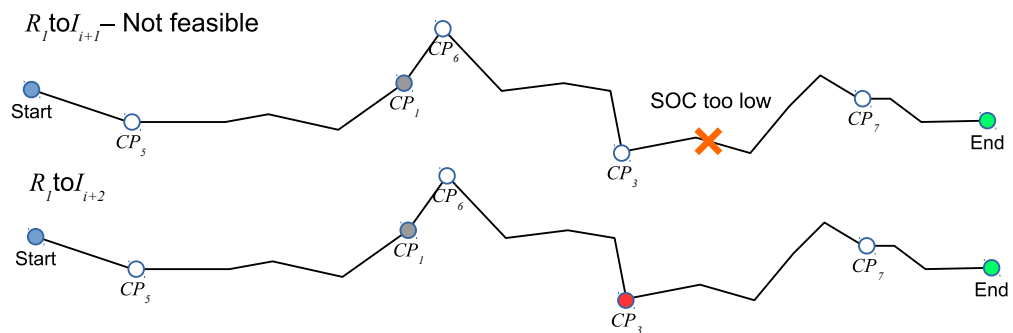
- **Itinerary computation** – of all the route information (including the charging points and connected information) are sent to DynAA, which simulates all given routes

WP6 – D6.10: Smart Travelling Demonstrator

and verify if constraints can be met, based on the current battery level (see Figure 3-8). In case state of charge (SOC) becomes too low, an itinerary (I) is not feasible;



..etc, continuing until itineraries for this route...



..etc...

Figure 3-8: Computation of KPI results by simulating driving all itineraries

- **Ranking of routes** - based on calculations performed by DynAA and user preferences contained in MECA all the different routes are ranked as shown in

WP6 – D6.10: Smart Travelling Demonstrator

Figure 3-9. In Figure 3-13 a scoring graph is shown indicating the level in which specific KPIs are fulfilled;

