This Project is co-funded by the European Union under the 7th Framework Programme

Deliverable D6.2

Design Criteria for the User Interface and Policy Assessment Reports (Final Version)

Project Information

Project number	321624
Project acronym	HIGH-TOOL
Project full title	Strategic high-level transport model
Start date of project	15 April 2013
Duration	40 months
Partners	KIT, FŐMTERV, MCRIT, MKmetric, Panteia, Significance, TNO, TML

Document Information

Work package	WP6: User Interface and Policy Assessment Reports
Deliverable Title	D6.2: Design Criteria for the User Interface and Policy Assessment Reports (Final Version)
Version	1.2
Date of submission	20 May 2016
Main Editor(s)	Oriol Biosca, Efrain Larrea, Andreu Ulied (MCRIT)
Contributor(s)	Eckhard Szimba (KIT), Benedikt Mandel (MKmetric)
Reviewer(s)	KIT, Panteia, MKmetric
This document should be referenced as	Biosca, O., Larrea, E., Ulied, A. (2016): Design Criteria for the User Interface and Policy Assessment Reports (Final Version), HIGH-TOOL Deliverable D6.2, co-funded by the European Commission under the 7th Framework Programme, Karlsruhe, Germany.

Classification – This report is:									
Draft		Final	X	Confidential		Restricted		Public	X

History			
Version	**Issue Date**	**Status**	**Distribution**
0.1	4 March 2016	1st draft	Consortium
0.2	10 March 2016	2nd draft	Peer Review
0.3	14 March 2016	3rd draft	Peer Review
1.0	14 April 2016	Final draft	Consortium, Project Officer
1.1	18 April 2016	Final	Consortium, Project Officer
1.2	20 May 2016	Final revised	Consortium, Project Officer

Table of Contents

Executive Summary 13

1 Objective 14

2 Design Criteria for the HIGH-TOOL User Interface 15

2.1 Conceptual Approach 15

2.2 Design Principles for User Interaction 16

2.3 Software Alternative Solutions 17

2.4 Technological Solution Adopted for the User Interface of the HIGH-TOOL Model 19

2.4.1 Solution Adopted for the Client Application 21

2.4.2 Solution Adopted for the Server Application 22

2.4.3 Multi-Platform Compatibility 22

2.4.4 Stand-Alone Version 23

2.4.5 Hardware Requirements 24

2.4.6 Security Solutions Considered by HIGH-TOOL Software 24

2.4.7 User Profiles and Access Restrictions 25

3 Description of the User Interface of the HIGH-TOOL Model 26

3.1 Overview 26

3.2 Framework Conditions 28

3.3 Defining Transport Policy Measures 30

3.3.1 Defining Single Transport Policy Measures at Aggregated EU Level 30

3.3.2 Defining Combined Transport Policy Measures at Aggregated EU Level 33

3.3.3 Defining Customised Transport Policy Measures at Aggregated EU Level 34

3.3.4 Definition of Time Evolutions 2010–2050 for Input Variables 36

3.3.5 Definition of Input Variables at Country and Regional Level 39

3.3.6 Policy Package Management 43

3.4 Expert Mode 43

3.4.1 Edit Tables 46

3.4.2 Edit Hyper-Network 46

3.5 Run Menu 47

4 Policy Assessment Report 49
4.1 Policy Assessment Report in MS Excel 49
4.2 Main Output Indicators 52
4.3 Export Table Contents 53
5 Stand-Alone Version 55
6 Future Improvements of HIGH-TOOL 56
7 References 57

Index of Figures

Figure 1: Architecture of the User Interface of the HIGH-TOOL model and its interaction with the database and HIGH-TOOL components 20

Figure 2: The User Interface of the HIGH-TOOL model visualised on a mobile phone 23

Figure 3: Login screen 25

Figure 4: Logical structure of HIGH-TOOL model and User Interface 27

Figure 5: Showing the set of Framework Conditions 29

Figure 6: Editing a Transport Policy Measure 31

Figure 7: Selecting a combination of TPMs on the interface 33

Figure 8: The policy levers for the combined TPM measure 33

Figure 9: Interface for building a customised TPM 35

Figure 10: Default time profile of a policy lever 36

Figure 11: Accessing the numerical and graphic editors 36

Figure 12: Interface for the numerical trajectory editor 37

Figure 13: Interface for the graphical trajectory editor 38

Figure 14: Accessing the map editor 39

Figure 15: Geographic distribution of a parameter (homogeneous) 40

Figure 16: Geographic distribution of a parameter (proportional to population density of NUTS-2 regions) 41

Figure 17: Manual modification of a value for a specific region 42

Figure 18: Graphic time editor for country/region 43

Figure 19: Workflow of the Interface 44

Figure 20: Enabling the expert mode 45

Figure 21: Expert mode interface 45

Figure 22: Editing the road Hyper-Network 47

Figure 23: View and manage your model runs 48

Figure 24: Main menu of the Policy Assessment Report in MS Excel 50

Figure 25: Full results displayed in tables, on a per theme basis (I) 50

Figure 26: Full results displayed in charts, on a per theme basis (II) 51

Figure 27: Full results displayed in charts, on a per theme basis (III) 52

Figure 28: Synthesis indicators displayed on the User Interface of the HIGH-TOOL model 53

Figure 29: Export table utility in the interface of the HIGH-TOOL model 54

Figure 30: Stand-alone HIGH-TOOL interface 55

Index of Tables

Table 1: World regions by intercontinental bundles 29

Table 2: Policy dimensions and TPMs considered 31

Table 3: Available options in the HIGH-TOOL model variable editor 32

Table 4: Available policy levers by module on the customised TPM interface 35

Table 5: Synthesis indicators of a HIGH-TOOL simulation 53

Glossary

Angular.js	A client framework to build web applications in JavaScript
Apache	Web server technology
API	Application Programming Interface: A set of routines, protocols, and tools for building software applications
Bootstrap	Mobile first responsive web framework
Bower	A package management system for client-side programming
C	General-purpose imperative computer programming language
C++	General-purpose programming language. It has imperative, object-oriented and generic programming features
C#	Multi-paradigm programming language. Developed by Microsoft within its .NET initiative and later approved as a standard by ECMA (ECMA-334) and ISO (ISO/IEC 23270:2006)
CO_2	Carbon Dioxide
CSV	Comma-Separated Values
D3.js	JavaScript library used to manipulate documents based on data. It produces HTML and SVG outputs from raw data
DOM	Document Object Model. Cross-platform and language-independent convention for representing and interacting with objects in XML documents
EC	European Commission
ECMA	European Computer Manufacturers Association
ECMAScript	Scripting language standardised by ECMA International in the ECMA-262 specification and ISO/IEC 16262
Electron	An open source software platform maintained by GitHub. Is a modified version of Node.js which can launch and control WebKit instances
EIB	European Investment Bank
EU	European Union
EUROPOP2010	Eurostat Population Projections 2010-based
GDP	Gross Domestic Product
GHG	Greenhouse Gas
GIS	Geographic information system
Grunt	JavaScript task runner
GTK	Cross-platform widget toolkit for creating graphical user interfaces
HTML	Hyper Text Markup Language. The standard markup language used to create web pages
HTML5	Fifth version of the HTML Language
HTTP	Hyper Text Transport Protocol. An application protocol for distributed, collaborative, hypermedia information system
HTTPS	Hyper Text Transport Protocol Secure. A communications protocol for secure communication over a computer network
IO	Input-Output
IOT	Input-Output Table
Java	Object oriented programming language
JavaScript	A dialect of ECMAScript programming language

JSON	JavaScript Object Notation. Open standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs. It is used primarily to transmit data between a server and web application, as an alternative to XML
JVM	Java Virtual Machine. An abstract computer used to translate Java programs into actual machine code and execute them
Linux	The kernel of the GNU/Linux operating system. It is usual to refer GNU/Linux just as Linux
MS	Microsoft
MVC	Model-View-Controller software architecture pattern
Node.js	A platform built on Chrome's JavaScript runtime for building network applications
NO_x	Nitrogen Oxides
NPM	A JavaScript package manager used in Node.js
NUI	Natural User Interface. A User interface design paradigm
NUTS	Nomenclature of Territorial Units for Statistics. A number assigned to it defines the level of granularity, such as 0 for the country level
OECD	Organisation for Economic Co-operation and Development
OGC	Open Geospatial Consortium
ORM	Object-Relational Mapping, a programming technique to convert data between incompatible type systems
OS	Operating System
Pagination bar	Pagination is the process of dividing (content) into discrete pages, either electronic pages or printed pages. A pagination bar is used in User Interface designs as horizontal bar allowing to select different option pages
Pagination tab	Pagination is the process of dividing (content) into discrete pages, either electronic pages or printed pages. Pagination tabs are used in User Interface designs as a set of horizontal tabs allowing to select different option pages
pkm	Passenger-kilometre
Porting	In software engineering, porting is the process of adapting software so that an executable program can be created for a computing environment that is different from the one for which it was originally designed (e.g. different CPU, operating system, or third party library). The term is also used when software/hardware is changed to make them usable in different environments
PostGIS	An open source software program that adds support for geographic objects to the PostgreSQL object-relational database
PostgreSQL	An object-relational database
Python	General-purpose, high-level programming language
Qt	Cross-platform application framework
RES	Renewable Energy Sources
RESTFul API	Representational State Transfer (REST) API. A software architecture style consisting of guidelines and best practices for creating scalable web services
RSS	Rich Site Summary. Family of standard web feed formats to publish frequently updated information: blog entries, news headlines, audio, video
SAB	Scientific Advisory Board
Sails.js	A server framework to build Node.js applications
SDK	Software Development Kit, a set of software development tools
SQL	Structured Query Language
SVG	Scalable Vector Graphics. A markup language used to create two-dimensional graphics with support for interactivity and animation

Swing	Graphical user interfaces widget toolkit for Java
Three.js	A cross-browser JavaScript library/API used to create and display animated 3D computer graphics on a Web browser
tkm	Tonne-kilometre
TPM	Transport Policy Measure
TRANS-TOOLS	Network-based European Transport Model
VAT	Value Added Tax
vkm	Vehicle-kilometre
WebSocket	Protocol providing full-duplex communication channels over a single connection
WebKit	An open source browser engine. It powers Google's Chrome web browser
WP	Work Package
XHR	XMLHttpRequest, an API used to send HTTP or HTTPS requests to a web server and load the server response data back into the script

Executive Summary

The HIGH-TOOL model is being designed as a user-friendly policy-assessment tool, which can be operated by transport experts without a deep modelling background.

The User Interface of the HIGH-TOOL model is therefore designed as a "user-centric" application, empowering users to make a productive and intelligent use of the HIGH-TOOL model. The User Interface will help users think more systematically and build consistent policy questions. On the other hand, the HIGH-TOOL User Interface is designed to reduce the risk of abusing models beyond their limits.

Design principles applied in relation to user interaction are based on Tognazzini (2013): Anticipation, Visible Navigation, Efficiency, Consistency, Explorable Interfaces, Learnability, and Readability, Intuitiveness and Self Explaining.

The User Interface is structured into four components that build two logic blocks. The first logic block is used to define scenarios (Simulation Packages) integrating a Transport Policy Measure and a set of socio-economic Framework Conditions (by default the EU Reference Scenario). The second logic block concerns the generation of Policy Assessment Reports, consisting of synthesis indicators and a full set of detailed results.

The User Interface is programmed as a stand-alone online application based on free and open source software. The full stack is programmed in JavaScript (ECMAScript v6).

Since October 2015, a restricted access working User Interface is available online to facilitate design discussions and testing of robustness and user-friendliness. It can be accessed at http://www.high-tool.eu/model.

This online version has been used to facilitate a clear visualisation and validation of the design criteria, as well as to demonstrate the usefulness of software technologies selected for development.

The design criteria were inspired by the survey results and discussions of the user workshops, as well as by the evaluation of the prototype version of the HIGH-TOOL model by the project's Scientific Advisory Board. Further interaction with users is expected to validate and adjust the design criteria introduced in this report. Also results from robustness and user-friendliness testing will provide extra adjustments on the final version of the tool.

At this stage all the particular options displayed by the User Interface and all images displayed in screenshots are not completely final and some adjustments will occur until the delivery date of the “final” version of the HIGH-TOOL model.

1 Objective

The objective of Work Package (WP) 6 is the development of an intuitive user-friendly interface. It is based on the findings of WP1 "User Requirements", WP3 "Basic Input Data", WP4 "Elasticities and Equations", and WP5 "Development of the HIGH-TOOL model". The approach follows the structure of the HIGH-TOOL model outlined in D2.2 (Mandel et al., 2016) and the user requirements outlined in D1.1 (Vanherle et al., 2013) and D1.2 (Purwanto, Szimba and Mandel, 2015).

This report presents the design criteria of the User Interface and the Policy Assessment Reports, corresponding with the outputs of the tasks 6.1 and 6.2, as well as sub-task 6.3.2.

The approach to the User Interface is based on the user requirements defined by D1.1, and in addition takes into account following feedback:

- 1st User Workshop and Online Survey (June 2013)
- 2nd User Workshop (June 2014)
- User Interface advice by Mateu Turró, HIGH-TOOL Scientific Advisory Board member, former associate director at EIB (January 2015)
- User Interface advice by Thomas Spiegel, HIGH-TOOL Scientific Advisory Board member, Austrian Ministry of Transport (February 2015)
- 3rd User Workshop (December 2015).

The design criteria have already been implemented in an advanced version of the User Interface of the HIGH-TOOL model, which is available online (http://www.high-tool.eu/model). This working version has been used to facilitate a clear visualisation and validation of the design criteria, and to demonstrate the usefulness of the software technologies selected for its implementation.

This report is structured as follows:

- **Chapter 2** presents the Design Criteria adopted for the development of the User Interface, as well as the technical solutions adopted.
- **Chapter 3** is organised as a user guide, presenting how in practice, users will define assumptions and how outputs will be presented. The main editors currently available in the User Interface are presented and their organisation is shown. Out of user perspective the steps necessary to define scenarios respectively to make inputs in the model are listed.
- **Chapter 4** presents the Policy Assessment Reports.
- **Chapter 5** introduces the stand-alone version of HIGH-TOOL.
- **Chapter 6** lists some possible improvements for future developments of HIGH-TOOL.

2 Design Criteria for the HIGH-TOOL User Interface

This chapter presents the design criteria adopted for the development of the User Interface of the HIGH-TOOL model, namely its conceptual approach and the core design principles taken into consideration. It also discusses the technological solutions adopted and their implications in relation to software and hardware requirements.

2.1 Conceptual Approach

Conceptually the User Interface of the HIGH-TOOL model is "user-centric" in contrast to other "model-centric" approaches.

The User Interface is conceptualised to support users in translating policy-questions into the specific questions suited for HIGH-TOOL. Therefore, the User Interface supports users in translating their questions into model inputs, which can then be run in a policy scenario by the HIGH-TOOL model. Because there is no perfect match between the complexity of policy questions and the unavoidable simplicity of the model's input variables, the User Interface needs to match human and model needs in an intelligent way.

Therefore, the main purpose of the User Interface is to facilitate the definition of consistent scenarios, in terms of policy questions and Framework Conditions. This is achieved by introducing consistency checks, reference values, thresholds and warnings in the User Interface. The User Interface guides users in avoiding errors and assists them systematically in order to generate consistent policy scenarios that models can properly run.

On the other hand, the interface is conceived to avoid abuse of the model and its results. Modelling outputs are presented in synthetic Policy Assessment Reports, designed with harmonised criteria in a standard format to be used as reference. However, users are free to download output datasets to execute their own analyses and develop their own reports and presentations.

The ultimate design criterion for the User Interface is optimising the time that users have to devote to interacting with the model. This is achieved by saving time from tedious and mechanical processes, as well as by facilitating a clear visualisation of the most relevant inputs and outputs, for instance by providing graphic and cartographic displays.

As a result, the HIGH-TOOL User Interface was designed to become a "user-centric" interface that is as policy-oriented as possible and as scientific as necessary. It preserves transparency when defining consistent scenarios and provides intelligence and assistance to users that ask legitimate questions the HIGH-TOOL model can investigate.

