D4.3

Translation of traffic management measures to iCS, scale-up, and wider deployment

Project Acronym	TransAID	
Project Title	Transition Areas for Infrastructure-Assisted Driving	
Project Number	Horizon 2020 ART-05-2016 – GA Nr 723390	
Work Package	WP4 Traffic Management Procedures for Transition Areas	
Lead Beneficiary	Transport & Mobility Leuven (TML)	
Editor / Main Author	Sven Maerivoet	TML
Reviewers	Alejandro Correa	UMH
Dissemination Level	PU	
Contractual Delivery Date	M30 (Feb20)	
Actual Delivery Date	M33 (May20)	
Version	v1.7	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 723390.

Document revision history

Version	Date	Comments
v0.1	15/01/2019	Initial draft version with proposed structure
v0.2	05/02/2019	Alignment with WP6
v0.3	12/02/2019	Added application development in the iCS
v0.4	25/02/2019	Added how-to-testing of the ported applications
v0.5	26/02/2019	Added information on the iTETRIS modular system
v0.6	28/02/2019	Small corrections Added conclusions
v1.0	28/02/2019	Final peer-reviewed version
v1.1	25/07/2019	Added an executive summary
v1.2	10/02/2020	Added deployment of dynamic traffic assignment
v1.3	01/04/2020	Updated third-party intermediary service aspects
v1.4	07/04/2020	Updated glossary Restructured Chapter 4 Major review of all sections
v1.5	24/04/2020	Integration of all revisions Restructured document into first and second iterations
v1.6	29/04/2020	Peer-reviewed version
v1.7	14/05/2020	Final peer-reviewed version

Editor / Main author

Sven Maerivoet (TML)

List of contributors

Kristof Carlier (TML)
Péter I. Pápics (TML)
Bart Ons (TML)
Robert Alms (DLR)
Yun-Pang Flötteröd (DLR)
Leonhard Lücken (DLR)
Evangelos Mintsis (CRT)
Vasilios Karagounis (CRT)
Dimitrios Koutras (CRT)
Anton Wijbenga (MAP)
Jaap Vreeswijk (MAP)
Alejandro Correa (UMH)
Xiaoyun Zhang (DYN)
Robbin Blokpoel (DYN)
Julian Schindler (DLR)

List of reviewers

Alejandro Correa (UMH)
Julian Schindler (DLR)

Dissemination level:

■ PU: Public
□ RE: Restricted to a group specified by the consortium (including the Commission Services)
□ CO: Confidential, only for members of the consortium (including the Commission Services)

Table of contents

1 Executive summary

This deliverable explains how simulations of both the baseline (WP3) and the traffic management schemes (WP4) can be ported from the SUMO simulation environment (with the help of the TraCI interface and Python scripts) to the iCS environment (using the C++ language).

We first gave an explanation on how to set up the creation of a traffic management application in the context of the iCS. Details were given on how to prepare the development of an application, based on the source code in the repository. We also explained the interactions between the iCS, SUMO, ns-3, and the various applications, using subscriptions and the exchange of messages.

To this end, the TransAID version of the iTETRIS platform defined in WP6 includes a basic application known as baseApp that manages the exchange of information between the applications and the iCS modules. The application developed for the different services of the TransAID project will inherit form this baseApp and extend the functionality with the traffic management procedures defined in WP4. In order to develop these applications, a new branch (transaid-apps) is added to the git repository. Note that all TransAID applications developed share the same baseApp. Hence, commits to the baseApp should be strictly separated from the commits to the TransAID applications in development. Changes to the baseApp as well as other iTETRIS modules like iCS or ns-3 should be integrated into the transaid-dev branch.

When porting the traffic management code from the WP4 to the WP6 environments, we need to make sure that the same logic is preserved. In order to guarantee this, all applications implemented in the use cases should create test suites, similarly as described in Deliverable D6.1. We use the same testing framework, called TextTest.

Tests are created in the transaid-apps branch of the repository, separated for each use case individually. All tests are stored in the `transaid/TransAIDScenarios/tests/scenarios` folder. All relevant data pertaining to a specific use case (i.e. SUMO networks, configuration files, …) are copied to the relevant scenario in the tests folder. Just as before, the testing concept employed by TextTest is to compare expected output of an entire program run with actual output (output files or stdout and stderr). However, here we need to be a bit more careful and considerate of the complexity involved with comparing the various iCS traffic management applications to their previously created SUMO counterparts. We explain this via a method of aggregate quantities, rather than explicitly comparing time-space diagrams.

A more detailed comparison of simulation outputs would be to use detector measurements and/or explicit vehicle trajectories, create time-space diagrams from these (of average speeds or flows), and then compare these with each other and define whether or not the deviation is significant. However, even though this type of analysis would certainly allow us to detect deviations in the time-space plane (e.g., congested areas that may appear/disappear as artefacts, …), it would be out of scope. In addition, such analyses have not been done widespread before, as they are also difficult to interpret, and still require some aggregation in order to test these ‘automatically’.

Finally, we provided information on how TransAID can be scaled up and generalised to become a third-party intermediary service provider. We approached this from both a technical and a business-oriented perspective. For TransAID to become part of a complete traffic management system, we focused on the technical side on how to detect TAs, select (and possibly combine) services, and then detect when they are most appropriately timed for deployment. To this end, detection can be done via the infrastructure (e.g., road sensors or even digital communication infrastructure), via the OEMs, or by comparing an infrastructure's newly-defined ISAD level (Infrastructure Support levels for Automated Driving) to the operational design domain (ODD) of the vehicle. From a business-oriented perspective, TransAID requires collaboration with both OEMs and (national) road authority (NRAs). Considering the previous technical challenges (detecting TAs, selecting services, and timing their deployment), TransAID bridges all these parties in such a way that the detection of TAs is performed in a centralised way, and OEMs and RAs have a single point of contact for providing and receiving information about TAs. The information and data flow in this construction is multidirectional, as for example the intermediary service can also provide support to the OEMs by sharing the locations of identified TAs and changes to ISAD. This can then in turn help the OEMs to improve their systems, because they now have specific examples of situations where the limits of their ODDs are reached. And finally, considering the detection of TAs we foresee two different time scales at which TransAID's traffic management service will operate, i.e. a slower changing situation (corresponding to a more strategic level that leads to static transition areas, such as changes to infrastructure or roadside systems), and a faster changing situation (leading to dynamic transition areas (because of changing weather conditions such as fog and heavy snow, incidents, or other specific traffic conditions), requiring timely action).