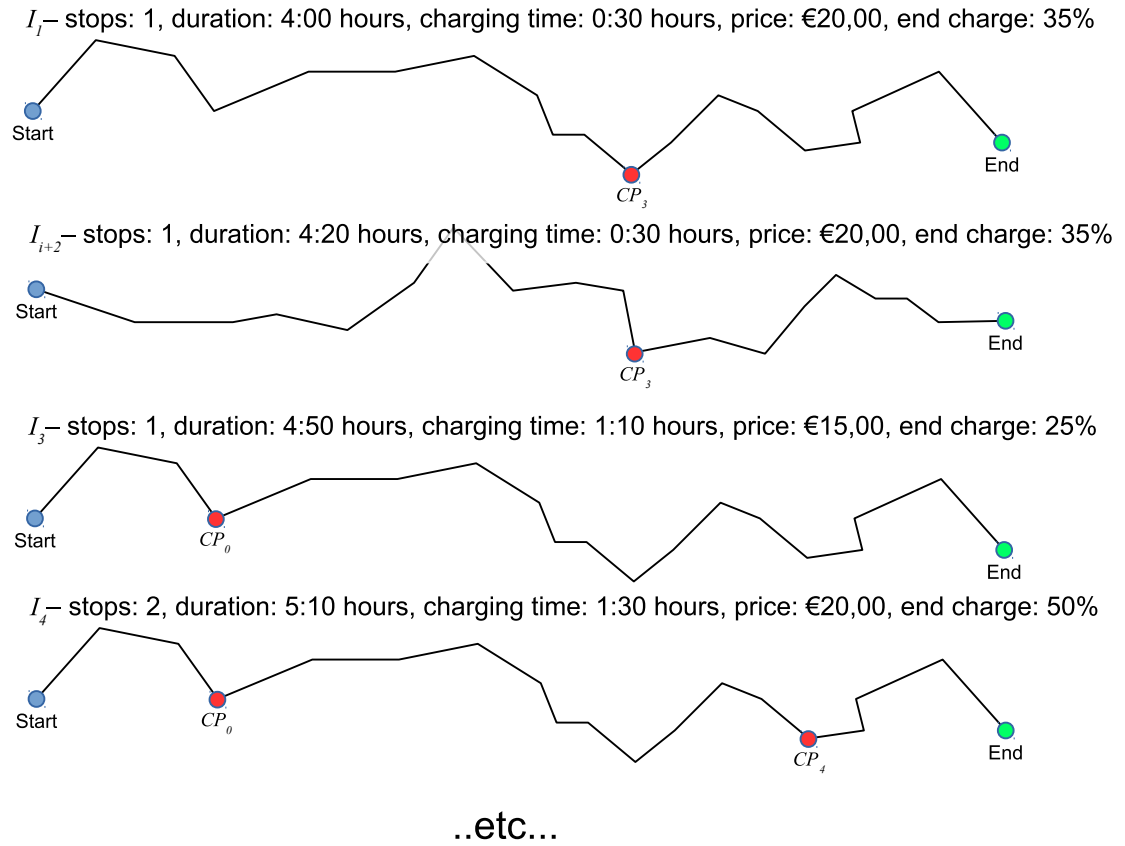


Figure 3-9: Ranked itineraries

In phase 2 the system will respond to off-nominal events, which could be internal events or external events. In the test scenario the external event “pole out of order” received from the charging network operator was defined, which will result in re-planning of the route based on current position of the car (see Figure 3-10).

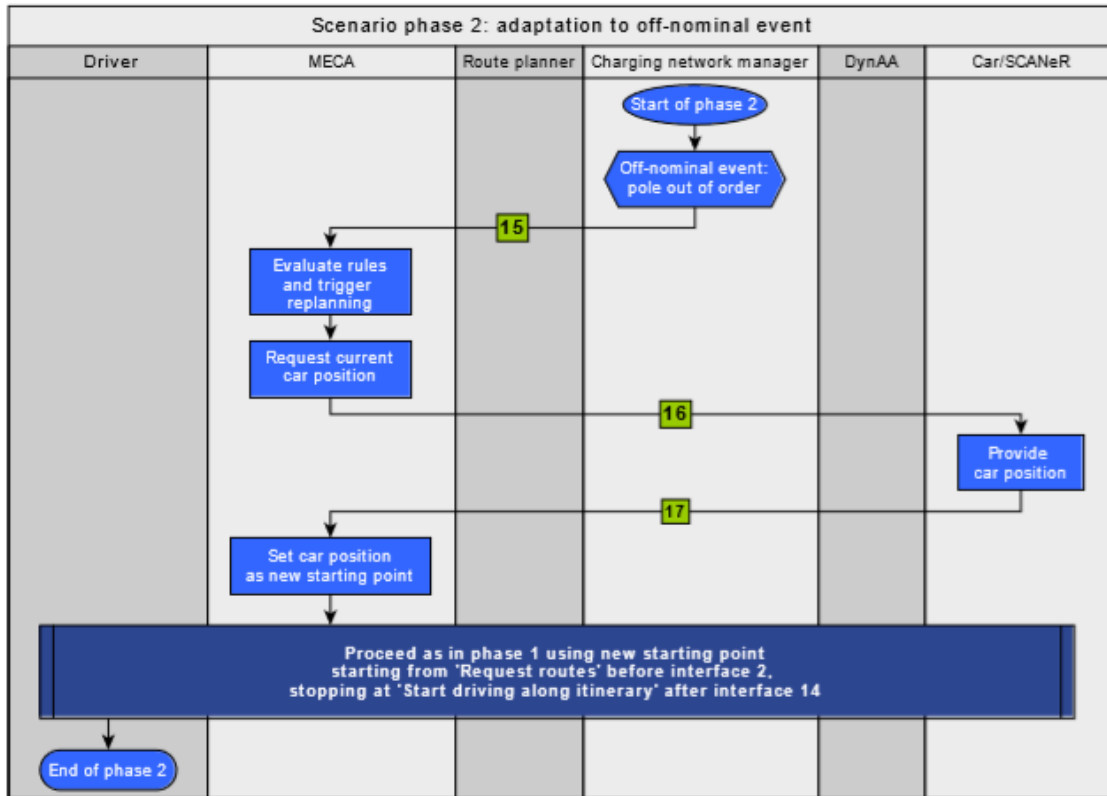


Figure 3-10: Phase 2: Adaptation to off-nominal event

As user interaction with the driver of the car is important in the ST demonstrator, specific user interface screens are used, through them MECA interacts with the driver. In following pictures (Figure 3-11 and Figure 3-12) samples of the developed interface screens are shown. They are used to present the itineraries to the user and to retrieve preferences and selections.