2.2 Design Principles for User Interaction

The design principles adopted for the User Interface are based on the criteria outlined by Tognazzini (2013). They include **Anticipation**, **Visible Navigation**, **Efficiency**, **Consistency**, **Explorable Interfaces**, **Learnability**, **Readability**, **Intuitiveness** and **Self Explaining**.

The criteria have been addressed by the User Interface of the HIGH-TOOL model as follows:

Anticipation: The User Interface is designed to anticipate user needs. Users are not expected to search for information or evoke necessary tools, but the information and tools needed for each step of the process are provided to the users in a natural way. Default values provided by the HIGH-TOOL model are 'intelligent'.

Visible Navigation: Navigation is reduced to a minimum. The User Interface is designed such that users can access configuration, hypotheses, calculations and result screens by a minimum number of clicks and actions. Navigation is clear and natural, allowing a user to quickly transition from novice to expert.

Efficiency: The User Interface facilitates user's productivity, not computer's productivity. The application is designed such that task flows are comfortable, clear and optimal for the user, rather than organised according to programming needs.

Consistency with user expectations:

- Logical procedures stay coherent along the application
- Buttons, input fields and navigational controls are always located in the same position of the interface
- Harmonious 'overall look' of the different screens.

Explorable Interfaces: Users of the HIGH-TOOL User Interface are offered a line of least resistance allowing them to do just what they want, getting the job done in the quickest way possible, while still supporting those who want to explore further. This means stable visual elements to allow fast navigation, making actions reversible and always allowing a way out while making it easier to stay in.

Learnability: The User Interface allows for a quick learning curve that supports the performance of tasks within the minimum time possible (Natural User Interface, NUI).

Readability: Texts displayed in the User Interface are incorporated in such a way that they can be read properly without effort, favouring dark text on pale backgrounds, avoiding grey backgrounds and using font sizes that are large enough to be readable on standard monitors.

Intuitive and Self Explaining: The layout of the HIGH-TOOL model and the logic of processes in the User Interface are conceived in a way that makes it possible for users to run the tool on an intuitive basis. Contextual information provided by the User Interface, information made available through icons and tool-tip displays support the self-explaining property of the HIGH-TOOL model.

All design criteria together empower users to work efficiently with the HIGH-TOOL model while enjoying all features.

2.3 Software Alternative Solutions

The market trend converges towards web technologies to support User Interfaces because:

- Applications are cross-platform, provided that developers restrict themselves to evergreen browsers and open standards.
- Easiness to update and distribute software.
- They can eventually be transformed into stand-alone applications if needed (using solutions like wrappers, web views).

However, there are also some pitfalls to overcome:

- Performance standards of web interfaces are usually worse than those developed using other technologies like Java, C/C++, C#, and UI/utility libraries (Qt, GTK, Swift).
- Web browsers are sandboxed due to security concerns (running programs from unverified third parties is restricted). Therefore web applications can only use web APIs to operate with the browser, which in some cases may not be sufficient for all needed functionalities.
- Browsers are single-threaded. Multi-threading is limited to web workers with a simple messaging system to coordinate threads.

To deal with the second pitfall, most applications use tiers to divide responsibility. The most common solution used by web applications is the three tier architecture, with a presentation tier (that renders the User Interface), an application tier (that handles the business logic), and a storage tier (including data persistence mechanisms and a data access layer that encapsulates persistence mechanisms and exposes the data). This solution is also adopted in the HIGH-TOOL model.

The common practice is to use frameworks to shorten the development cycle. Frameworks are pieces of software that abstract common features from the low level implementation. The following frameworks have been considered when analysing the best software solution for the HIGH-TOOL model (see Table 1, Table 2 and Table 3). After evaluating their advantages and disadvantages, it has finally been decided to use AngularJS and SailsJS frameworks (see chapters 2.4.1 and 2.4.2).

Table 1: Frameworks considered for the development of the User Interface: client applications

Client Framework	Open Source	Programming Language	Main Features	Potential Disadvantages
Web2Py	Yes	Python JavaScript	- Web framework - Easy to run, requires no installation - Commitment to backward compatibility - Multiple database drivers - Speaks multiple protocols - Prevents most common types of vulnerabilities (Cross Site Scripting, Injection Flaws, Malicious File Execution) - Includes a web-based integrated development environment - Allows mapping via standard OGC services (Geoserver, Mapserver, …)	- Requires implementation of Python 2.5, 2.6 or 2.7 (not ported to Python 3) - Less popular than other frameworks (less resources and assistance available) - Monolithic framework
Django	Yes	Python JavaScript	- Web framework - Rich featured - Large community using it, increases completeness of resources - Allows mapping via standard OGC services (Geoserver, Mapserver, …)	- Monolithic framework
AngularJS	Yes	JavaScript	- Web framework - Client side web framework - Eases development of HTML5 web applications - Promotes good practices like modularity, reuse of components and unit testing - Decouples the client side of the application from the server side - Decouples DOM manipulation from application logic - Dependency injection - Allows mapping via standard OGC services (Geoserver, Mapserver, …)	- Cannot be pre-rendered in the server to improve performance
OpenLayers	Yes	JavaScript	- WebGIS framework - Large community using it, increases completeness of resources - Allows mapping via standard OGC services (Geoserver, Mapserver, …)	- Need 3rd party sources to render data or install an OpenStreetMaps tile server - Library, not a full framework

Table 2: Frameworks considered for the development of the User Interface: client/server applications

Server/Client Framework	Open Source	Programming Language	Main Features	Potential Disadvantages
Geomedia Webmap & Smart-client	No	.NET	- GIS platform - Good documented APIs and SDKs	- Subscription-based and proprietary
ArcGIS	No	Core in C++ and OpenGL. APIs and SDK in multiple languages	- GIS platform - Good documented APIs and SDKs	- Subscription-based and proprietary
Joomla	Yes	PHP JavaScript	- Web framework - CMS - Easiness in creating and editing web-based User Interfaces - Serves multiple protocols like HTML, JSON, RSS - Follow a MVC paradigm - Modify the database schema - Allows mapping via standard OGC services (Geoserver, Mapserver, …)	- Most functionality based on contents and user administration - Create modules require knowledge of the platform - Requires ad hoc API for mapping visualisation

Table 3: Frameworks considered for the development of the User Interface: server applications

Server Framework	Open Source	Programming Language	Main Features	Potential Disadvantages
SailsJS	Yes	JavaScript	- Web framework - Client side web framework - 100% JavaScript - Waterline ORM, which abstracts Models to JavaScript objects - Auto-generate REST APIs - WebSocket Support - Reusable security policies - Front-end agnostic - Automation of tasks (using Grunt) - Allows mapping via standard OGC services (Geoserver, Mapserver, …)	- Less popular than Python frameworks

2.4 Technological Solution Adopted for the User Interface of the HIGH-TOOL Model

The User Interface of the HIGH-TOOL model has been programmed as a stand-alone online application based on AngularJS and SailsJS, both free and open source software components programmed in JavaScript (ECMAScript v6). The architecture of the HIGH-TOOL User Interface is defined by:

- A web client application developed in AngularJS to run in web browsers, complemented with other libraries used to handle screen responsiveness and GIS visualisation.

- A Node.js server application developed in SailsJS. It exposes a REST API, handles authentication and can modify the models either directly or via HIGH-TOOL modules.
- PostgreSQL + PostGIS database. It hosts the HIGH-TOOL database and extra schemata to save user data and to give persistence to the results of model runs.
- A Java Virtual Machine (JVM) capable of running Java v8 code. The SailsJS server application calls it to execute HIGH-TOOL modules, which then modify the database. Via call backs, the SailsJS server application controls execution order and replication of the results.

Figure 1 illustrates the architecture of the User Interface of the HIGH-TOOL model and its interaction with the HIGH-TOOL components and the database.

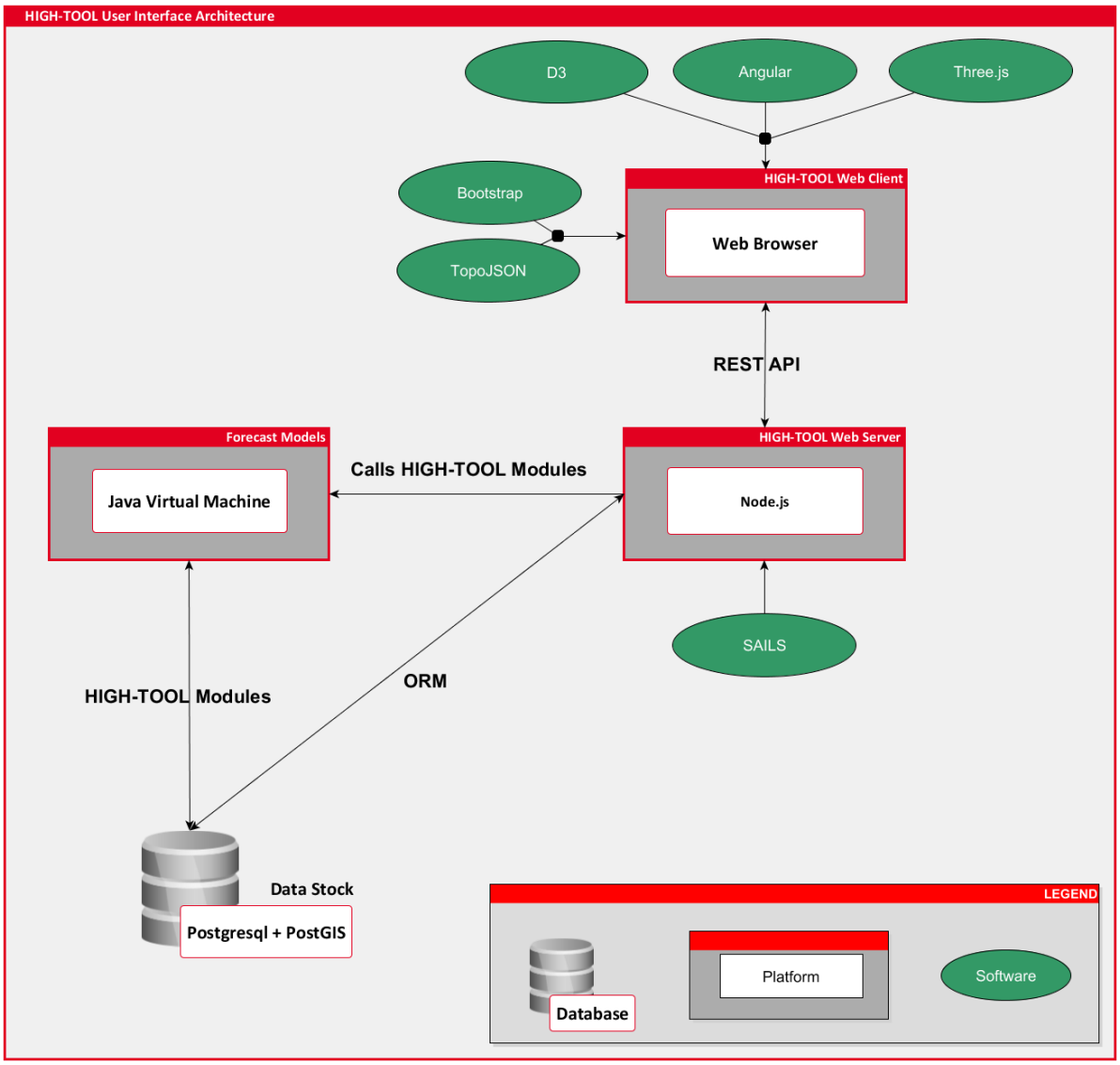


Figure 1: Architecture of the User Interface of the HIGH-TOOL model and its interaction with the database and HIGH-TOOL components

2.4.1 Solution Adopted for the Client Application

AngularJS has been selected for the client application because it eases the development of HTML5 web applications and promotes good practices like modularity, reuse of components and unit testing.

The main characteristics of AngularJS are:

- Based on plain 100% JavaScript objects, making code easier to test, maintain and reuse.
- Eliminates DOM manipulation with data-binding protocols, updating the screen view each time the model changes, and updating the model each time the view changes.
- Client-side form validation protocols allow declaring validation rules without having to write code.
- Built-in services are provided on top of XHR using third party libraries, allowing simplification of code by handling asynchronous return of data.
- Uses directives to create reusable components and hiding complex DOM structure, CSS, and behaviour. This allows focusing separately on what the application does or how the application looks.
- Works well embedded in other technologies, whereas many frameworks require full commitment.
- Is designed to be fully testable, encouraging behaviour-view separation. Comes pre-bundled with mocks and takes full advantage of dependency injection. It also comes with end-to-end scenario runner, which eliminates test flakiness.

The AngularJS framework in the User Interface is complemented with the following specialised libraries:

- Bootstrap, a mobile first responsive framework. Fastens the design of rich User Interfaces that scale from small mobile devices to full desktop screens (allows building of responsive web templates).
- D3js, a library used to manipulate documents based on data. It produces high quality HTML and SVG outputs from raw data (allows the creation of charts based on model data).
- Three.js, a lightweight cross-browser JavaScript library/API used to create and display animated 3D computer graphics on a web browser (allows implementing 3D functionalities, applied in the HIGH-TOOL model's GIS application).

2.4.2 Solution Adopted for the Server Application

Using SailsJS for the server application allows the development of event driven, non-blocking IO applications. SailsJS is a Node.js web framework built on Google's Chrome V JavaScript engine. The recommended Node.js version is 0.12.2.

Some characteristics of SailsJS are provided next:

- Based on plain 100% JavaScript objects, making code easier to test, maintain and reuse.
- Provides a simple data access layer that works with most databases.
- Built on the familiar relational model, aimed at making data modelling more practical.
- Comes with blueprints that help jumpstart an application server without writing any code thanks to auto-generated REST APIs.
- Easy WebSocket support translating incoming socket messages compatible with every route.
- Reusable security policies with role-based access control by default.
- Designed to be compatible with any front-end strategy (e.g. Angular, Backbone, iOS/ObjC, Android/Java, Windows Phone).
- Completely customisable asset workflow as it ships with Grunt. Supports already developed Grunt modules.
- Solid foundation thanks to being built on Node.js. Uses Express for handling HTTP requests, and wraps socket.io for managing WebSocket.

Both client and server components use Grunt, NPM and Bower to manage dependencies, environments (development, production, testing) and deployment. PostgreSQL and a JVM capable to run Java v1.8 are required to allow model and database interaction.

2.4.3 Multi-Platform Compatibility

The User Interface of the HIGH-TOOL model is compatible with all devices (desktops, tablets, mobile phones) (see Figure 2). The User Interface provides responsive web templates, which consider the following design features:

- Text reorganises dynamically with varying screen widths. This allows text to be displayed completely without the need for horizontal scrolling, regardless of the device used.
- Menu entries reorganise themselves dynamically. When insufficient space is available for them to show horizontally, they fold in a vertical menu that can be visible or folded.

- Control buttons and scroll bars change their appearance based on the device being used, so that all functions remain usable even in small cell phone screens, e.g. sliders in the input editors change to +/– buttons when insufficient space to be displayed is available.