2 Introduction

In the following sections, we first give a concise overview of the TransAID project, then highlight the purpose of this document, and finally present its structure.

2.1 About TransAID

As the introduction of automated vehicles becomes feasible, even in urban areas, it will be necessary to investigate their impacts on traffic safety and efficiency. This is particularly true during the early stages of market introduction, where automated vehicles of all SAE (Society of Automotive Engineers) levels[1], connected vehicles (able to communicate via V2X, vehicle-to-everything) and conventional vehicles will share the same roads with varying penetration rates.

There will be areas and situations on the roads where high automation can be granted, and others where it is not allowed or not possible due to missing sensor inputs, highly complex situations, etc. Moving between those areas, there will be areas where many automated vehicles will change their level of automation. We refer to these areas as "Transition Areas" (TAs).

TransAID develops and demonstrates traffic management procedures and protocols to enable smooth coexistence of automated, connected, and conventional vehicles, especially at Transition Areas. A hierarchical approach is followed where control actions are implemented at different layers including centralised traffic management, infrastructure, and vehicles.

First, simulations are performed to find optimal infrastructure-assisted management solutions to control connected, automated, and conventional vehicles at Transition Areas, taking into account traffic safety and efficiency metrics. Then, communication protocols for the cooperation between connected/automated vehicles and the road infrastructure are developed. Measures to detect and inform conventional vehicles are also addressed. The most promising solutions are then implemented as real world prototypes and demonstrated under real urban conditions. Finally, guidelines for advanced infrastructure-assisted driving are formulated. These guidelines also include a roadmap defining activities and needed upgrades of road infrastructure in the upcoming fifteen years in order to guarantee a smooth coexistence of conventional, connected, and automated vehicles.

Iterative project approach

TransAID performed its development and testing in two project iterations. Each project iteration lasted half of the total project duration. During the first project iteration, the focus was placed on studying Transitions-of-Control (ToCs) and Minimum-Risk Manoeuvres (MRMs) using simplified scenarios. To this end, models for automated driving and ToC/MRM were adopted and developed. The simplified scenarios were used for conducting several simulation experiments to analyse the impacts of ToCs at transition areas (TAs), and the effects of the corresponding mitigating measures.

During the second project iteration, the experience accumulated during the first project iteration has been used to refine/tune the driver models and enhance, extend, and even combine some of the proposed mitigating measures, while also increasing the complexity and realism of the tested scenarios.

[1] https://www.sae.org/misc/pdfs/automated_driving.pdf

2.2 Purpose of this document

This deliverable D4.3 relates to task T4.3 of WP4, and explains how simulations of both the baseline (WP3) and the traffic management schemes (WP4) are to be ported from the SUMO simulation environment (with the help of the TraCI interface and Python scripts) to the iCS environment (using the C++ language). The latter code transformation step is necessary, as it gives us a concrete outlook for further developments and deployment of the TransAID approach, even after the project's lifetime.

2.3 Structure of this document

The document starts with an explanation on how to set up the creation of a traffic management application in the context of the iCS. This entails the preparation to develop an application (which is what the traffic management schemes are called within the iCS), i.e. which files need to be created where within the software repository. Once this has been done, the document explains how testing schemes are to be defined for WP6. This is needed to check if the implemented traffic management logic complies with the previously obtained results from WP4. Note that the required communication protocols are, at this point, already elaborated in WP5. Next, the document provides information on how TransAID can be scaled up and generalised to become a third-party intermediary service provider, from both a technical and a business-oriented perspective. The former part elaborates on the various ways for the detection of transition areas, and the latter deals with the possible collaborations with both OEMs and (national) road authority.

2.4 Glossary

CAM	Cooperative awareness message
COP	Common operational picture
CPM	Cooperative perception message
FCD	Floating-car data
iCS	iTETRIS control system
ISAD	Infrastructure Support levels for Automated Driving
LDD	Loop-detector data
MCM	Maneouvre coordination message
(N)RA	(National) road authority
ODD	Operational design domain
RSU	Road-side unit
SAE	Society of Automotive Engineers
SUMO	Simulation of Urban Mobility
TA	Transition area
TMC	Traffic management centre
TraCI	Traffic Control Interface
TransAID	Transition Areas for Infrastructure-Assisted Driving
V2X	Vehicle-to-everything
WP	Work package

3 Setting up traffic management implementations

The previous simulations of both the baseline (WP3) and the traffic management schemes (WP4) were directly implemented for the SUMO simulation environment with the help of the TraCI interface and Python scripts to set up multiple runs (to smooth out random variations and obtain stable results).

After these preliminary simulations and evaluations of traffic management strategies, the traffic management logic now has to be translated to the C++ language to be used in the iTETRIS platform. The work provides a direct input – and runs partly in parallel – to WP6, where the full integrated platform for the simulation and the assessment of traffic management procedures is used. In that sense, the work in Task 4.3 of WP4 is dedicated to generating code that can be used for simulations in WP6.

In the following sections, we first look at how to set up the creation of a traffic management application in the context of iTETRIS, and then describe how tests are to be defined for WP6 to check the performance of the implemented traffic management logic compared to previously obtained results from WP4.

Note that the structure of iTETRIS was already explained in Deliverable D6.1[2]. Here, we provide extra information for porting the existing SUMO applications to the iCS that was not present in the aforementioned D6.1.

3.1 Develop an application

The TransAID version of the iTETRIS platform defined in WP6 includes a basic application known as baseApp that manages the exchange of information between the applications and the iCS modules. The application developed for the different services of the TransAID project will inherit form this baseApp and extend the functionality with the traffic management procedures defined in WP4. In order to develop these applications, a new branch (transaid-apps) is added to the git repository. Note that all TransAID applications developed share the same baseApp. Hence, commits to the baseApp should be strictly separated from the commits to the TransAID applications in development. Changes to the baseApp as well as other iTETRIS modules like iCS or ns-3 should be integrated into the transaid-dev branch.

3.1.1 Building the project

This goes similar to the instructions from iTETRIS_quick_installation_ubuntu.txt (branch transaid-dev), which have been explained in Deliverable D6.1.