WP6 – D6.10: Smart Travelling Demonstrator

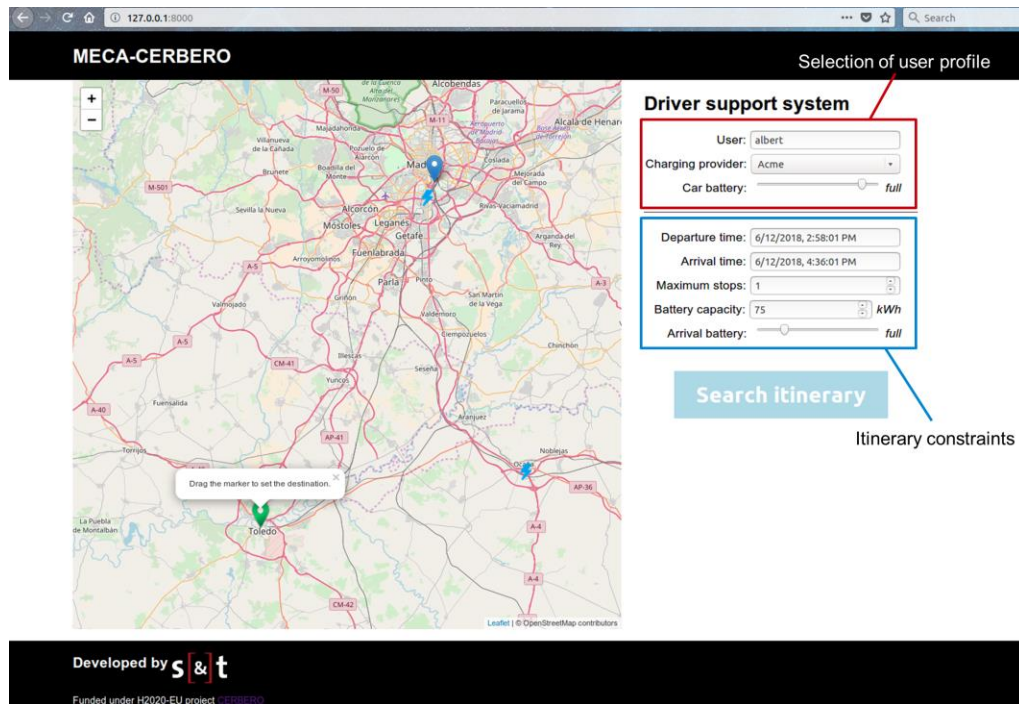


Figure 3-11: Driver support screen

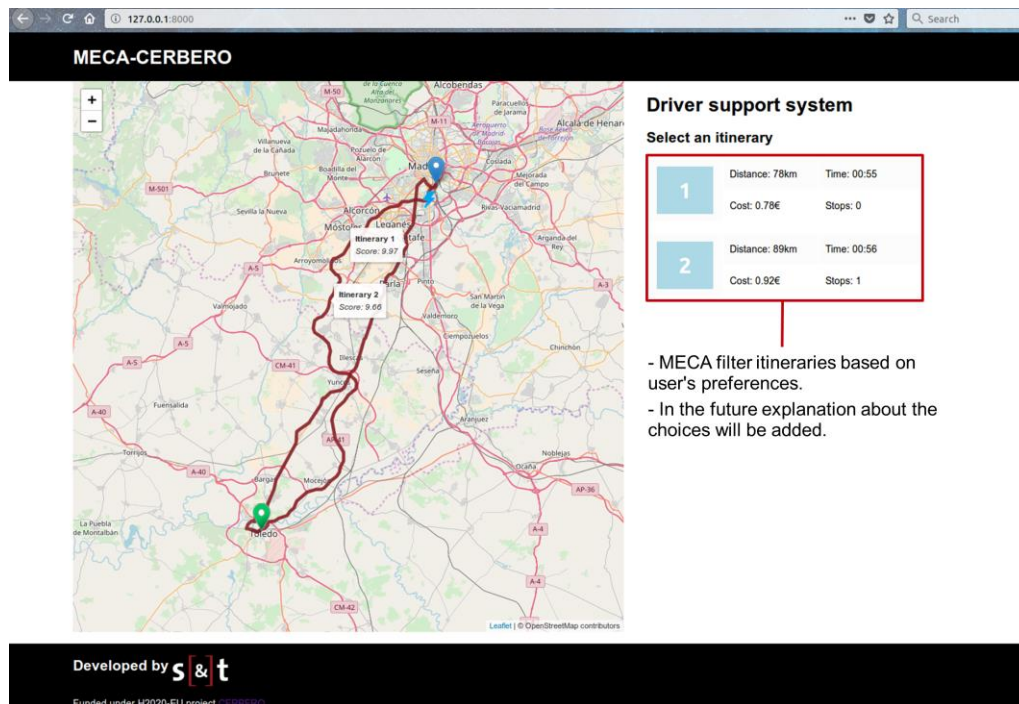


Figure 3-12: Screen with calculated itineraries

In Figure 3-13 multiple KPIs scoring of a specific itinerary is shown. The system can use stored user preferences to rate and filter calculated itineraries. In this way the system will

WP6 – D6.10: Smart Travelling Demonstrator

already suggest best alternative(s) to the user, where the user can make final choice on which itinerary to select.