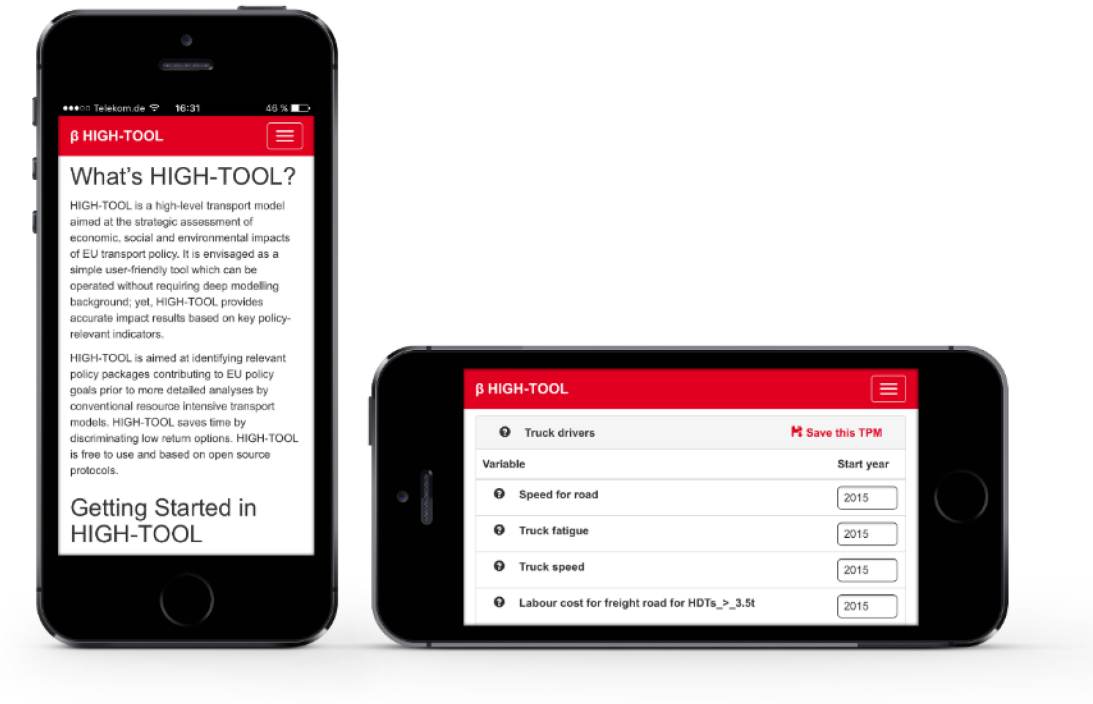


Figure 2: The User Interface of the HIGH-TOOL model visualised on a mobile phone

2.4.4 Stand-Alone Version

In addition to the client-server model, the HIGH-TOOL User Interface is bundled in a single installable file that will run on Windows 7 64-bit or newer. The stand-alone uses GitHub Electron. Electron is a software platform based in a modified version of Node.js that allows running instances of WebKit, the engine behind the Chromium web browser (the open source components of the Google Chrome web browser) and controlling them using an API that adds to the usual browser-server conventions.

The stand-alone version uses the same architecture that the client-server version and adds utilities to automate the edition of configuration files.

2.4.5 Hardware Requirements

The HIGH-TOOL model is located in a remote server with Linux OS/Apache, interacting with the client's User Interface by RESTFul API. The HIGH-TOOL database is managed by PostgreSQL – PostGIS.

Hardware requirements in HIGH-TOOL are given by the operation of the different HIGH-TOOL model components rather than by the User Interface, which is envisaged as a simple resource efficient multiplatform component.

Computational modules of the HIGH-TOOL model require that the HIGH-TOOL servers meet the following specifications:

- Intel® Xeon® processor E5-1620 (3.60 GHz Turbo, 10 MB)
- 24 GB (4 × 2 GB) 1600 MHz DDR3 ECC RDIMM memory, potentially 32 GB
- 1 TB 3.5" SSD2 drive, potentially SSD PCIe.

All the software components in the User Interface are open source and portable to the most popular operating systems. This avoids dependence on a specific technology, meaning:

- The client application will work on most environments, including Microsoft Windows, Mac OS, or Linux.
- The server software is currently based on Linux but works in other environments like Microsoft Windows or Mac OS.
- The stand-alone version can be packed for any major OS – Microsoft Windows, Mac OS, or Linux – but only a Windows 64-bit installer is provided.

Since HIGH-TOOL is a research project, the software developed does not yet have security audits and software certification. Therefore, it is recommended to install it on a separate computer and/or server environment.

2.4.6 Security Solutions Considered by HIGH-TOOL Software

Given that the HIGH-TOOL model will run on an Internet-connected server, it is important to envisage suitable security solutions to protect the application by preventing and mitigating undesired security threats. The communication between client and server is done via a RESTFul API secured with JSON-Web-Tokens. This strategy replaces session cookies allowing device agnostic communication with the server, Cross Site Request Forgery protection and improved user privacy.

In order for it to be a secure authentication and identification mechanism, JSON-Web-Tokens require HTTPS to prevent man-in-the-middle attacks, e.g. any machine between the client and the server can get the token from the HTTP header and use it to send requests to the server until the token expires.

2.4.7 User Profiles and Access Restrictions

The User Interface of the HIGH-TOOL model is by default user and password protected. Only authorised users can access the HIGH-TOOL model. The first time a user opens the interface, the web page will request an email address and a password (see Figure 3).

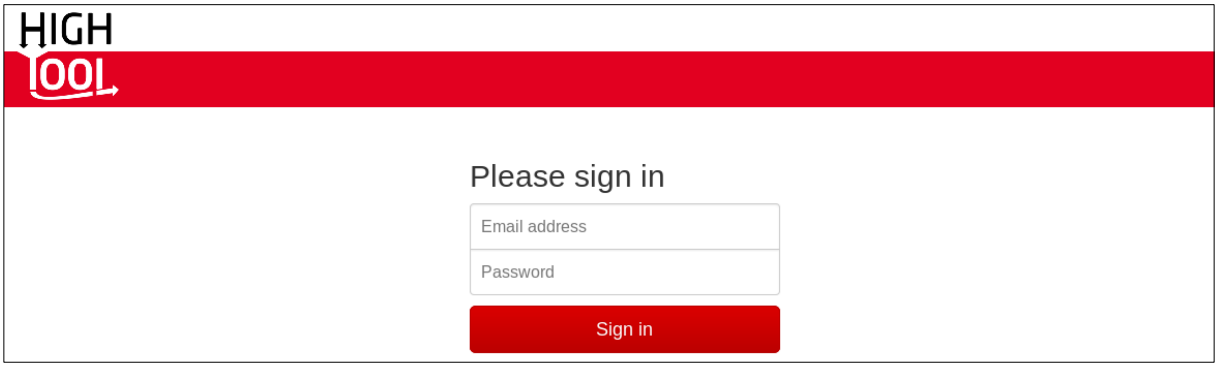


Figure 3: Login screen

User related options of the HIGH-TOOL model are displayed in the top right corner of the User Interface together with other assistance and help tools. It is envisaged that a Standard User will be able to contact the Administrator via an option button and to change its current profile. A logout option is also provided to the user in this menu.

Two different user profiles are defined in the User Interface:

- **Standard Users**: They can create new packages (specification of Transport Policy Measures, Framework Conditions, and Scenarios – see chapter 3) which can be edited, deleted and run. They may declare their packages as public so that any other Standard User can read them. A public package can be opened by other users, but cannot be deleted or modified unless it is saved as a new package (copy).
- **Administrators**: They can edit and delete any packages without taking into account their ownership. They manage the accounts of Standard Users: they can create new users and delete/ban exiting ones.

3 Description of the User Interface of the HIGH-TOOL Model

This chapter is organised as a user guide, presenting how in practice users define assumptions and how a model's output is presented. The main editors available in the User Interface are presented and their organisation is shown. Out of the user perspective, the steps necessary to define scenarios respectively to make inputs in the model are listed.

3.1 Overview

The User Interface follows a hierarchical approach classifying indicators by their relevance. The user gets access to indicators in an intelligent manner, either progressively or with all indicators together.

The User Interface reflects the inner logic of the HIGH-TOOL model. The paramount aim of the HIGH-TOOL model is to assess impacts of **Transport Policy Measures**. Transport Policy Measures are always linked to a set of socio-economic **Framework Conditions**, because their impacts may differ depending on the socio-economic context in which they are applied. At the end of a simulation, the HIGH-TOOL model returns a **Policy Assessment Report** containing all results in a standardised format. In practice, users are required to undertake the following processes to define and assess transport policies:

- Definition of a **Transport Policy Measure** (or loading an existing one) using interface tools. It contains specification of the transport policy actions, e.g. related to transport prices and taxation, transport regulation, or further integration of the single market.
- Definition of socio-economic **Framework Conditions** and/or modification of other parameters of the model (Optional step controlled in the expert mode). By default these conditions are the ones derived from the EU Reference 2013 scenario.
- Definition of a **Simulation Package** consisting of a set of a Transport Policy Measure and the optional modifications of the framework conditions.

The HIGH-TOOL model runs a given Simulation package and provides results in the form of a **Policy Assessment Report** (full details provided in chapter 4). A Policy Assessment Report is composed out of two different components as displayed in Figure 4:

- A set of Synthesis Indicators (see chapter 4.1) summarising the simulation. The set includes the Transport Policy Measure specifications made by the user, Framework Conditions selected, and results. This way, results are always linked to the specifications and conditions under which the simulation was executed.

- A complete report in MS Excel format containing all results disaggregated by territories and modes. It is generated by the HIGH-TOOL model automatically and can be downloaded from the online application. It is pre-formatted and the user might modify the graphics if needed.

The assessment report is displayed as soon as the simulation run terminates. All results, the set of Synthesis Indicators and the full report are stored automatically. In case a user starts a simulation and leaves (switching off the device) all information will be available when logging in the next time.

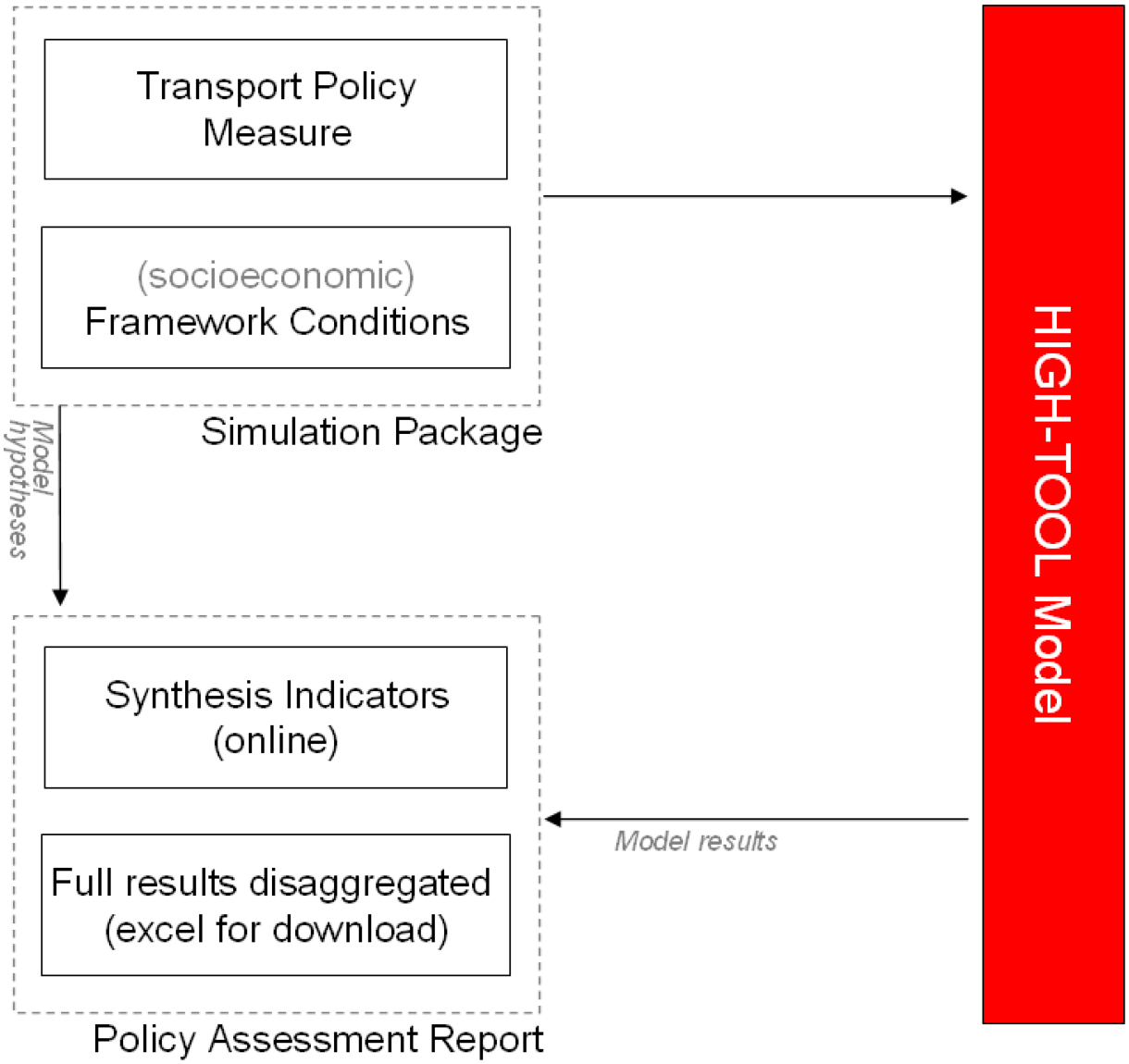


Figure 4: Logical structure of HIGH-TOOL model and User Interface

The User Interface consists of the following elements:

- **Framework Conditions**: Shows the assumptions of the model in terms of Framework Conditions (see chapter 3.2).
- **Policies**: Contains all options related to the specification and management of Transport Policy Measures (see chapter 3.3).
- **Expert mode**: Allows doing detailed modifications of the model inputs by manipulating raw tables of the data stock (see chapter 3.4).
- **Run**: All options related to running a scenario (see chapter 3.5).
- **Outputs**: Contains all options related to Policy Assessment Reports. Users are directed automatically to this area when a run finalises, are shown the synthesis indicators and are allowed to download the full Policy Assessment Report. However, users can also revisit and load Policy Assessment Reports from previous runs, export them or delete them (see chapter 4).
- **Export**: Allows exploring in detail the contents of any finalised run of the model by downloading any given table (see chapter 4.3)
- **User Profile**: Allows logging in, logging out, managing users (only for administrator profiles), displaying help materials (getting started, user manual, FAQ, downloading reference reports). This menu is separated from other menus and displayed on the right side of the screen (see chapter 2.4.7).

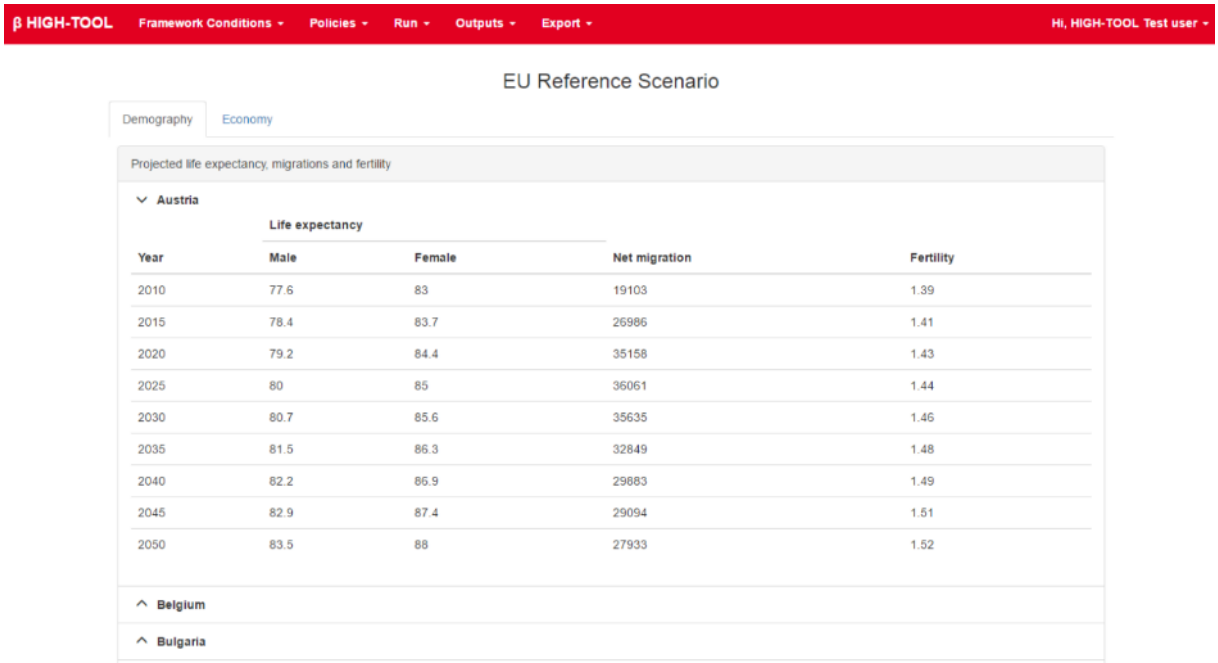
When new Transport Policy Measures are created, all variable fields are assigned default reference values, so that users only need to change parameters that will specifically be different from the reference package.