Building with clang for extended error messages:

To configure an clang++ build for the iCS, which reports a lot of additional warnings, we can use the following command:

```
./configure CXX=clang++ CXXFLAGS="-stdlib=libstdc++ -
fsanitize=undefined,address,integer,unsigned-integer-overflow -fno-
omit-frame-pointer -fsanitize-
blacklist=$SUMO_HOME/build/clang_sanitize_blacklist.txt" --
prefix="$PWD/../.."
```

[2] An integrated platform for the simulation and the assessment of traffic management procedures in Transition Areas (2018). TransAID Deliverable D6.1

To suppress the repeating warnings from the mersenne-twister, add -Wno-deprecated-register:

```
./configure CXX=clang++ CXXFLAGS="-stdlib=libstdc++ -
fsanitize=undefined,address,integer,unsigned-integer-overflow -fno-
omit-frame-pointer -fsanitize-
blacklist=$SUMO_HOME/build/clang_sanitize_blacklist.txt -Wno-
deprecated-register" --prefix="$PWD/../.."
```

3.1.2 Setup/Preparation

To start with an application for a specific use case, we can use commits b658a946[3] and ba933001[4] (and also 913d7b58[5]) as a template and rename the occurrences of uc1with the appropriate use case (e.g., uc5). A helpful snippet for thethis to be executed in the transaidUCApp directory:

```
find . -type f -exec sed -i -- 's/UC1/UC5/g' {} +
find . -type f -exec sed -i -- 's/uc1/uc5/g' {} +
```

(this replaces all occurrences of uc1 in the sources with uc5.)

3.1.3 Adapt applications to iTETRIS

As previously mentioned, the earlier simulations of WP3 and WP4 where developed employing the SUMO simulation environment and Python scripts that interact with SUMO employing the TraCI interface. These scripts need to be modified in order to implement the traffic management measures of the different services within the iTETRIS platform. The latter coordinates the applications with the traffic simulator SUMO and the communications simulator ns-3 employing the iCS module (see Figure 1). To do so, the iCS module of the iTETRIS platform coordinates the execution of the different simulators and the applications. Following a cyclic sequence, the iCS commands to the connected modules (i.e. SUMO, ns-3, and the applications) to execute the simulation for a given time period known as a simulation step. Once each connected module is executed, the next iteration of the simulation is prepared in the iCS by updating the connected modules with the information of the other connected modules (i.e if a new node is created in SUMO, this information is sent to the applications and ns-3 to create also the node in the other modules). This sequential execution implies that the information that an application may request to the iCS in one simulation step, will be sent to the application during one of the next time steps. Thus, the applications must know in advance the information needed at each simulation step in order to request this information from the iCS.

[3] https://gitlab.imet.gr/hit/transaid/commit/b658a946c40db28d960116afae4cdf1f49862a55

[4] https://gitlab.imet.gr/hit/transaid/commit/ba933001f39e3275c89c267fe44a14762e015254

[5] https://gitlab.imet.gr/hit/transaid/commit/913d7b58ba05351693079d0e2ead196e48e83415

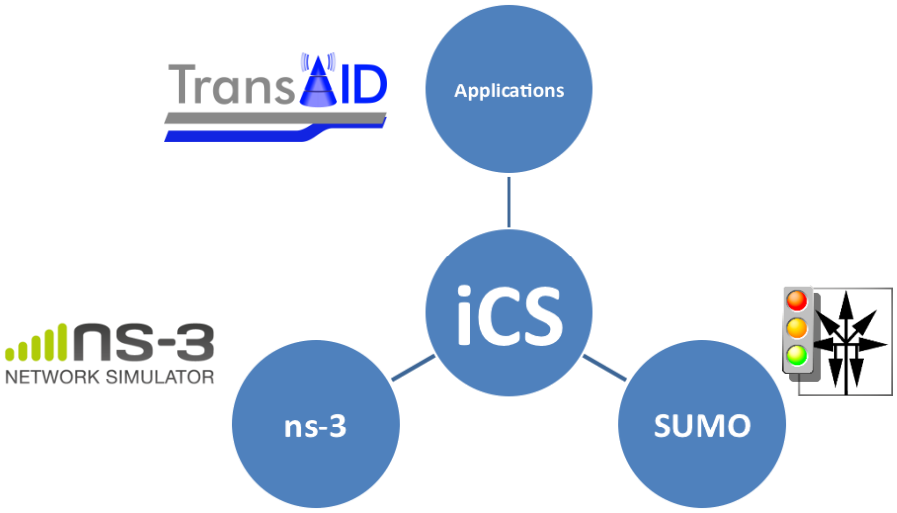

Figure 1: Schematic structure of the iTETRIS platform. The iCS manages the coupling of traffic simulation (SUMO), communications simulation (ns-3), and traffic management applications (diagram reproduced from Deliverable D6.1).

The iTETRIS platform individually simulates the different nodes involved in the simulations, thus instead of having a single application that runs with all the information of all the nodes of the simulation environment, we have different applications for vehicles and for RSUs running. If these applications want to share information, then a message must be sent between the nodes in which these applications are installed. Figure 2 shows the interactions between the TMC and vehicles simulated in the applications layer with the iCS module and the SUMO and ns-3 modules. If the TMC wants to send a message to a vehicle (i.e. a lane change advice) it needs to request from the iCS to schedule the transmission of a message in ns-3. Then, the message will be simulated in ns-3 and if the message is received by the vehicle, the iCS will pass the message to the application of the vehicle. Similarly, if the application of a vehicle wants to modify the behaviour of the vehicle in SUMO, it will need to request to the iCS that sends the command to SUMO. Afterwards, SUMO will modify the behaviour of the vehicle and then it will inform to the TMC application that the behaviour of the vehicle has been modified.

In order to adapt the scripts developed in WP3 and WP4 to the iTETRIS platform, the code will need to be modified to adapt to the event-driven interactions described in iTETRIS. The applications must employ two different types of functions: (i) functions that are executed at every simulation step, and (ii) functions that are executed when an event is triggered. The functions that are executed at each simulation step are the main core of the applications and that is where the traffic management measures of the services are implemented. On the other hand, the applications will also include functions that respond to the occurrence of events triggered by the iCS. These functions will complement the definition of the traffic management measures by for example providing information. Examples of triggering events can be the reception of a CAM message or the detection of a vehicle by a road sensor. In both cases, the information received will be processed and based on this information new action of the application may be triggered. The applications need to use a scheduler function that will schedule the execution of the different functionalities of the application depending on the current simulation step. For example, the application can schedule the periodic request of information to the iCS or schedule the transmission of messages such as the CAM or the CPM.