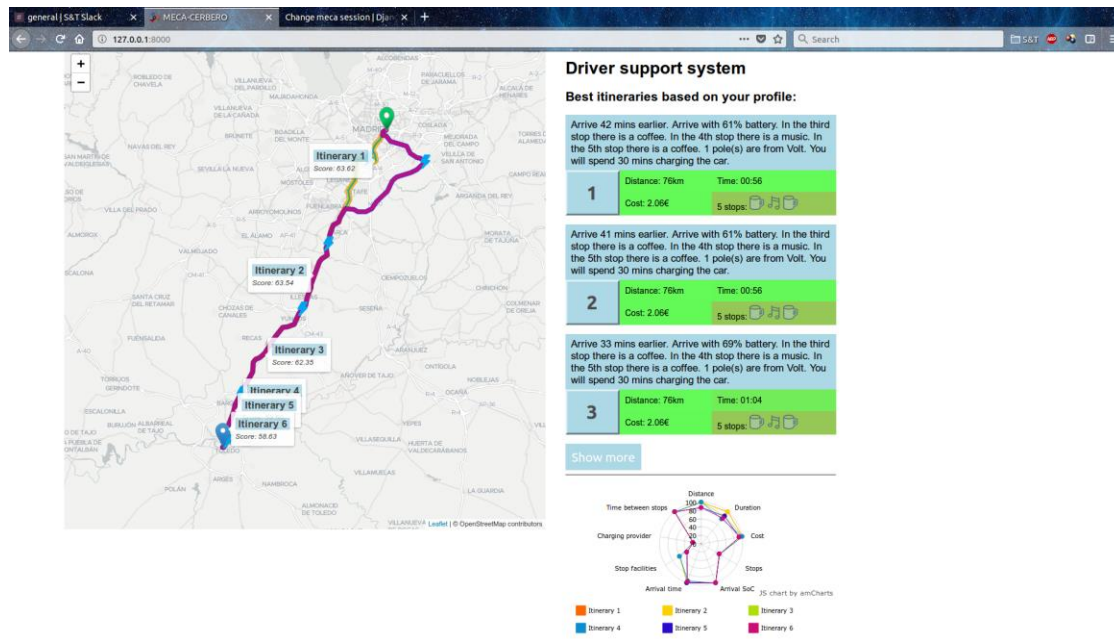


Figure 3-13: Screen with multiple KPIs scoring of specific itinerary

The implementation of the M18 demonstrator is performed in following steps:

- Create test scenarios with all relevant functionalities (based on scenarios defined in D2.4 and D2.3), see [ST-TEST-SCENARIO];
- Specify data formats (e.g. specification of routes) and interfaces (SCANeR-DynAA, DynAA-MECA, SCANeR-MECA);
- High level design of process and related DynAA (TNO) and MECA (S&T) functionalities to support the defined test scenario;
- Implement Smart Travelling functionalities in DynAA (TNO) and MECA (S&T);
- Update TNO battery and motor models to support predictions scenarios with integration in DynAA;
- Module tests (SCANeR and DynAA by TNO and MECA by S&T);
- Integration test (SCANeR, DynAA and MECA) in TNO test platform.

WP6 – D6.10: Smart Travelling Demonstrator

- Execute adaptation test scenario in demonstrator (and possibly record execution in video for later presentation).

3.1.3. Development of the data fusion application

As CRF was confronted with non-synchronized logging files from different parts of the simulator, it was very difficult and time consuming to analyze the results of performed test runs. Because of this CRF and TNO decided to develop a separate data fusion application that can be used to fuse and synchronize the data from the existing simulator components and the new logfiles that will be produced by the new CERBERO modules.

The implementation is performed in following steps:

- Implementation of data fusion application at TNO location.
- Test of the application via sample logfiles from CRF and files generated by the new demonstrator components. It is expected that the data fusion application can be deployed in the operational CRF simulator environment directly after the M18 deadline (for the existing CRF components).