3.2 Framework Conditions

The magnitude of impacts derived from different Transport Policy Measures will be different depending on the socio-economic Framework Conditions considered. For instance, road taxation in Eastern European countries will provide different impacts on the transport system depending on the level of economic growth expected for concerned Member States.

By default, HIGH-TOOL considers the EU Reference Scenario 2013, mostly based on EUROPOP2010 (Eurostat, 2011), the Ageing Report (European Commission, 2011; 2014) and Energy Trends 2050 (European Commission, 2013b). These projections consider the years 2050 (in some cases 2060) as a temporal horizon. A summary of these Framework Conditions is shown on the interface and can be seen on Figure 5.

Users can define different sets of Framework Conditions to check how Transport Policy Measures affect differently under alternative socio-economic contexts. Framework Conditions include elements of demographics, economy and the energy market. They can be changed using the expert mode explained on chapter 3.4, but then it is responsibility of the user to ensure consistency on the hypothesis chosen.



Year	Life expectancy Male	Life expectancy Female	Net migration	Fertility
2010	77.6	83	19103	1.39
2015	78.4	83.7	26986	1.41
2020	79.2	84.4	35158	1.43
2025	80	85	36061	1.44
2030	80.7	85.6	35635	1.46
2035	81.5	86.3	32849	1.48
2040	82.2	86.9	29883	1.49
2045	82.9	87.4	29094	1.51
2050	83.5	88	27933	1.52

Figure 5: Showing the set of Framework Conditions

When editing the Framework Conditions for countries outside the EU please consider that they are aggregated into intercontinental bundles in HIGH-TOOL (see Table 1).

Table 1: World regions by intercontinental bundles

Region identifier	Region name	ETIS Country_ID	ETIS Zone1_ID	ISO Country_ID and HIGH-TOOL bundle
1170000	Iceland	117	11700	IS
4000000	Africa Nord	400	40000	AFC_NORD
5000000	Africa Central and South	500	50000	AFC_SOUTH
6000000	Africa East	600	60000	AFC_EAST
7000000	Middle East Mediterranean	700	70000	MEA_MEDITERRANEAN
8000000	Middle East East	800	80000	MEA_EAST
10000000	Commonwealth of Independent States	1000	100000	CIS
11000000	Russia, east of Ural	1100	110000	RU_EAST

Region identifier	Region name	ETIS Country_ID	ETIS Zone1_ID	ISO Country_ID and HIGH-TOOL bundle
12000000	Asia/Pacific Indian Subcontinent	1200	120000	ASP_IND
13000000	Asia/Pacific Southern Asia	1300	130000	ASP_SOUTH
14000000	Asia/Pacific Australia/Oceania	1400	140000	ASC_AUS
15000000	Asia/Pacific Far East	1500	150000	ASC_EAST
16000000	America Canada	1600	160000	CDN
17000000	America USA	1700	170000	USA
18000000	America Mexico	1800	180000	MEX
19000000	America Central	1900	190000	AMC_CENTRAL
20000000	America Caribbean	2000	200000	AMC_CARIBBEAN
21000000	America South	2100	210000	AMC_SOUTH
22000000	Antarctica	2200	220000	ANC_ANTARCTICA

3.3 Defining Transport Policy Measures

Based on the results of the User Workshops, the HIGH-TOOL model incorporates Transport Policy Measures (TPMs). The HIGH-TOOL model allows assessing different policy intensities. Maximum and minimum recommended values for all variables are provided, as well as reference baseline values.

TPMs can be defined in three ways:

- Single TPM, where the user can choose a predefined policy package
- Combined TPM, where the user can combine two or more single TPMs
- Customised TPM, where the user may choose any number of policy levers from all available ones in order to prepare a fully customised policy package.

3.3.1 Defining Single Transport Policy Measures at Aggregated EU Level

The following information and options are provided (see Figure 6):

- A pagination bar groups individual Transport Policy Measures (TPMs) by different policy dimensions. The following dimensions are considered: Internal Market, Research and Innovation, Efficiency Standards and Flanking Measures, and Pricing.
- For each policy dimension, the individual TPMs are shown in a continuous list of boxes, with the variables associated to each TPM inside the same box (see Table 2).
- Individual parameter and variable values can be specified using the options in the variable editor of each TPM.

Table 2: Policy dimensions and TPMs considered

Internal Market	Research and Innovation	Efficiency Standards and Flanking Measures	Pricing
- Opening the internal rail market - Blue Belt - Port services - IWW - Single European road market - ERA - Truck drivers - Freight corridor - Access to rail infrastructure - Rail safety - Dangerous goods - ERTMS - River info system - Acceleration of TEN-T implementation	- Road safety - Safer vehicles - ITS traffic information - ITS dynamic traffic management - ITS intelligent vehicles	- Improving local transport - Car replacement - Electric cars - H_2 cars - Maximum speeds - Enhance service quality at airports - City logistics - Promotion low emission road - New vehicles	- External costs - Circulation tax cars - CO_2 feebates - Urban road - Heavy duty charge

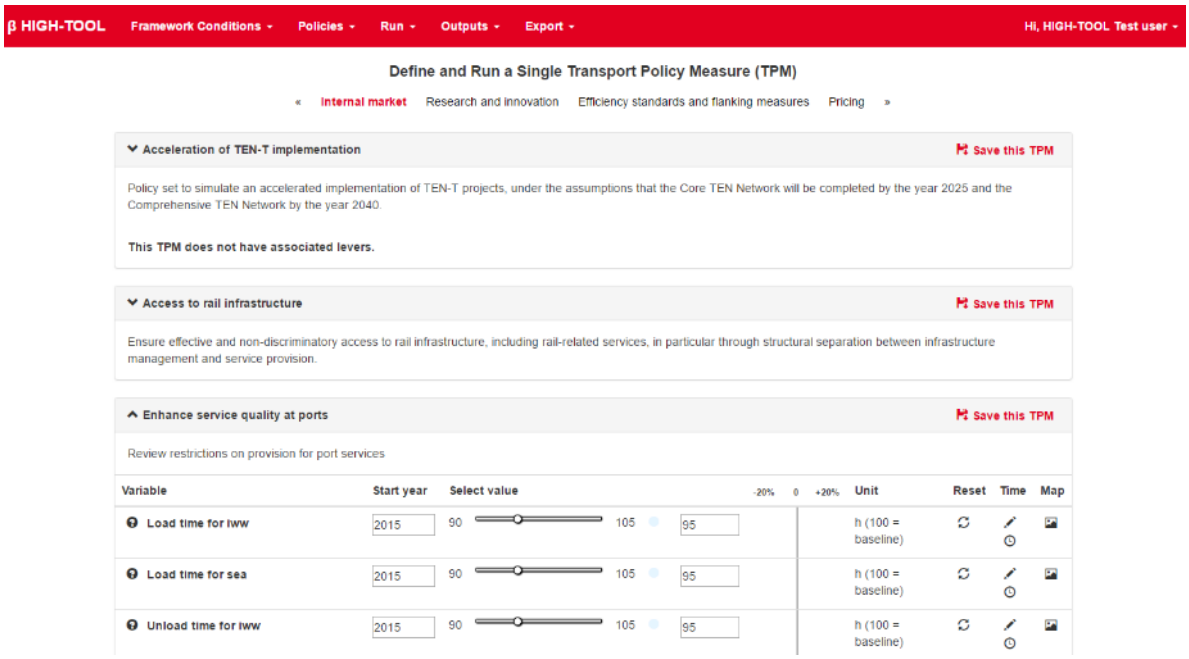


Figure 6: Editing a Transport Policy Measure

The policy levers associated with a TPM are folded by default to help navigating through the list of TPMs. By clicking on the arrow sign at the left of the TPM name the policy levers are unfolded and can be edited.

The variable editor is organised by rows, where each row corresponds to an input. The available options are, from left to right (see Table 3).

Table 3: Available options in the HIGH-TOOL model variable editor

Option	Interface Display
A help icon and the name of the input parameter or variable	Better use of truck capacity
A selector for the year that the policy comes into full effect. By default this is always set to 2015.	2015
A slider allowing the user to select a value for the input parameter in the year chosen with the previous selector. Moving the slider left or right will set out a new value. Minimum and maximum values are established for each particular input field. By default it is assumed that the selected value is applied uniformly from 2015 up to 2050.	70 100
A colour semaphore. The colour of the semaphore is a linear gradient between the predefined value and the maximum and minimum values for the input field. Dark tones indicate that the selected value for the input filed is far from the default value, whereas lighter tones indicate being close to the predefined value	
A quantitative reference value for the input field as orientation for the user. This value corresponds to "baseline" policy intensity, which could be identified in HIGH-TOOL with 2011 White Paper policy aims and actions. This value can be manually updated.	90
A visual indicator of how far the chosen value is from the reference value, measured in %. Thus a +20% indicates that the chosen value is 20% higher than the proposed reference value for that parameter.	-20% 0 +20%
The unit of the indicator.	eur_vkm (100 = baseline)
The reset button, that returns all values to its initial state for a given policy lever.	
The TIME Editor icon allows determining the evolution or trajectory of the input field in the interval 2010–2050 as a table	
The TRAJECTORY Editor icon allows determining the evolution or trajectory of the input field in the interval 2010–2050 in a graphical way.	
The MAP Editor icon allows determining the geographic distribution of the input across Europe.	

Each policy lever on the interface targets a specific variable and a set of different dimensions (e.g. "Load time for rail" affects the average loading time of freight on rail mode).

Most of the policy levers are measured in relative terms with respect to the baseline, but there are a few policy levers that are expressed in absolute terms (e.g. road tolls per country). In any case, this is indicated on the units of the lever.

A special TPM works differently from all the other: "Acceleration of TEN-T implementation". It does not have any explicit policy levers, as it internally changes the transport impedances to simulate a faster implementation of the TEN-T networks. It can be activated by ticking on the checkbox.

3.3.2 Defining Combined Transport Policy Measures at Aggregated EU Level

HIGH-TOOL allows the user building a more complex simulation of transport policies by combining different predefined TPMs. This option presents the list of available TPMs and by ticking on the checkbox of each TPM they will be added to the simulation. For each TPM added, the interface will display the policy levers related to it (see Figure 7 and Figure 8).

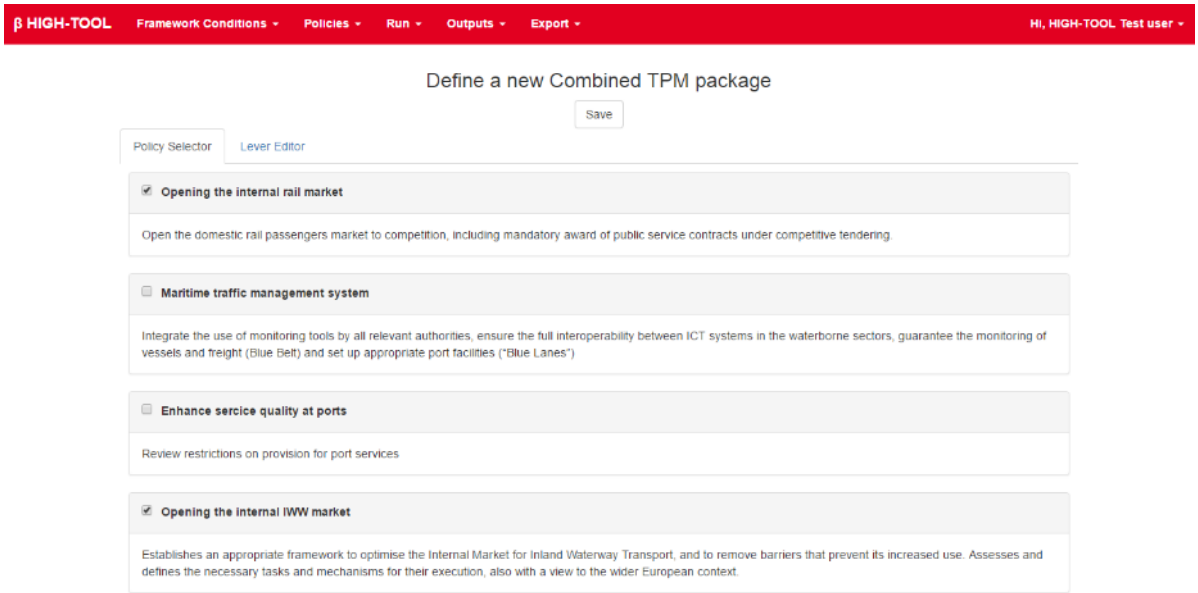


Figure 7: Selecting a combination of TPMs on the interface

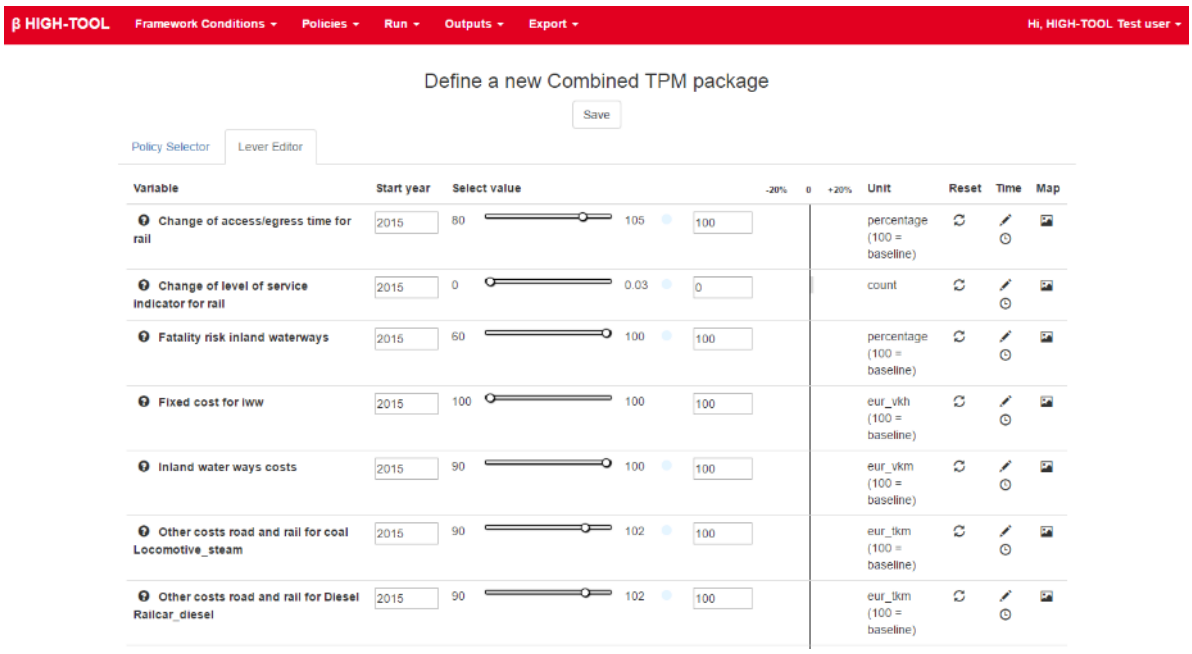


Figure 8: The policy levers for the combined TPM measure

Some policy levers are common to different TPMs, but the interface will show only one instance of these repeated policy levers in order to avoid the possibility of entering contradictory information. The lower and upper limits that the combined policy lever can take depend on the TPMs being combined.

The HIGH-TOOL consortium has analysed all possible TPM combinations on WP4 and found that, while most of times the combined effect of TPMs is purely additive, in some cases, it has a limited effect. The interface handles this information on the possible combinations and delivers common extended value ranges for the combined policy levers:

Whenever a combined policy lever is purely additive, then the lower and upper limits are increased by adding the limits of each instance of the policy levers (e.g. if one instance of the policy levers has ±5% and another has ±10%, then the combination will show ±15%).