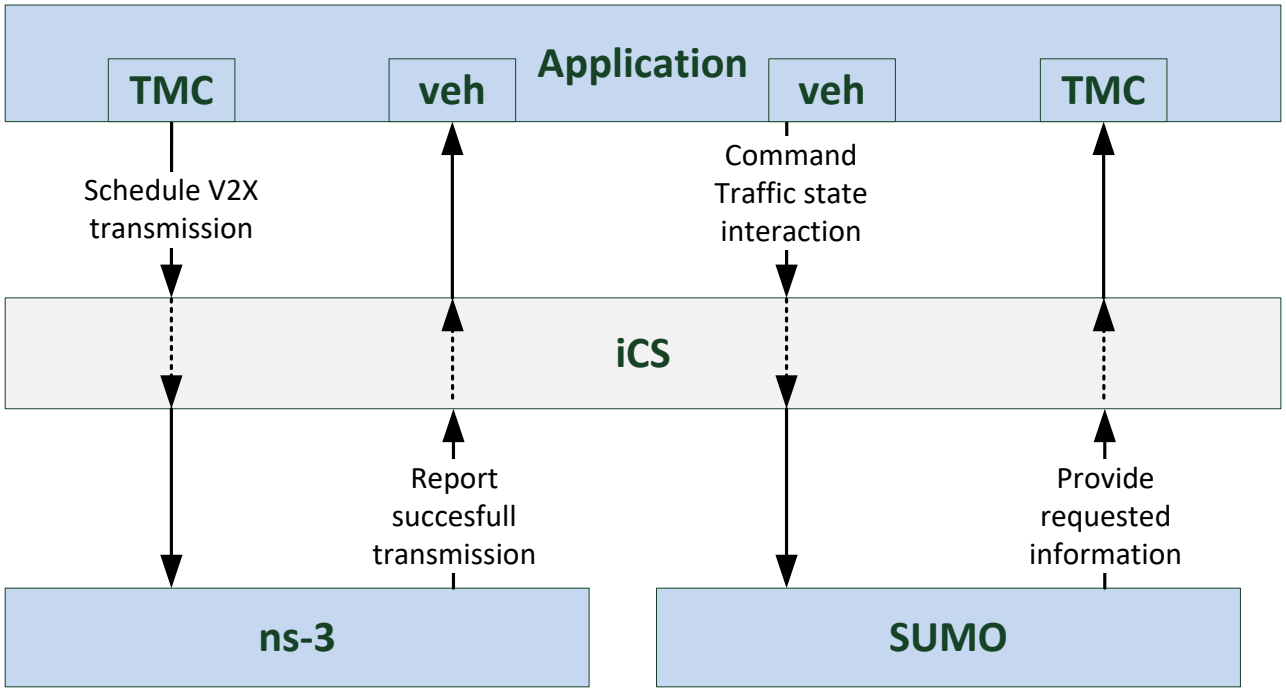

Figure 2: Interactions between the different iTETRIS modules.

In the following sections we describe the main classes and functions employed by the applications and the interfaces defined for the interactions between the applications and the iCS.

3.1.4 The behaviour classes

As of now, three behaviours are used in the applications:

- **RSU Behaviour**: this class is executed in the RSU nodes and includes the functions that model the behaviour of the RSU. This class will obtain information about the traffic stream, define the appropriate traffic management measures according to the TransAID service executed, and send the adequate advices to vehicles in order to execute the defined traffic management measures.

- **Vehicle Behaviour**: this class is executed in the vehicle nodes and includes the functionality of the applications that needs to be executed in the vehicles. This includes basic functionality as the transmission of CAMs and CPMs and also the response of the vehicles to the advices received by the infrastructure.

In the future we can also consider:

- **TMC Behaviour**: this class will manage multiple RSU nodes (as current we work with a single RSU Behaviour node that acts as the TMC and defines the traffic management procedures of the simulation).

The ‘behaviour’ classes inherit from their respective base classes BehaviourNode and BehaviourRsu (located in baseApp/application/model). Here we will put the logic for the application. The behaviours implement different event triggered methods, see next paragraph.

3.1.5 Entry points for actions

The behaviour class defined in the TransAID applications is called from the baseApp at different events. At each one of these events we need to program the response of our applications to the information received.

Initially, the interface calls

- **Start**() - executed at application initialisation. This method must initialise the application and schedule the first actions to be done. For example, at the start of the application the transmission of periodic CAM messages should be scheduled.

Regular events, which are fired on every simulation step are:

- **onAddSubscriptions**() - called at the beginning of each simulation step when the iCS asks the application whether it requires interaction. The interaction between the applications and the iCS are done using subscriptions. This subscriptions can be of different types, for exemple, triggering information retrieval, e.g., calling TraCI-getter methods or commanding the transmission of a messages.

- **Execute**() - called at the end of each simulation step. This is the main function of each application that should coordinate the traffic management measures defined by the application. In the case of an RSU this function must contain the basic functionality of the infrastructure in order to execute the service. Here the behaviour has the possibility to transmit result data back to the iCS, which may be used by different applications. Note that results of TraCI requests (direct information retrieval from the SUMO may be accessed via GetLastTraCIResponse() of the Behavior class).

Truly event-driven methods which are called in between **onAddSubscriptions**() and **Execute**() are:

- **Receive**() - called on reception of a unicast or broadcast message.
- **processCAMmessagesReceived**() - called on reception of CAM messages.
- **processTraCIResult**() - called on reception of a TraCI response requested from onAddSubscriptions() in the current simulation step or from another behaviour method in the previous simulation step.

3.1.6 Actions

The behaviour class can access the iCSInterface, which allows it to interact with the iCS to send messages, retrieve simulation state information or manipulate the simulation. It can be accessed via the method **GetController**(). In the following we list the most important available methods:

One-shot subscriptions:

- Send()
- SendTo()
- AddTraciSubscription()
- AddTraciStop()
- commandTraciOpenGap()
- setTraciParameter()
- requestToC()

Persistent subscriptions:

- StartSendingCAMs()
- receiveCAMs()
- receiveUnicast()
- receiveGeobroadcast()
- requestMobilityInfo()

Other:

- GetLastTraCIResponse() (in Behaviour)
- GetCurrentTimeStep()

3.2 Creating test suites

When porting the traffic management code from the WP4 to the WP6 environments, we need to make sure that the same logic is preserved. In order to guarantee this, alle applications implemented in the use cases should create test suites, similarly as described in Deliverable D6.1. We use the same testing framework, called *TextTest*[6]. An example of running a suit of such tests is shown in Figure 3.