3.1.4. Prepare CERBERO intermediate format interfaces

As the final goal of the project is to develop CERBERO toolchain, it is important that the interfaces developed between the tools will not be use case specific, but generic enough to be used among different components of the CERBERO framework or even applied to other use cases in the future. For this purpose it is important to use the CERBERO intermediate format between the CERBERO tools (also see D5.7 and section 3.1.3 in D3.6). As the intermediate format of CERBERO is only partially developed, TNO started with experimenting with intermediate format solution as discussed with other partners during the last General Assembly session in Haifa.

In parallel to the two demonstration parts (self-adaptation and the data fusion application) an initial proof-of-concept version of the CERBERO intermediate format software will be implemented (to be integrated in the M36 Smart Travelling demonstrator and optionally also for the demonstrators of the other use cases).

3.2. Integrated and Evolved Tools

The tools used for the development of the demonstrator are standard mostly Java based development environments. In the Smart Travelling use case the CERBERO tools are thus not used for the design phase.

The CERBERO tools are used however in the deployment phase where the tools provide the required real time simulation (to provide electric vehicle models) and the prediction and user support functionalities (which include user, system and environment driven adaptation).

The CERBERO tools used in the deployment of the M18 demonstrator are DynAA and MECA (also see D2.4 section 2.4).

3.3. Development and deployment environment

The development and deployment environment used for M18 Smart Travelling demonstrator is located at TNO in the Hague (for DynAA and SCANeR) and at S&T in Delft (for MECA), both in the Netherlands.

For the SCANeR tool (see [SCANeR]) special hardware was arranged as SCANeR required high-end graphical hardware in order to run the real time 3D environment simulations. Other parts of the demonstrator, like DynAA, could be run on already available hardware (Windows based laptops and Linux based cloud servers). For flexibility DynAA can also be containerized using Docker (so it can easily be run on Windows and Linux machine supporting Docker). MECA will use a Docker container to run and can be deployed on any of the (Linux or Windows) machines.

For sharing designs, issues and task status and generated code a Gitlab environment was set up by TNO where all relevant material of the M18 demonstrator was stored. By using a shared Gitlab environment, a central repository could be used to properly manage all developed code. In this way all the developers from TNO, S&T and CRF were able to easily share issues, development plans, documents and code.

WP6 – D6.10: Smart Travelling Demonstrator

We plan to move the developed and tested demonstrator modules (including the DynAA and MECA tools) to the CRF simulator environment in Turin after the M18 milestone to verify if the software is compatible with the software deployed in the real CRF simulator.

4. Tests, Results and Feedback

4.1. Tests

For testing and validating the M18 Smart Travelling demonstrator a test scenario was developed (see [ST-TEST-SCENARIO]), containing all essential functionalities defined in the three use case scenarios defined in D2.3 and D2.4. As the focus of the M18 demonstrator was to develop the skeleton of modules, no specific work was done to develop end to end test scenario based on the use case scenarios. Instead only small parts of the complete scenario are executed as test to verify the implemented functionalities.

Our plan is to execute parts of the scenario essential to verify the developed functions and interfaces of the M18 demonstrator. For the M36 demonstrator complete end to end scenarios will be executed in order to verify developed functions and interfaces.

As development of the M18 demonstrator is split up in four parts, for each part specific tests will be performed.

- The test of the real time system-in-the-loop simulation is combined with test scenario for self-adaptation (as both need battery and motor simulations).
- For the data fusion application a number of test are performed to verify the correctness of the fusion functionalities (including specified interpolations for specific data series).
- For the test of intermediate format (proof-of-concept) implementation, some sample data and queries are used to verify the mechanism.

4.2. Test Results

In the table below the results from the demonstrator development activity were added (in *italic*) to the expected results as specified in D6.7.

Table 4-1 –Smart Travelling Goals and Results

ID	Goal	Results M18 demonstrator
ST1	Development of parametric, modular and extendable physical-cyber co-simulation environment.	<p>Reduction of costs, increase of reuse of software components in different simulation scenarios.</p> <ul style="list-style-type: none"> - <i>Addition of a co-simulation module to DynAA for system-in-the-loop simulations.</i> - <i>The same TNO models for battery and motor were used in both the driving simulation mode and the planning (route prediction) mode.</i> - <i>The implemented planning and adaptation phases can be used in all scenarios involving planning and planning adaptation..</i>
ST2	Development of an integrated open toolchain for design space exploration and co-simulation, with system-in-the-loop capabilities.	<p>Reduce time of development, verification, integration, along with the related costs, exploiting a library of reusable components/metrics integrated by common framework in different levels of abstraction. Incremental prototyping.</p> <ul style="list-style-type: none"> - <i>The TNO battery and motor models can be reused in all Electric Vehicle simulations.</i> - <i>DynAA has been currently integrated with MECA to test user preferences and user-triggered adaptivity, while system-in-the-loop co-simulation extension allows system awareness.</i>
ST3	Development of a self-adaptation.	<p>Efficient support of functional adaptivity, according to system, human and environment triggers.</p> <ul style="list-style-type: none"> - <i>Based on MECA and DynAA an adaptation manager was developed, which could trigger adaptation of the generated advice based on any internal or external event.</i>