If the combination of TPMs has a limited effect, then one out of three possible formulations is applied to calculate the combined lower and upper limits of the policy lever (the one chosen depends on the nature of the combination and is not controlled by the user):

- From all possible instances of the policy lever, the maximum upper and minimum lower values are chosen.
- A combined limit calculated by adding the limits of the widest instance first, then adding the 50% of the limits of the second widest range, then adding a 25% of the third widest instance and so on for each instance (e.g. there are two instances of the same lever, one with ±5% limits and the other with ±10% limits, then the combined limits will be ±12.5%)
- A mix of the two previous formulations for some combinations of three TPMs. First, a combined limit is calculated from two instances of the policy lever and then we select the maximum between this value and the third instance.

Apart from this special behaviour on the policy lever limits, the interaction of the user is equal to that of the Single TPM menu and contains the same tools: the user can modify the values for each policy lever globally, edit the time trajectories or apply different values across the EU map for countries or NUTS-2 regions.

3.3.3 Defining Customised Transport Policy Measures at Aggregated EU Level

The interface includes a third option for defining a policy scenario that gives more freedom to the user. The customised TPM allows selecting any combination of all available individual policy levers.

The policy levers are organised in tabs according to the module they belong to, and their exact number and classification is shown on Table 4.

Table 4: Available policy levers by module on the customised TPM interface

Module	Number of individual levers
Demography	0
Economy and Resources	3
Vehicle Stock	430
Passenger Demand	100
Freight Demand	79
Environment	127
Safety	60

Figure 9 shows the arrangement of the interface for customised TPM.

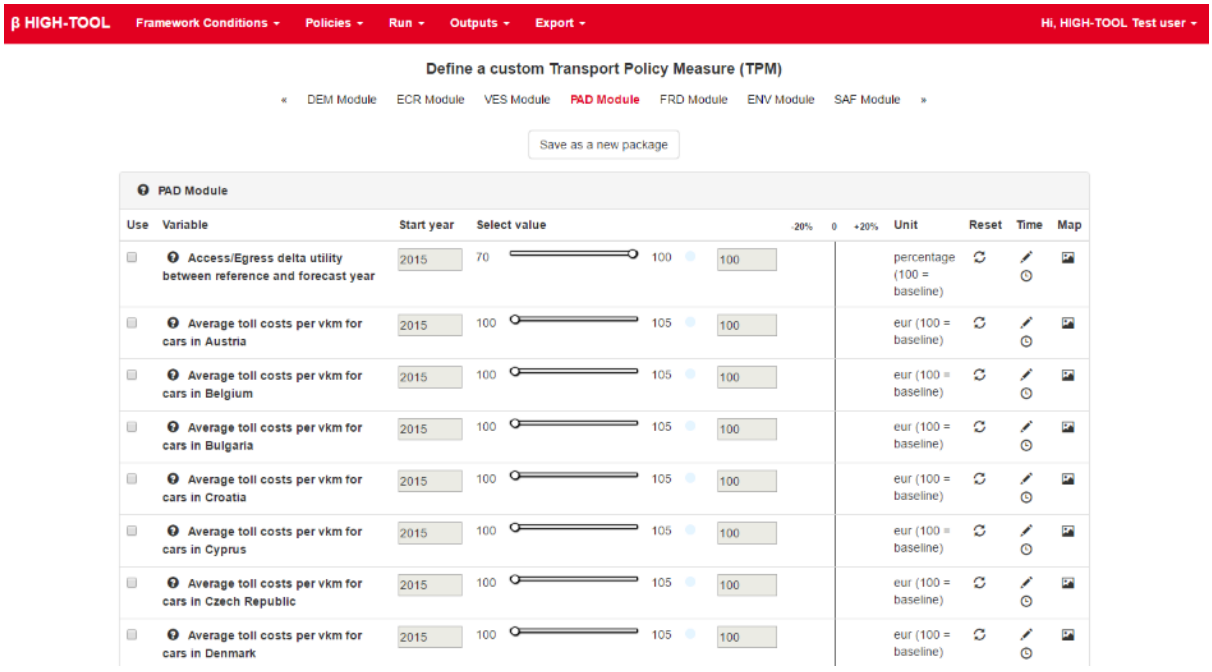


Figure 9: Interface for building a customised TPM

The way of interacting with the policy levers is identical to that of the interface for Single TPM but, in order to use any given policy lever, the user must first tick on the checkbox of that lever so that it becomes active.

The list of available policy levers is more extensive than that of the Single TPM interface, as there are some levers that are not used in any predefined TPM but exist nonetheless as possible inputs for HIGH-TOOL (e.g. "Load capacity for rail" or "Purchase VAT for electric cars").

The lower and upper limits for each policy lever are in general wider than the ones appearing on the Single TPM interface, so that the user has a higher degree of freedom for defining more intense policies.

3.3.4 Definition of Time Evolutions 2010–2050 for Input Variables

The default behaviour of the interface for all policy levers is that the policy has a time profile defined only with two inputs by the user: the final value and the year of application.

A resulting profile example is shown on Figure 10. In this case, the user has selected 80 as the value for the policy and 2020 as the year of application of the policy.

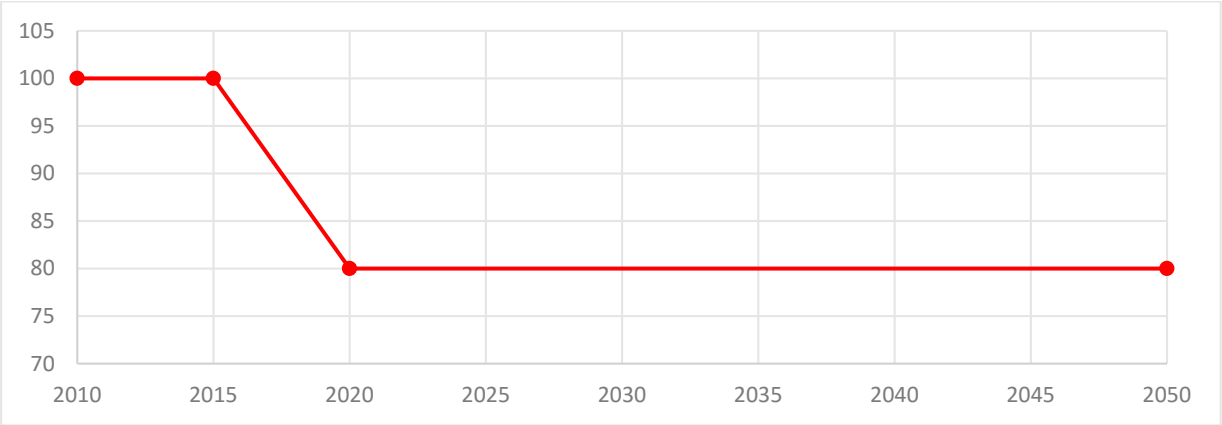


Figure 10: Default time profile of a policy lever

However, the interface has two tools to modify the time profile, which will appear for any policy lever available on the interface:

- The first one is the numerical editor, selectable by clicking on the icon
- The second one is the graphic editor, selectable by clicking on the icon (see Figure 11).

Figure 11: Accessing the numerical and graphic editors

The **numerical editor** is a table allowing introducing intermediate years and values for the selected policy lever. This way the policy implementation can be progressive as certain policies may be introduced stepwise. The path from the current situation in 2010 (e.g. *current level of road investments*) until the final policy level (year 2050) can be reflected (e.g. *level of road investments in 2050*).

The following options are available (see Figure 12):

- To add a new row click on the icon + and on the boxes that appear you can write a year and a value.
- Click on the icon to reorder the rows chronologically.
- Click on the icon × to remove any row.
- Click on "validate list" to correct automatically the values if they are beyond the bounds, to remove duplicate rows having the same year and to order them chronologically.
- By clicking "save" all the modifications are stored in the policy scenario being built.

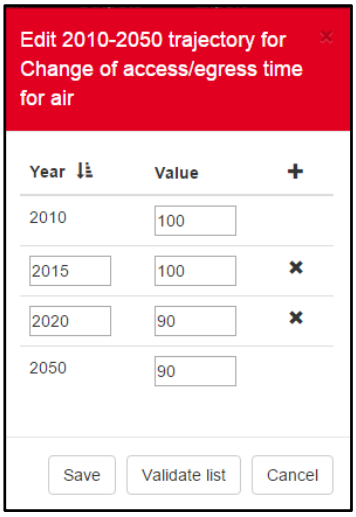


Figure 12: Interface for the numerical trajectory editor

For a more accurate and visual definition of the time trajectory of a policy lever, the interface offers the **graphical editor** that will launch an editable graphic with the years in the horizontal axis and the policy intensity on the vertical axis (see Figure 13).

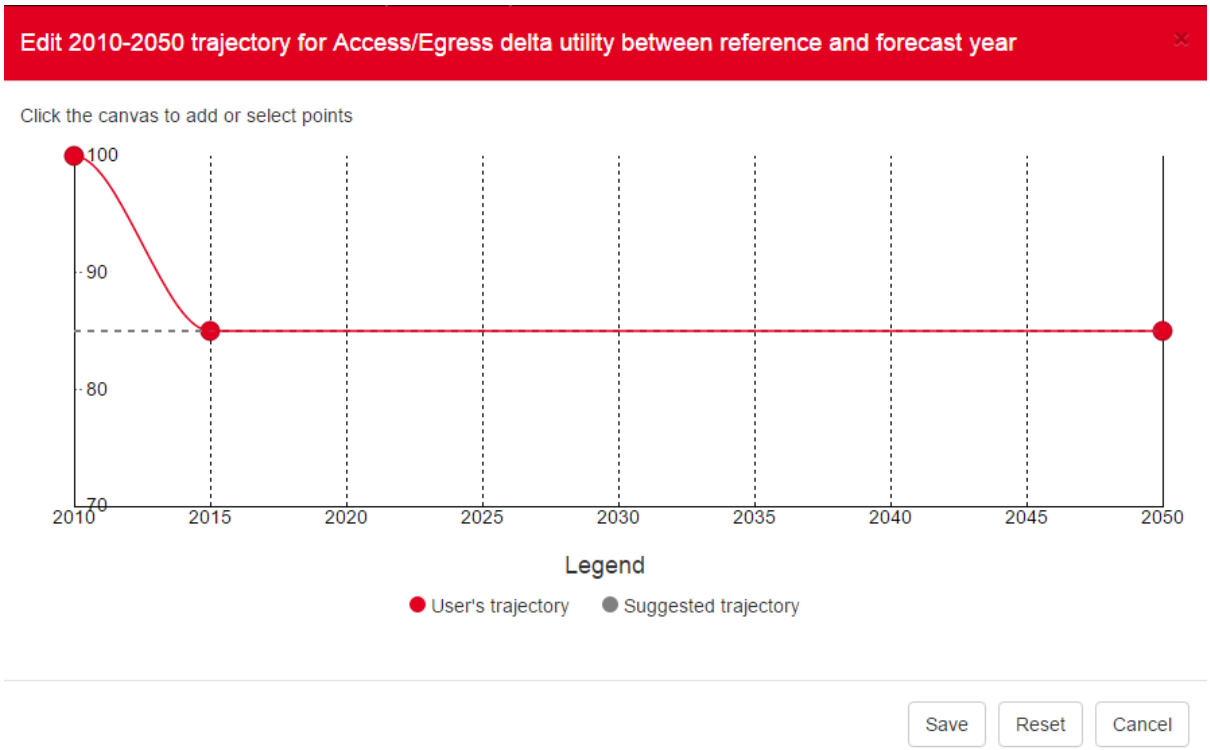


Figure 13: Interface for the graphical trajectory editor

The graphical editor allows users to change any value shown on the graphic by selecting and moving the existing red dots (except the value for 2010 which is fixed), or to insert a new control point by clicking at any point in the canvas. After adding a point, the application interpolates using Bezier curves.

By clicking on a control point and pressing the delete key, the point will be removed.

Upon opening the trajectory editor for the first time in a given policy lever, it will show the default trajectory defined in the TPM editing interface or, if it has been used, the one created with the numerical trajectory editor.

As a reference, a grey dotted trajectory is provided to show the initial suggestion.

The trajectories defined by the numerical and graphic editors are applied uniformly to all EU28 countries. The shown curve is sampled at the exact calculation years (2015–2050 every five years) for use by the model, so any maximum or minimum value at any year not being a multiple of five will not be considered in the calculations.

After saving a new trajectory, the icon on the main interface will become red to indicate that values differ from the default ones. By clicking on the reset button on the main interface, all changes on the graphical editor will also be removed.

3.3.5 Definition of Input Variables at Country and Regional Level

The map editor will appear when clicking the icon for a given policy lever available in the Transport Policy Measures editor (see Figure 14). Not all policy levers allow changes in this geographic dimension, due to the structure of the modules and the formulas used for the calculations, as they would not be meaningful. If a variable cannot be modified via the map editor, then this option will not be active and the icon will be greyed out.

Figure 14: Accessing the map editor

Using the map editor, values can be particularised at regional level for the different regions in Europe. For example, the level of road investments may be set higher in Eastern European countries in comparison with Western European countries.

After clicking the map editor icon, the application canvas opens showing a map of Europe (see Figure 15). By default, all the European territory will be displayed equally meaning that the input parameter or variable is being considered uniformly applied across Europe. Therefore, the colour is homogeneous reflecting an equal intensity.

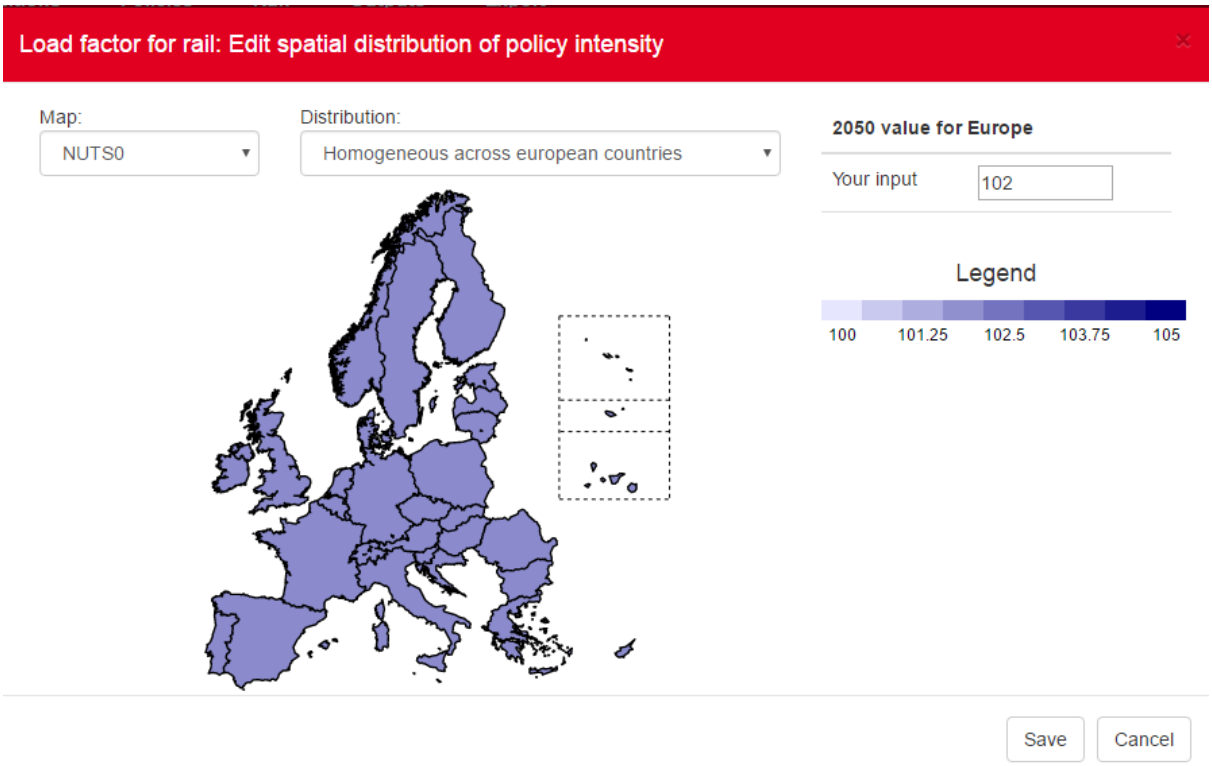


Figure 15: Geographic distribution of a parameter (homogeneous)

As some policies might be more or less appropriate for some Member States, selectors of the dialog can be used to change the distribution of the policy based on different criteria (see Figure 16):

- Homogenous distribution (default setting)
- Proportional/inversely proportional to population density
- Proportional/inversely proportional to income per capita.