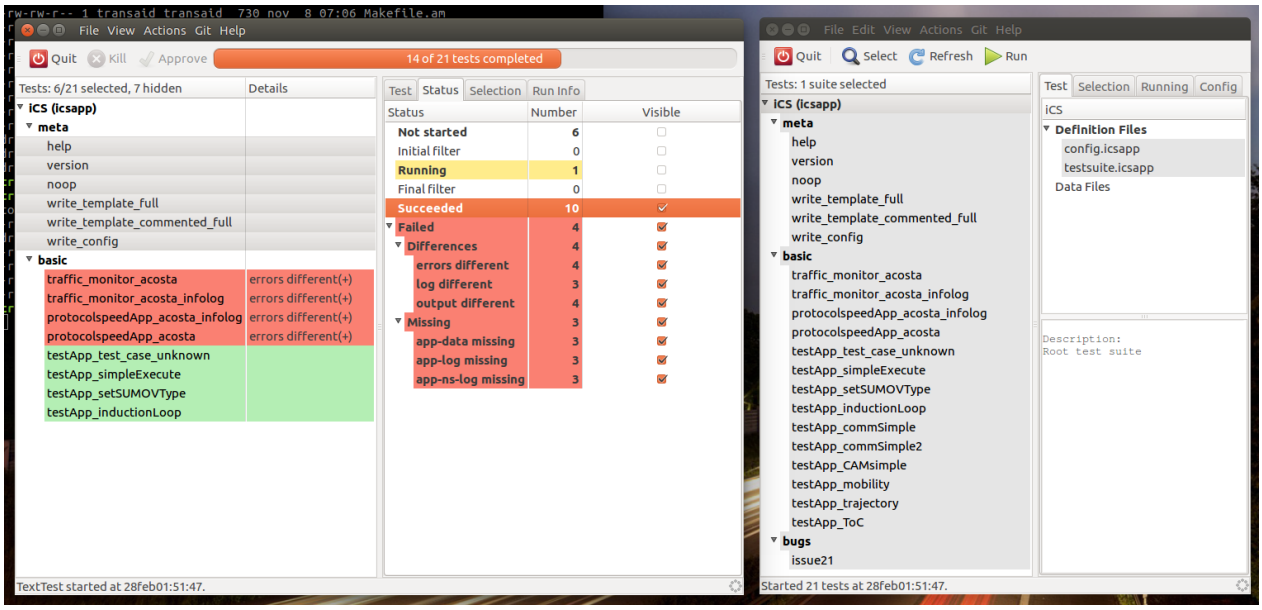

Figure 3: An example of a test suite running in TextTest.

Tests are created in the transaid-apps branch of the repository, separated for each use case individually. All tests are stored in the transaid/TransAIDScenarios/tests/scenarios folder. There, they are split between:

- **baseline**: the scenario without traffic management,
- **TM**: the scenario with traffic management, as implemented in SUMO, and
- **full**: the scenario with traffic management, as implemented in the iCS.

All relevant data pertaining to a specific use case (i.e. SUMO networks, configuration files, …) are copied to the relevant scenario in the tests folder.

Just as before, the testing concept employed by *TextTest* is to compare expected output of an entire program run with actual output (output files or `stdout` and `stderr`). However, here we need to be a bit more careful and considerate of the complexity involved with comparing the various iCS traffic management applications to their previously created SUMO counterparts.

Answering the question "What to test?" yields a plethora of possibilities. The underlying idea however is that what is newly created for the iCS mimics the implementations already done for SUMO. This means that we need a way to compare entire simulation results. The most straightforward way for doing this, is by comparing results on an aggregated level. Each simulation run yields quantities like the average network speed, the total throughput, … These can directly be compared with each other. In this spirit, we should allow some flexibility in interpreting the results, that is to say tests do not fail because an exact comparison yields a false result. Rather, we put a small margin on all obtained quantities, and check whether or not the tested ones fall within these

[6] http://texttest.sourceforge.net/

margins (similar to how floating-point numbers are compared by using an epsilon, albeit a larger one in this case).

A first set of quantities that can be compared, are the ones listed by SUMO's simulation output summary. This output contains the simulation-wide number of vehicles that are loaded, inserted, running, waiting to be inserted, have reached their destination and how long they needed to finish the route. The last value is normalised over all vehicles that have reached their destination so far. Table 1 presents an overview of these quantities.

Table 1: Overview of SUMO simulation output summary.

Quantity name	Type	Description
time	(simulation) seconds	The time step described by the entry.
loaded	#	Number of vehicles that were loaded into the simulation so far (including reported time step).
inserted	#	Number of vehicles inserted so far (including reported time step).
running	#	Number of vehicles that were running within the reported time step.
waiting	#	Number of vehicles which were waiting for insertion (could not be inserted) within the reported time step.
ended	#	Number of vehicles that have reached their destination so far (including reported time step).
meanWaitingTime	s	The mean time all vehicles up to now and within the reported time step had to wait for being inserted; -1 if no vehicle has been inserted, yet.
meanTravelTime	s	The mean travel time of all vehicles that have left the simulation within the previous and the reported time;-1 if no vehicle has been removed from the simulation, yet
halting	#	The number of vehicles in the network with speed below 0.1m/s (which are not waiting at a <stop>).
meanSpeed	m/s	The mean speed over all vehicles in the network (which are not waiting at a <stop>).
meanSpeedRelative	m/s	The mean speed over all vehicles in the network relative to the speed limit (which are not waiting at a <stop>).
duration	ms	The computation time for that simulation step in milliseconds).

A more detailed comparison of simulation outputs, would be to use detector measurements and/or explicit vehicle trajectories, create time-space diagrams from these (of average speeds or flows), and then compare these with each other and define whether or not the deviation is significant. An example of this is a study[7] whereby a first-order macroscopic model was compared to a traffic cellular automaton, comparing the time-space diagrams, as shown in Figure 4.