4.3. Feedback

To be able to implement scenarios defined in D2.4 we first needed to set up the skeleton of tools. Based on the high level functionalities defined in D2.4, a Smart Travelling test scenario was created (see [ST-TEST-SCENARIO]) containing some more details on the scenario to allow better specification and design of the required functionalities and interfaces.

Before CERBERO tools could be adapted and extended to provide the planned functionalities, some effort was needed to set up the SCANeR simulation environment. As an important part of the SCANeR software is involved with real time 3D environment rendering, very specific hardware was required. As SCANeR is a commercial product, a test license needed to be arranged to be able to perform initial test and development work in the TNO test environment (as no specific test and release facilities exist at the CRF location).

As S&T has used Docker to containerize their MECA tooling, it was quite simple install and run MECA in the TNO test environment. Furthermore TNO and S&T were able to closely work together on the demonstrator by using Gitlab as a code sharing environment and by using a direct network connection between the S&T development site and the TNO test environment. To provide needed user interaction a use interface was developed to provide the basic interaction.

Based on the development performed on the M18 demonstrator it is advised to make a more detailed description of the Smart Travelling use case scenario in D2.4 CERBERO Scenarios Description – Final version (using further insights on adaptation, charging network optimization and required user interaction).

The current D2.4 document does not contain sufficient level of detail to develop more elaborate M36 demonstrator which can be used to execute the defined use case scenarios.

5. Conclusion

Given the complexity of the CRF car simulation environment and the limited timescale, the project was able to successfully develop and implement a preliminary version of the demonstrator using part of the upper part of the CERBERO framework, in particular DynAA and MECA integrated chain.

Optimization and adaptability required the development of route calculation and evaluation functionality capable of handling multiple KPIs in parallel. This functionality has been developed and integrated into both MECA (for initiating the route planning and evaluating the results) and DynAA (for simulating and calculating possible routes).

Most important KPIs addressed in the M18 demonstrator were:

- Response time to triggers (especially important for the real time simulator mode execution the TNO battery and motor models);
- Energy (needed to perform the trip from origin towards destination);
- Cost (mainly the cost of energy obtained from the charging points);
- Travel time (of the route(s) towards the destination, including time to charge).

Additional KPIs and use of driver preferences will be added in the M36 demonstrator.

Although the M18 demonstrator does not yet contain all the required functionalities, it was possible to successfully run parts of the test scenario (see [ST-SCENARIO]) with this first demonstrator. As the same software version of SCANeR is run in the TNO test environment, it is expected that the CERBERO functions can easily be integrated into the CRF simulator. This however will need to be tested and validated in the following phase of the project.

In order to validate the software against the initially defined use case scenarios (see D2.4), the demonstrator will need to be further extended with additional functionalities. Special attention is needed to interface the tools in a more generic way (e.g. using the CERBERO intermediate format).

Also additional optimization of charging infrastructure using AOW is planned to better support the different driving use case scenarios.

6. References

- [CERBERO 2018] <http://www.cerbero-h2020.eu>
- [ST-TEST-SCENARIO] “CERBERO Smart Travelling for Electric Vehicles – Scenario definition”, S&T, 2018.03.13.
- [D2.7] CERBERO D2.7 Technical Requirements (Ver. II)
- [D2.4] CERBERO D2.4 Description of Scenarios (Ver. II)
- [D3.4] CERBERO D3.4 Modeling of KPI (Ver. I)
- [D3.6] CERBERO D3.6 Cross Modelling Methodology for CPS (Ver. I)
- [D5.6] CERBERO D5.6 Framework Components (Ver. I)
- [D6.7] CERBERO D6.7 Demonstration Skeleton (Ver 1)
- [SCANeR] Automotive simulator software developed by OKTAL:
<http://www.oktal.fr/en/automotive/range-of-simulators/software>