For example, the intensity of road pricing can be set more intense in wealthy territories while less intense in lagging ones. Alternatively, the intensity of CO_2 taxation can be set more intense in most densely populated areas in Europe and less intense in sparsely populated areas).

These predefined settings are applied always in the same direction independently of what "more intensity" of the policy means. So when choosing for instance directly proportional to population density, the policy will take higher values on more densely populated regions and lower on the less dense ones, but irrespectively of the fact that higher values might imply less intensity. If the effect obtained by using a predefined setting is contrary to the one desired, please choose the opposite pre-set setting.

Territorial distributions of input parameters can be automatically defined based on NUTS-0 or NUTS-2.

Some policy levers do not allow changing values at NUTS-2 level, as this would not be meaningful from a policy perspective. In these cases the selector NUTS-0/NUTS-2 will not be active.

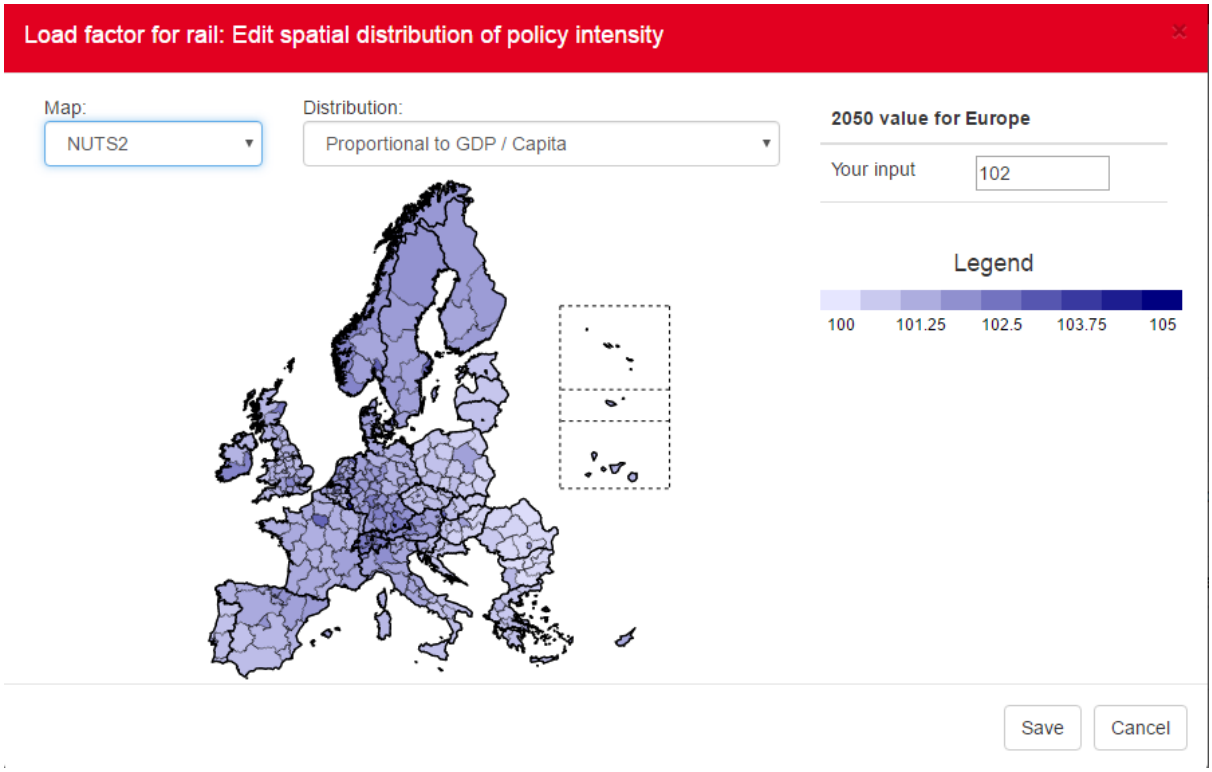


Figure 16: Geographic distribution of a parameter (proportional to population density of NUTS-2 regions)

Users can also manually change a specific country or region's value by clicking on the country/region and changing its input parameter on the right side of the map editor ("Your input" option, see Figure 17). While hovering the mouse over a region, the region name and parameter value in 2050 will be displayed. Double clicking an area of the map allows zooming in for better visualisation of territorial detail.

Whenever a country/region has a manual input of data, it is changed to a striped pattern for the sake of keeping track of the active modifications.

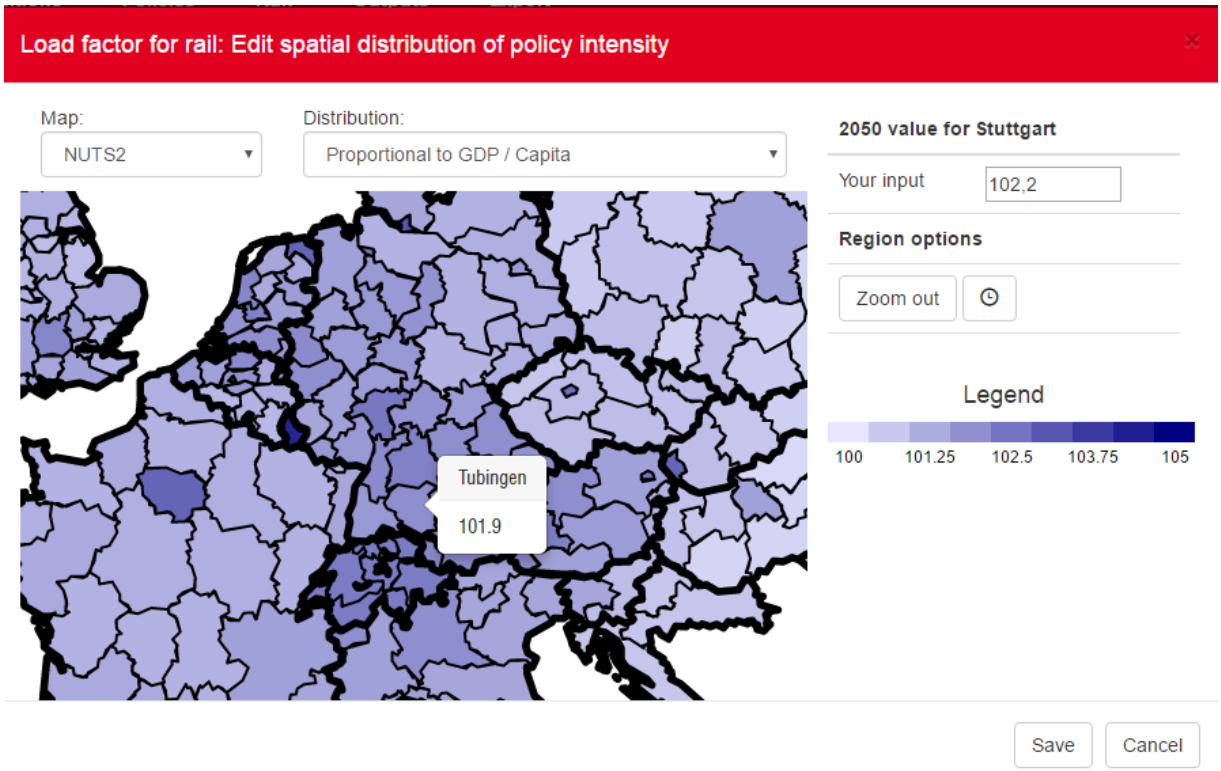


Figure 17: Manual modification of a value for a specific region

Whenever a specific country or region is selected, there is also the option to input a time trajectory by clicking on the icon. This will launch a new window with a graphic editor that works in the same way as the feature explained on chapter 3.3.4.

The country/region time editor is shown on Figure 18. Apart from setting the trajectory in the canvas by adding/removing control points, the user needs to click on the "Set" button to save the changes. There is a button to reset the trajectory to the general one applied for the whole EU28 and finally the button "Done" will return to the map.

After saving a new map, the icon on the main interface will become red to indicate that values differ from the default ones. By clicking on the reset button on the main interface, all changes on the map editor will also be removed.

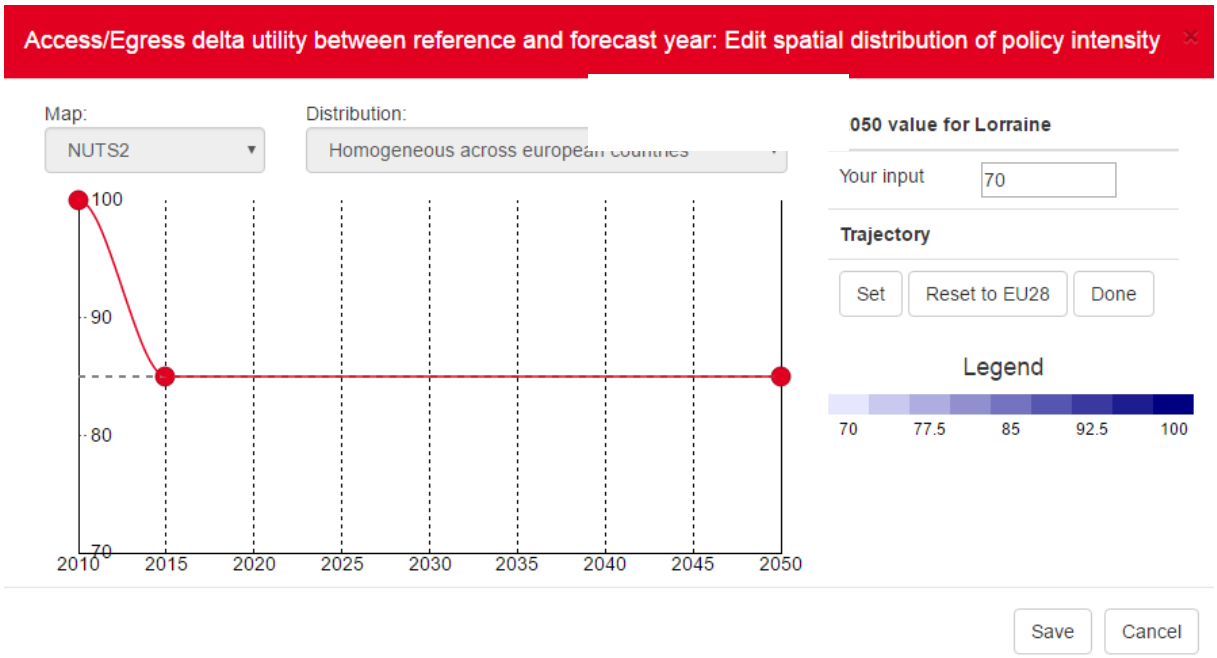


Figure 18: Graphic time editor for country/region

3.3.6 Policy Package Management

For any given policy package (single TPM, combined TPM, customised TPM) several possible actions can be called from the Policy menu list:

- **Load a policy package**: The user selects a previously created policy package and uploads it to the HIGH-TOOL interface.
- **Delete a policy package**: The user deletes a previously created policy package.
- **Save**: The user saves the changes made to a policy package previously loaded to the HIGH-TOOL interface.
- **Save as a new policy measure**: The user makes a copy of the active policy package under a new name.

3.4 Expert Mode

After having defined any transport policy measure through one of the three available procedures (Single TPM, Combined TPM, Custom TPM), the user can then enter the expert mode with which to edit input tables of the model at a low level and/or edit the Hyper-Network to control the impedances used in the model. This feature is optional and reserved to users that wish to make some advanced editing to the actual database values before running the model.

The expert mode comes after having made changes to the TPMs using the main interface, because these need to be applied in advance so that they show up on the raw tables. Thus, the logical flow of work in the model is as displayed in Figure 19.

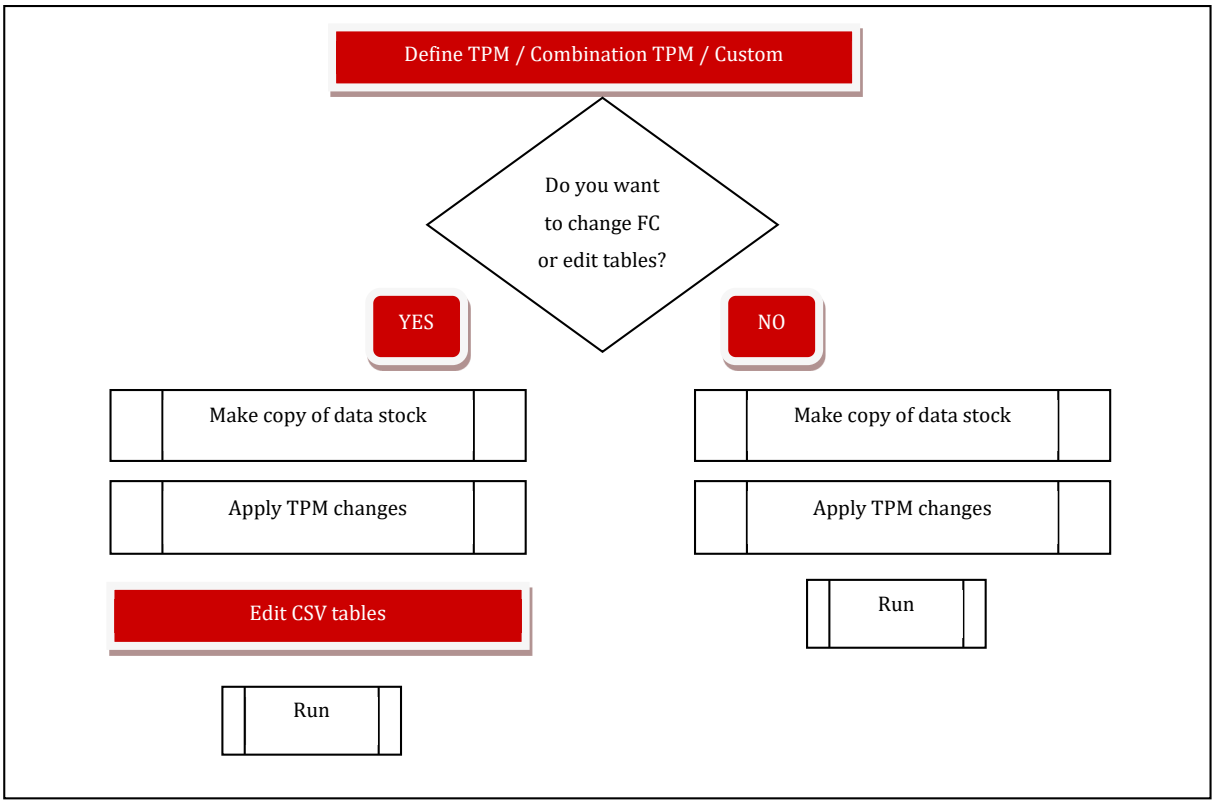


Figure 19: Workflow of the Interface

Using the expert mode the user can check and adjust the variables that have been previously affected by the changes in policy levers, as well as change other model parameters or inputs such as Framework Conditions. To get to the expert mode, tick on the checkbox at the window for selecting a policy package to run (see Figure 20).