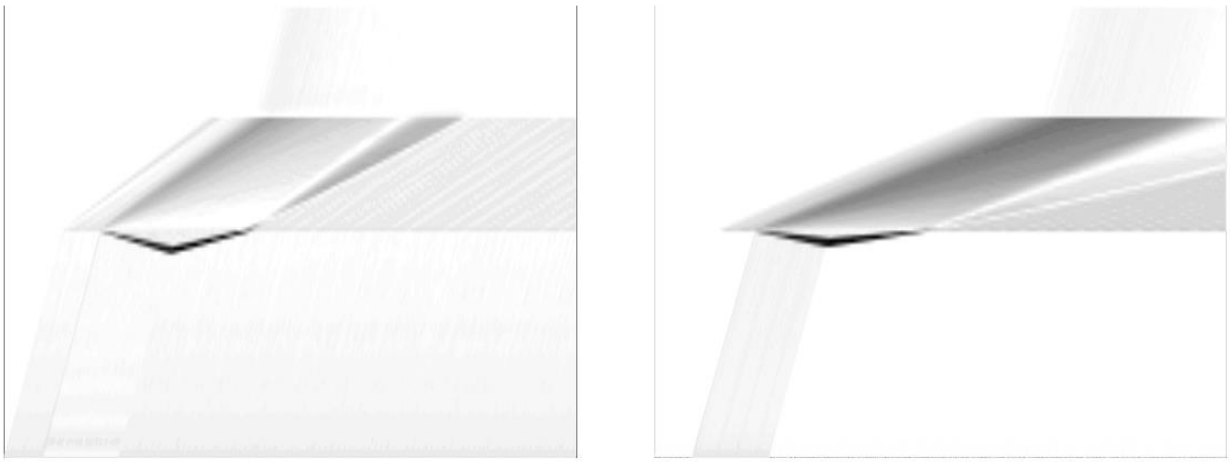

Figure 4: Time-space diagrams showing the differences in local traffic densities for a first-order macroscopic model and a traffic cellular automaton. The modelled road consists of three sections, with the middle one having a lower speed, hence leading to the formation of a congestion region. Darker regions indicate large differences between both modelling approaches (the left and right plots are to illustrate various parametrisations).

Even though this type of analysis would certainly allow us to detect deviations in the time-space plane (e.g., congested areas that may appear/disappear as artefacts, …), it would be out of scope for now. In addition, such analyses have not been done widespread before, as they are also difficult to interpret, and still require some aggregation in order to test these 'automatically'.

[7] Maerivoet S., (2006) Relating the dynamics of the STCA to the LWR model, in Modelling Traffic on Motorways: State-of-the-Art, Numerical Data Analysis, and Dynamic Traffic Assignment, Catholic University of Leuven.

4 Deploying dynamic traffic management

One of the main things that TransAID aspires to achieve is to provide the concrete means to perform successful traffic management under specific conditions, such as those in Transition Areas (TAs). The purpose of this chapter is to indicate and suggest a number of aspects that must be taken into account when further developing TransAID in terms of the technical and organisational side of a business case.

4.1 TransAID in an intermediate role

Most of the work in WP4 during the first and second iteration focused on addressing which types of traffic management could work for a specific use case, and what levels of performance (in terms of throughput, network stability, emissions, and increased traffic safety) can be expected when applying such measures. However, for TransAID to become part of a traffic management framework, we need some ideas on how it can be (1) generalised and (2) scaled up. This intention was also underlying the idea of TransAID as a third-party intermediary service provider, as was expressed in Section 4.2.1 of Deliverable D4.1; to recapitulate:

> *AV-fleet managers and road authorities both operate backend centres to manage their fleets and traffic networks, respectively. To effectively and systematically manage TAs on a large scale and for multiple AV fleets and multiple road authorities, we propose a trusted third party (and where possible mandated)* ***intermediary service****. This will then act as the single-point-of-contact for road authorities and traffic participants (or indirectly, via their OEMs). Based on status and disengagement information from AV fleet managers and traffic management plans from road authorities, this intermediary service acts as a delegated traffic manager who digitally implements the TransAID infrastructure support measures. With support of the right tools, an operator continuously monitors in real-time the traffic system and disengagement reports, based on triggers and scenarios, identifies TAs, and finally selects the appropriate measure. An advantage of this service is that measures taken by AV-fleet managers and road authorities can be coordinated and harmonised across multiple AV fleets and geographical areas (managed by different road authorities). Moreover, smaller and/or rural road authorities, which may not have backend centres or not a suitable operational overview of the road and traffic flow dynamics, can benefit from an intermediary service that can perform this task for them. The concept of the intermediary service approach adopted within TransAID's traffic management scheme is depicted in Figure 5.*

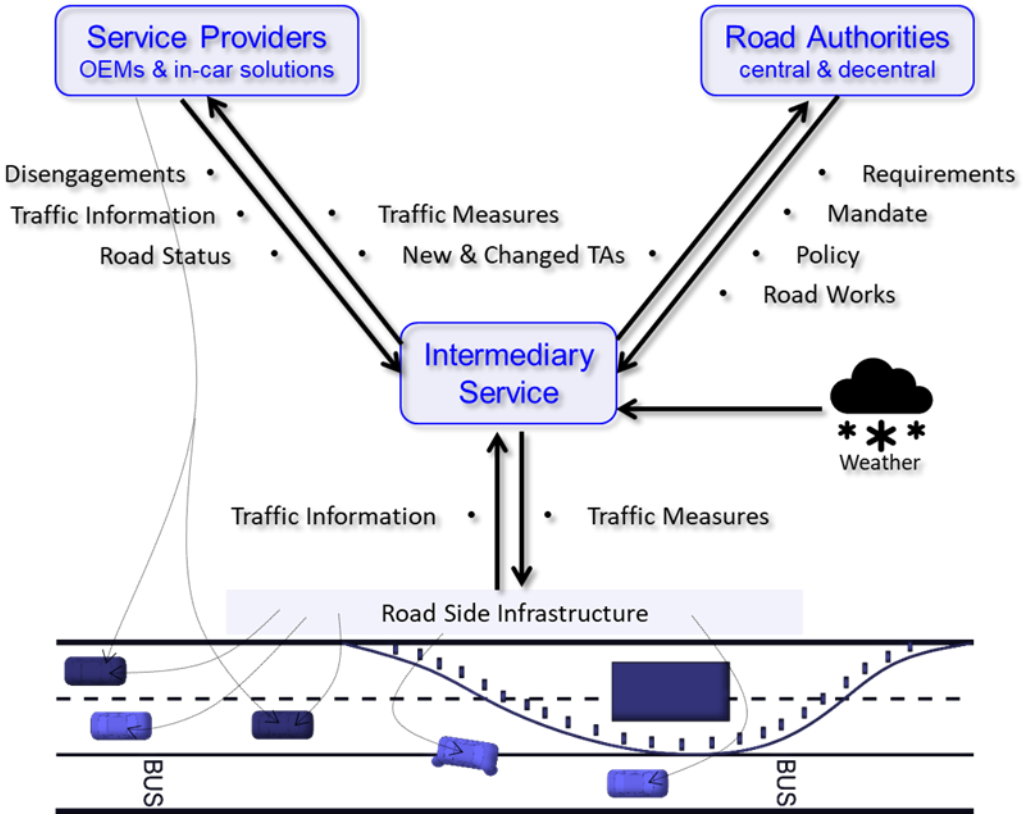

Figure 5: Schematic overview of TransAID's intermediary service approach.

In order to better understand the background of TransAID's intermediary service approach, we pose some guiding questions which allow us to frame some of the decisions taken in the subsequent sections.

- **From a technical perspective:**
  - How would TransAID get access to infrastructure/data?
  - Would TransAID offer its own hardware solution as a service (cf. business model)?

- **From a business-oriented perspective:**
  - Is TransAID actually a physical layer with both hardware and software as an operating, private traffic management organisation/company, or is it / can it be like a plug-in service in terms of a software (perhaps with added hardware) layer that can be bought and integrated by a traffic management centre?
  - Who would be the actual operator of such a service?
  - What would be the business model of such an existing service provider?
    - Providing data to OEMs which would pay for these?
    - Would OEMs pay the TransAID service for managing TAs that the OEMs defined in respect to their defined ODDs?

The next two sections will explore some of the options available to use for answering these questions, with Section 4.2 dealing with the more technical perspective, and section 4.3 dealing with the more business-oriented perspective.

4.2 Detecting transition areas

After the simulations of the first and second iterations, as elaborated upon in Deliverable D4.2, we know what can be expected from each specific traffic management measure. Also, D4.2 describes the recipe for each traffic management measure under which traffic conditions it is most applicable. However, there, the presence and location of a TA is always known, and which traffic measures to apply to the TA is predetermined.

For TransAID to become part of a complete traffic management system, we need to be able to detect TAs, select (and possibly combine) services, and detect when they are most appropriately timed for deployment. This means that TransAID measures should be deployed on different areas of the network for different TAs and/or possibly multiple measures to the same area.

Note that TransAID has not extensively evaluated the combination of services in the same area and so the effects of this are unknown. They might support each other (e.g., distribute ToCs and manage MRMs by guidance to a safe spot), but they could also adversely impact each other. The same can be said for measures that are deployed relatively close to each other where the effects of one measure influence that of another.

TransAID foresees several possibilities to detect TAs and their characteristics. The schema below gives an overview:

1. Via infrastructure:
    a) Via traditional road sensors (e.g., cameras, loop-detector data (LDD), floating-car data (FCD), …)
    b) Via digital infrastructure (ITS-G5, 3G/4G/5G)
2. Via OEMs (possibly with some ToC cause identification)
3. By comparing ISAD to the ODD (see further on for a detailed explanation)

One way is to use the infrastructure. Slowdowns of vehicles detected through LDD and/or FCD could point to ToCs. Also, cameras in conjunction with advanced video processing techniques could be used to detect the detailed behaviour of cars, thereby identifying behaviours that relate to a ToC situation (1a). In a more direct way, the digital infrastructure can receive messages about ToCs. This can be done for example via MCM messages or extended CAM messages which give a direct indication of ToCs (1b), if the currently available pre-standard in combination with the TransAID additions will find its way to be an often used or even mandatory part of CAV communications.

Another possibility is to collect ToC information from OEMs. Their backend can receive feedback from automated vehicles performing a ToC through their proprietary connection, collect these, and provide them for the purpose of identifying TAs. Since OEMs have knowledge of a car's functions, they might also be able to indicate why a ToC was needed. The cause for the ToC can help to select the correct TransAID measure in the short-term, and in the long-term point to a possible solution (e.g., changing infrastructure, changing road works layout, …). On the other hand, this would require an (unlikely) openness of the OEMs or a legal framework which requires OEMs to make such information available.

Finally, one of TransAID's Horizon 2020 affiliate projects, INFRAMIX, has introduced and developed a helpful methodology called Infrastructure Support levels for Automated Driving (ISAD[8]), as shown in Figure 6. These levels would have to be defined for the road network and provides OEMs with an insight as to the 'road readiness' for automated driving. As a counterpart, the OEMs could define the Operational Design Domain (ODD) which specifies in which circumstances vehicles are able to operate automated. By matching ISAD levels to the ODDs, one could, theoretically, identify mismatches which point to TAs. Note that at the time of writing, both the ISAD concept and ODD concept have not (yet) been developed to a point that enables such a comparison. In addition, it is not yet clear if OEMs will make their ODD restrictions available to the public, or if a format is found which allows the easy exchange of ODD (and ISAD) information, possibly even online over the air. Note that such developments have recently started, e.g., the definition of ODD taxonomies in ISO 34503 or BSI PAS 1883, or the definition of open formats like OpenODD[9].

	Level	Name	Description	Digital information provided to AVs			
				Digital map with static road signs	VMS, warnings, incidents, weather	Microscopic traffic situation	Guidance: speed, gap, lane advice
Digital infrastructure	A	Cooperative driving	Based on the real-time information on vehicles movements, he infrastructure is able to guide AVs (groups of vehicles or single vehicles) in order to optimize the overall traffic flow	X	X	X	X
	B	Cooperative perception	Infrastructure is capable of perceiving microscopic traffic situations and providing this data to AVs in real-time	X	X	X	
	C	Dynamic digital information	All dynamic and static infrastructure information is available in digital form and can be provided to AVs	X	X		
Conventional infrastructure	D	Static digital information / Map support	Digital map data is available with static road signs. Map data could be complemented by physical reference points (landmarks signs). Traffic lights, short term road works and VMS need to be recognized by AVs	X			
	E	Conventional infrastructure / no AV support	Conventional infrastructure without digital information. AVs need to recognise road geometry and road signs				

Figure 6: The various levels of infrastructure support for automated driving (ISAD) as an outcome of the INFRAMIX project.

The first two parts of the schema in Figure 6 mainly deal with ToCs as opposed to TAs. Some kind of clustering algorithm would thus be needed to group the ToCs in space and time to highlight areas with a high probability of ToC occurrences. Those areas can then be selected as potential candidates for TAs.