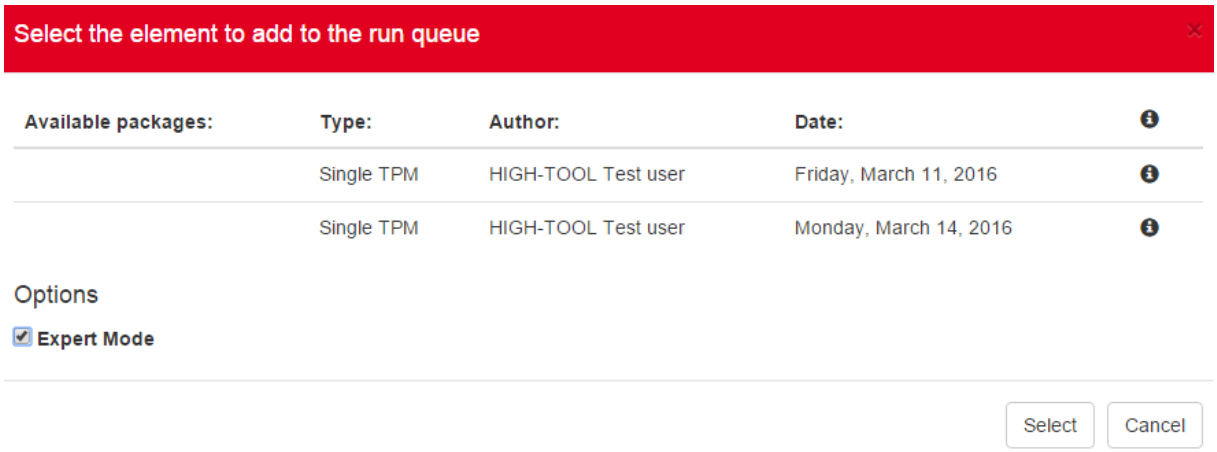


Figure 20: Enabling the expert mode

After launching the policy package to run with the expert mode enabled, the system will start the process of copying the data stock including the modifications from the policy package. Once the copy is ready, you may enter the expert mode by clicking on the icon in the "View and manage your model runs" window or by choosing the option "Packages ready for Expert Mode" from the run menu and selecting the policy package to edit from the list. This will open a new window (see Figure 21).

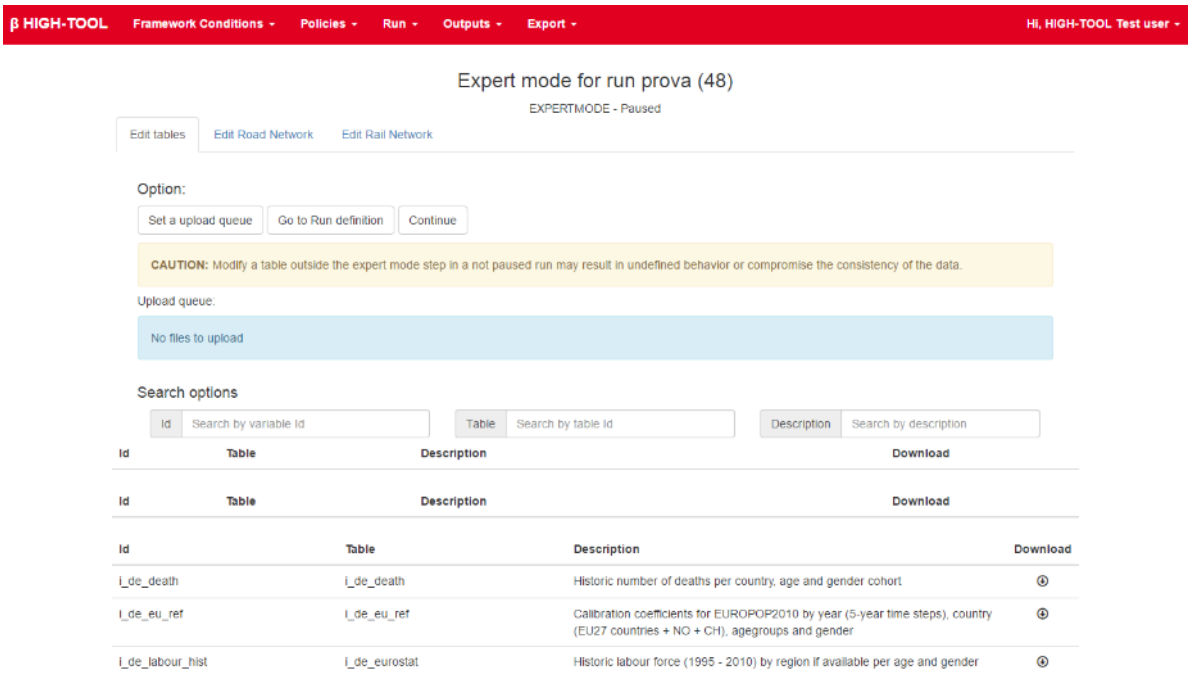


Figure 21: Expert mode interface

On the expert mode interface we can access two different tools, the table editor and the Hyper-Network editor, using the tabs on the upper part of the interface.

3.4.1 Edit Tables

The "Edit tables" tool (see Figure 21) offers a direct connection to the Data Stock, showing the full list of available input variables, the table they belong to, and a description. Three search fields allow a fast and easy filtering of the displayed items.

The available inputs are policy levers but also Framework Conditions and model parameters. Manually modifying any of these inputs might result in non-meaningful results of the model if the chosen values are outside the limits of applicability, as this might imply the need of a recalibration of the equations inside each module. Please refer to D4.2 and consult the metadata of the tables in the Data Stock before using this feature.

In order to make a change to any input value:

- Find it on the list by using the search fields.
- Download the table by clicking on the icon .
- Open the downloaded csv file with a spreadsheet editor, modify the desired values and save it (please refer to the Data Stock documentation to obtain a list of the codes used on each table).
- Upload the table by clicking on the button "Set upload queue" and then selecting the modified csv file using the file explorer.

These steps can be repeated for any number of inputs. Once all desired variables are modified, the simulation scenario is ready to continue by clicking on the "Upload all valid tables" button (this button appears once there is at least one table in the upload queue). Finally, on the run window (see Figure 23) click on the resume icon so that the model carries on with the simulation.

3.4.2 Edit Hyper-Network

HIGH-TOOL offers an aggregated way of changing the network impedances of the different transport modes, through the policy levers available in the TPM interfaces.

However, if the user wishes to explore and make detailed changes to the network, the expert mode has the Hyper-Network interface, which is a graphical simplified representation of the road and rail connections between NUTS-2 regions (see Figure 22).

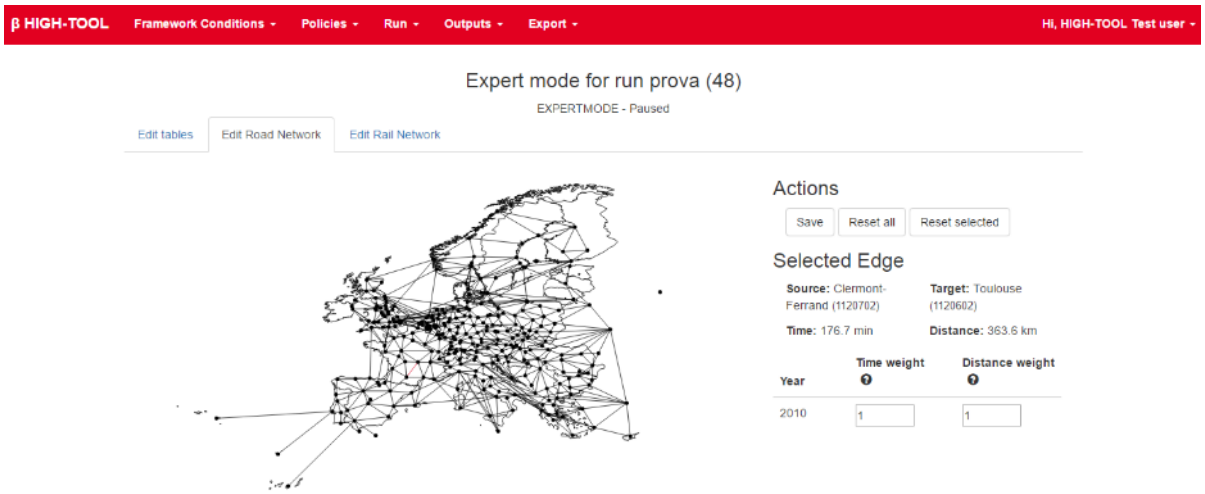


Figure 22: Editing the road Hyper-Network

The Hyper-Network editor has a navigable map that can be zoomed using the mouse wheel. Clicking on an edge will show its associated data at the right side of the screen. In addition, the user can input data on the fields for the relative time and distance weights for the different years. Therefore, the user cannot change the absolute values of time and distance for a given relation, but can affect it by setting the relative change.

Clicking on a node will also show its associated data at the right side of the screen. In addition, the user can input data on the fields for the relative access/egress time and distance weights for the different years. Therefore, the user cannot change the absolute values of time and distance for a given relation, but can affect it by setting the relative change.

The button "Save" stores the values, edited edges and nodes are shown in red. The reset button returns the values to their initial state.

All changes done to the edges and nodes will be used by HIGH-TOOL to compute new impedances for the Passenger module.

3.5 Run Menu

The Run menu controls the runs of the model and allows sending a new run based on any TPM package previously defined through the different tools including the Expert mode.

The option "View and manage model runs" opens a page that shows the existing model runs on the server, either finished or in process (see Figure 23).

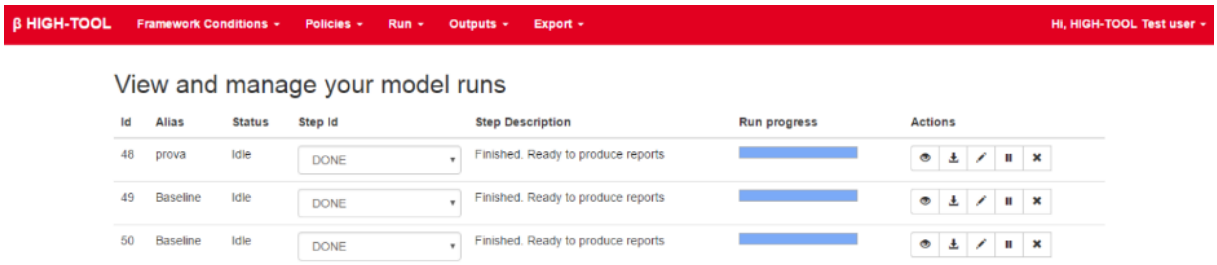


Figure 23: View and manage your model runs

On this page, we see the runs that belong to our user profile organised in a table with the following elements:

- **Id**: a number assigned to the run. It is an internal code with no use for the final user.
- **Alias**: a text describing the run, taken from the input of the user.
- **Status**: an explanation of the current status of the run. It can be one of the following options:
  - Active: indicating that the server is running this scenario
  - Paused: indicating that the run has been paused but has not finished yet
  - Paused and waiting for Expert Mode: indicates that the copy of the Data Stock is ready and the user might do manual changes to the tables
  - Idle: indicating that the run has been completed
  - Error: showing that the run has failed for some reason. This is of use for the administrator to solve it.
- **Step Id**: indicates the point the simulation is at that moment. If the run is paused the user can select any step to make the model recalculate from that point onwards.
- **Step Description**: a text explaining the current step that the interface is running.
- **Run progress**: a graphical bar showing the progress of the run.
- **Actions**: five icons that do different tasks shown in a tooltip (view the results of the run, download tables of the run, edit the tables, pause the run, cancel the run).

4 Policy Assessment Report

Successful runs will generate a Policy Assessment Report, which will remain stored on the server. The user will be able to retrieve results from previous runs as well.

Within the Policy Assessment Report, users will be able to:

- **Visualise Synthesis Indicators** online, including the user inputs and critical results using a limited number of indicators. Synthesis Indicators are intended to be policy relevant. They ought to fully illustrate scenario results produced by a HIGH-TOOL model run.
- **Download the full report in Excel format** including all results disaggregated by territories and modes, supported by tables and figures.
- **Download any table from the Data Stock** either related to inputs or outputs of the model run, in CSV format to be displayed in a spreadsheet editor.

4.1 Policy Assessment Report in MS Excel

A selection of the relevant simulation results are available in a report on MS Excel format, downloadable from the server through the User Interface. This report is generated automatically by the User Interface and designed as an interactive Excel report. Tables, graphs or single values can be imported to a presentation tool (MS PowerPoint) or a text editor (MS Word, OpenOffice) by using the ordinary copy and paste functionalities.

The full Policy Assessment Report contains the following elements:

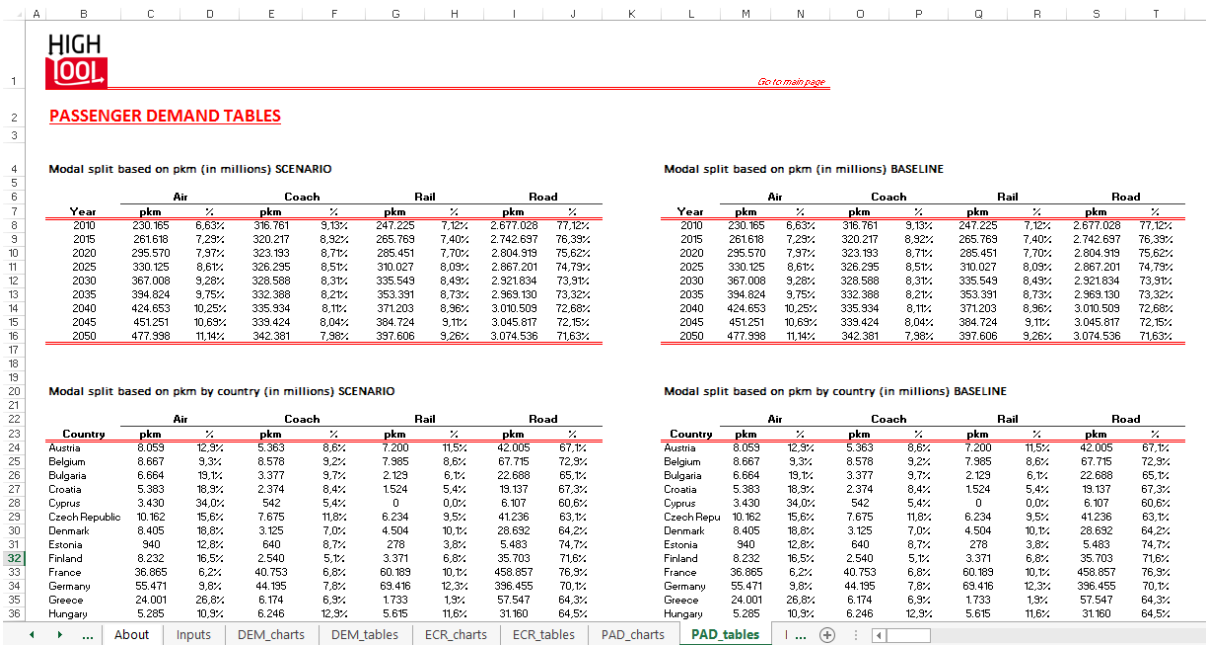
Contextual information including the name of the Simulation package and its abstract, as well as the names of the Transport Policy Measures being simulated and the underlying socio-economic Framework Conditions.

Synthesis of assumptions: Contains synthesis information related to a Transport Policy Measure applied. These are the user inputs and contain values for the policy levers, their time trajectories and the associated distribution maps.

Full results by thematic area: Results of each thematic area (demography, economy including resources, passenger transport, freight transport, safety, and environment) produced by the HIGH-TOOL model are presented by tables and charts, each on a separate page.

The following sequence of figures illustrates the assessment report produced based on preliminary results (Figure 24 to Figure 27).