As a next step, for each TA, an analysis of the direct environment (i.e. physical and digital infrastructure, traffic conditions, and other traffic measures) needs to be done. The result of this analysis gives the needed input to select and apply the correct TransAID measure(s).

[8] https://www.inframix.eu/infrastructure-categorization/

[9] https://code.asam.net/simulation/proposal/openodd

4.3 TransAID bridging NRAs and OEMs

In the sections above, three parties can be identified:

1. The TransAID intermediary service.
2. OEMs.
3. The road authority (RA) (or entity responsible for implementing traffic management).

All these parties will have to work in close collaboration. To detect TAs, select services, and time the deployment, TransAID proposes a third-party intermediary. That way the detection of TAs is performed in a centralised way, and OEMs and RAs have a single point of contact for providing and receiving information about TAs. Moreover, because the ToCs (or disengagements) of OEMs' fleets might say something about the performance of their automation, some level of trust is needed as that data is quite sensitive. A trusted third-party intermediary service could sign agreements with the OEMs about how they handle their data. The intermediary can, for example, aggregate and anonymise the collected data from OEMs, before sharing it with road authorities or other service providers. For TransAID, only the sharing of TAs and possibly some cause category, without any data pointing to OEMs (i.e. 'who' provided the ToCs) is needed.

In addition, the intermediary service can provide support to the OEMs by sharing the locations of identified TAs and changes to ISAD. The information about TAs can help OEMs to improve their systems, because they have specific examples of situations where the limits of their ODDs are reached. Also, that information could be used for geofences or warnings to the driver which inform about limitations for automated driving. The changes to ISAD could be essential to OEMs when that information is used to determine where automation is possible and where it is not. Providing the changes could therefore be quite important. The changes can be the result of short-term issues like malfunctions, or long-term changes due to changed infrastructure.

The intermediary can use high-level logic to identify the type of TA and (automatically) select which service is the most applicable to deploy given the current situation (road layout and traffic conditions), as explained for each use case in Deliverable D4.2. However, other traffic measures might also have been deployed by the RA which could cause interfere. A possible way for the intermediary and the RA to collaborate and prevent the interference is to exchange the traffic measures. The intermediary service receives those deployed by the RA and takes them into consideration when setting the variables (i.e. geographic demarcation and timing) for the chosen TransAID measure. Then, the intermediary sends a service request to the RA (i.e. TMC) which can put the request in the context of the common operational picture (COP). Based on the COP, the RA can accept, deny, or postpone the request. A protocol for exchanging traffic measures in this way already exists in the Netherlands and is called DVM-Exchange[10].

[10] http://www.dvm-exchange.nl/ and https://trid.trb.org/view/1264837

Finally, considering the detection of TAs we foresee two different time scales:

1. A slower changing situation, corresponding to a more strategic level that leads to static transition areas, such as changes to infrastructure or roadside systems.

2. A faster changing situation, leading to dynamic transition areas (because of changing weather conditions such as fog and heavy snow, incidents, or other specific traffic conditions), requiring timely action.

For the first type of time scale, it is important to have a close and clearly understood collaboration between the intermediary and any applicable road authorities and OEMs, in order to convey the right information at the right time (note that suitable harmonisation of C-ITS services is a necessary ingredient here). In Belgium and the Netherlands for example, all road works are supposed to be entered (well in time) in a central database[11,12], with yet another database that contains all the road signs. Changes in infrastructure and specifically ISAD would need to be communicated. Considering the OEM's perspective, the (slower) changing ODD information resulting from the evolution of AD systems (i.e. new models and regular software updates) would be necessary for the intermediary to identify TAs when matching the ODD to ISAD.

The second, faster reacting time scale, also requires some organisational work. There, it would be very beneficial to have direct interaction with the OEMs' backends, so that the intermediary is able to detect for example the root causes of disengagements, thereby quickly identifying temporary TAs. A close cooperation will need to be put in place here, whereby, e.g., the OEMs mandatorily report ToCs/disengagements to the intermediary service provider, thereby allowing TransAID to, among other benefits, increase traffic safety.

[11] http://www.geopunt.be/

[12] https://www.rijkswaterstaat.nl/zakelijk/verkeersmanagement/wegverkeer/spin.aspx

5 Conclusions

This deliverable D4.3 explained how simulations of both the baseline (WP3) and the traffic management schemes (WP4) can be ported from the SUMO simulation environment (with the help of the TraCI interface and Python scripts) to the iCS environment (using the C++ language).

We first gave an explanation on how to set up the creation of a traffic management application in the context of the iCS. Details were given on how to prepare the development of an application, based on the source code in the repository. We also explained the interactions between the iCS, SUMO, ns-3, and the various applications, using subscriptions and the exchange of messages.

We then provided information on how to test the implemented traffic management schemes, i.e. checking of the newly implemented logic would comply with the previously obtained results from WP4. We explained this via a method of aggregate quantities, rather than explicitly comparing time-space diagrams.

Finally, we provided information on how TransAID can be scaled up and generalised to become a third-party intermediary service provider. We approached this from both a technical and a business-oriented perspective. For TransAID to become part of a complete traffic management system, we focused on the technical side on how to detect TAs, select (and possibly combine) services, and then detect when they are most appropriately timed for deployment. To this end, detection can be done via the infrastructure (e.g., road sensors or even digital communication infrastructure), via the OEMs, or by comparing an infrastructure's newly-defined ISAD level (Infrastructure Support levels for Automated Driving) to the operational design domain (ODD) of the vehicle. From a business-oriented perspective, TransAID requires collaboration with both OEMs and (national) road authority (NRAs). Considering the previous technical challenges (detecting TAs, selecting services, and timing their deployment), TransAID bridges all these parties in such a way that the detection of TAs is performed in a centralised way, and OEMs and RAs have a single point of contact for providing and receiving information about TAs. The information and data flow in this construction is multidirectional, as for example the intermediary service can also provide support to the OEMs by sharing the locations of identified TAs and changes to ISAD. This can then in turn help the OEMs to improve their systems, because they now have specific examples of situations where the limits of their ODDs are reached. And finally, considering the detection of TAs we foresee two different time scales at which TransAID's traffic management service will operate, i.e. a slower changing situation (corresponding to a more strategic level that leads to static transition areas, such as changes to infrastructure or roadside systems), and a faster changing situation (leading to dynamic transition areas (because of changing weather conditions such as fog and heavy snow, incidents, or other specific traffic conditions), requiring timely action).