Figure 24: Main menu of the Policy Assessment Report in MS Excel



HIGH TOOL

Go to main page

PASSENGER DEMAND TABLES

Modal split based on pkm (in millions) SCENARIO

	Air		Coach		Rail		Road	
Year	pkm	%	pkm	%	pkm	%	pkm	%
2010	230.165	6,63%	316.761	9,13%	247.225	7,12%	2.677.028	77,12%
2015	261.618	7,29%	320.217	8,92%	265.769	7,40%	2.742.697	76,39%
2020	295.570	7,97%	323.193	8,71%	285.451	7,70%	2.804.919	75,62%
2025	330.125	8,61%	326.295	8,51%	310.027	8,09%	2.867.201	74,79%
2030	367.008	9,28%	328.588	8,31%	335.549	8,49%	2.921.834	73,91%
2035	394.824	9,75%	332.388	8,21%	353.391	8,73%	2.969.130	73,32%
2040	424.653	10,25%	335.934	8,11%	371.203	8,96%	3.010.509	72,68%
2045	451.251	10,69%	339.424	8,04%	384.724	9,11%	3.045.817	72,15%
2050	477.998	11,14%	342.381	7,98%	397.606	9,26%	3.074.536	71,63%

Modal split based on pkm (in millions) BASELINE

	Air		Coach		Rail		Road	
Year	pkm	%	pkm	%	pkm	%	pkm	%
2010	230.165	6,63%	316.761	9,13%	247.225	7,12%	2.677.028	77,12%
2015	261.618	7,29%	320.217	8,92%	265.769	7,40%	2.742.697	76,39%
2020	295.570	7,97%	323.193	8,71%	285.451	7,70%	2.804.919	75,62%
2025	330.125	8,61%	326.295	8,51%	310.027	8,09%	2.867.201	74,79%
2030	367.008	9,28%	328.588	8,31%	335.549	8,49%	2.921.834	73,91%
2035	394.824	9,75%	332.388	8,21%	353.391	8,73%	2.969.130	73,32%
2040	424.653	10,25%	335.934	8,11%	371.203	8,96%	3.010.509	72,68%
2045	451.251	10,69%	339.424	8,04%	384.724	9,11%	3.045.817	72,15%
2050	477.998	11,14%	342.381	7,98%	397.606	9,26%	3.074.536	71,63%

Modal split based on pkm by country (in millions) SCENARIO

	Air		Coach		Rail		Road	
Country	pkm	%	pkm	%	pkm	%	pkm	%
Austria	8.059	12,3%	5.363	8,6%	7.200	11,5%	42.005	67,1%
Belgium	8.667	9,3%	8.578	9,2%	7.985	8,6%	67.715	72,9%
Bulgaria	6.664	19,1%	3.377	9,7%	2.129	6,1%	22.688	65,1%
Croatia	5.383	18,9%	2.374	8,4%	1.524	5,4%	19.137	67,3%
Cyprus	3.430	34,0%	542	5,4%	0	0,0%	6.107	60,6%
Czech Republic	10.162	15,6%	7.675	11,8%	6.234	9,5%	41.236	63,1%
Denmark	8.405	18,8%	3.125	7,0%	4.504	10,1%	28.632	64,2%
Estonia	940	12,8%	640	8,7%	278	3,8%	5.483	74,7%
Finland	8.232	16,5%	2.540	5,1%	3.371	6,8%	35.703	71,6%
France	36.865	6,2%	40.753	6,8%	60.189	10,1%	458.857	76,9%
Germany	55.471	9,8%	44.195	7,8%	69.416	12,3%	396.455	70,1%
Greece	24.001	26,8%	6.174	6,9%	1.733	1,9%	57.547	64,3%
Hungary	5.285	10,9%	6.246	12,9%	5.615	11,6%	31.160	64,5%

Modal split based on pkm by country (in millions) BASELINE

	Air		Coach		Rail		Road	
Country	pkm	%	pkm	%	pkm	%	pkm	%
Austria	8.059	12,3%	5.363	8,6%	7.200	11,5%	42.005	67,1%
Belgium	8.667	9,3%	8.578	9,2%	7.985	8,6%	67.715	72,9%
Bulgaria	6.664	19,1%	3.377	9,7%	2.129	6,1%	22.688	65,1%
Croatia	5.383	18,9%	2.374	8,4%	1.524	5,4%	19.137	67,3%
Cyprus	3.430	34,0%	542	5,4%	0	0,0%	6.107	60,6%
Czech Repu	10.162	15,6%	7.675	11,8%	6.234	9,5%	41.236	63,1%
Denmark	8.405	18,8%	3.125	7,0%	4.504	10,1%	28.632	64,2%
Estonia	940	12,8%	640	8,7%	278	3,8%	5.483	74,7%
Finland	8.232	16,5%	2.540	5,1%	3.371	6,8%	35.703	71,6%
France	36.865	6,2%	40.753	6,8%	60.189	10,1%	458.857	76,9%
Germany	55.471	9,8%	44.195	7,8%	69.416	12,3%	396.455	70,1%
Greece	24.001	26,8%	6.174	6,9%	1.733	1,9%	57.547	64,3%
Hungary	5.285	10,9%	6.246	12,9%	5.615	11,6%	31.160	64,5%

... About | Inputs | DEM_charts | DEM_tables | ECR_charts | ECR_tables | PAD_charts | PAD_tables | ...

Figure 25: Full results displayed in tables, on a per theme basis (I)

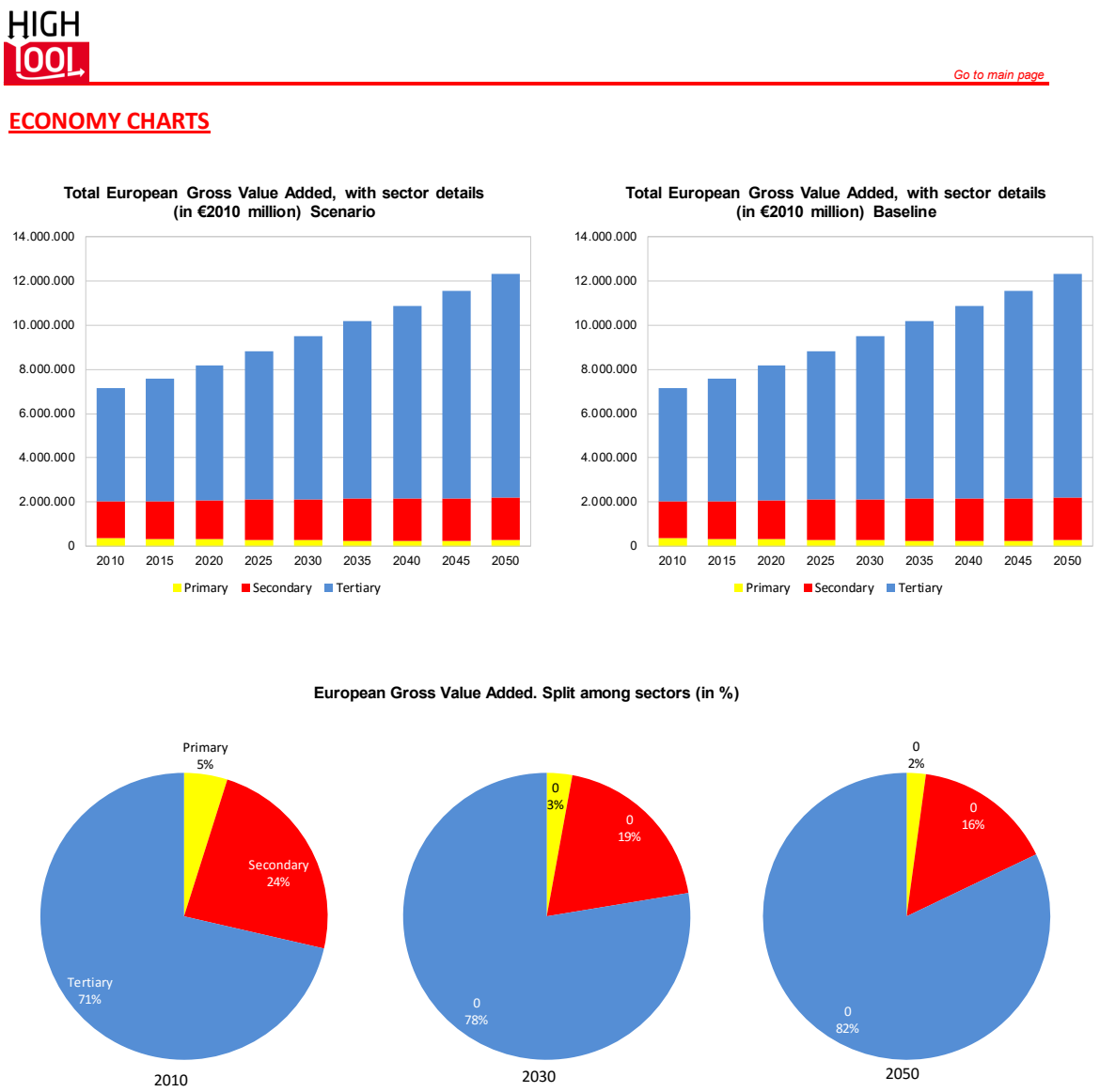


Figure 26: Full results displayed in charts, on a per theme basis (II)

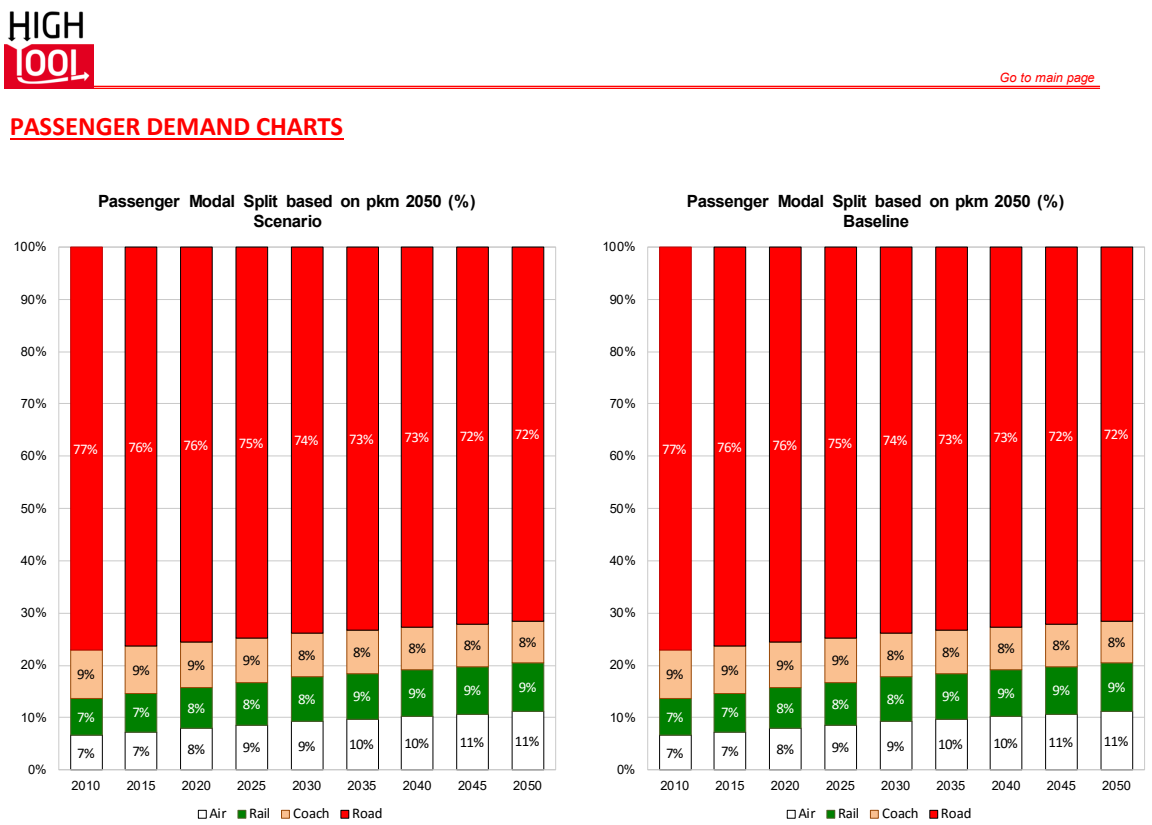


Figure 27: Full results displayed in charts, on a per theme basis (III)

4.2 Main Output Indicators

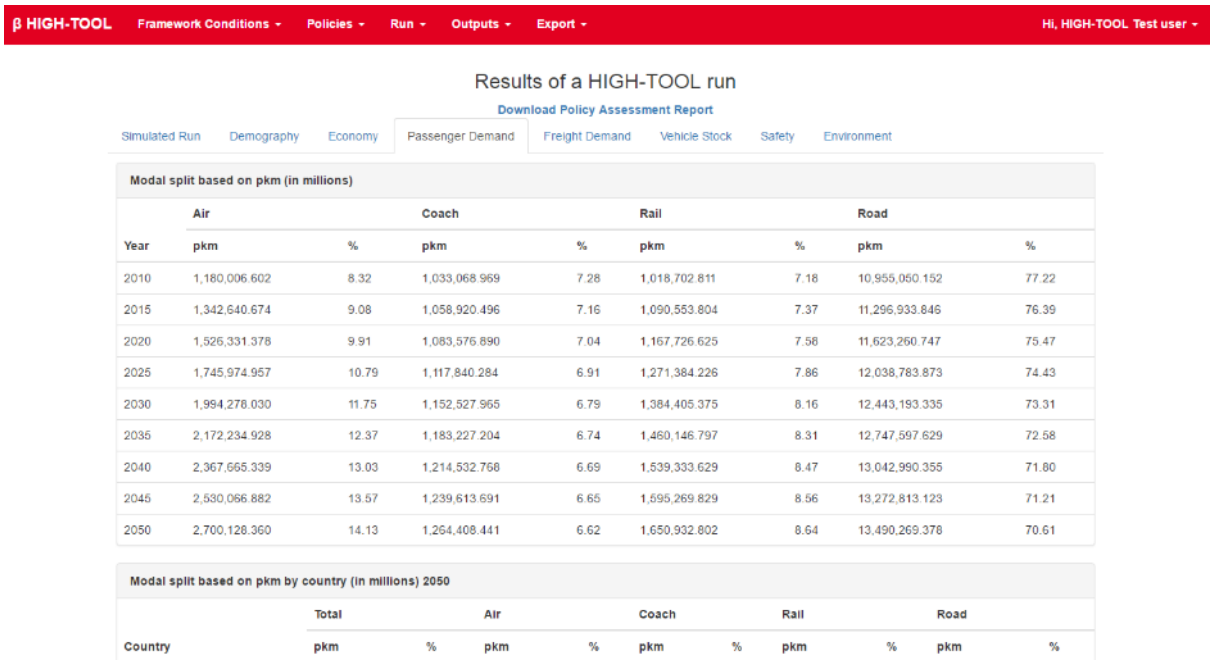
After a HIGH-TOOL model run, synthesis results are displayed online within the User Interface. In brief, synthesis indicators are intended to fully describe a model run by:

- Framework Conditions
- A synthesis of the Transport Policy Measures implemented
- A synthesis of impacts.

The following indicators are considered for the synthesis (see Table 5).

Table 5: Synthesis indicators of a HIGH-TOOL simulation

Framework Conditions	Policy Implementation	Impacts on Transport
European population	Values of the policy levers	Transport performance per mode (pkm, tkm)
Labour force	Time trajectories of the policy levers	Modal share
GDP and GDP/capita	Maps with policy intensity of each lever	Vehicle stock
GVA per sector		Accidents and fatalities per mode
		Fuel consumption
		CO_2 emissions from transport
		Pollutant emissions from transport



	Air		Coach		Rail		Road	
Year	pkm	%	pkm	%	pkm	%	pkm	%
2010	1,180,006.602	8.32	1,033,068.969	7.28	1,018,702.811	7.18	10,955,050.152	77.22
2015	1,342,640.674	9.08	1,058,920.496	7.16	1,090,553.804	7.37	11,296,933.846	76.39
2020	1,526,331.378	9.91	1,083,576.890	7.04	1,167,726.625	7.58	11,623,260.747	75.47
2025	1,745,974.957	10.79	1,117,840.284	6.91	1,271,384.226	7.86	12,038,783.873	74.43
2030	1,994,278.030	11.75	1,152,527.965	6.79	1,384,405.375	8.16	12,443,193.335	73.31
2035	2,172,234.928	12.37	1,183,227.204	6.74	1,460,146.797	8.31	12,747,597.629	72.58
2040	2,367,665.339	13.03	1,214,532.768	6.69	1,539,333.629	8.47	13,042,990.355	71.80
2045	2,530,066.882	13.57	1,239,613.691	6.65	1,595,269.829	8.56	13,272,813.123	71.21
2050	2,700,128.360	14.13	1,264,408.441	6.62	1,650,932.802	8.64	13,490,269.378	70.61

	Total		Air		Coach		Rail		Road	
Country	pkm	%	pkm	%	pkm	%	pkm	%	pkm	%

Figure 28: Synthesis indicators displayed on the User Interface of the HIGH-TOOL model

4.3 Export Table Contents

The full set of inputs and outputs of the model is stored inside a database that is accessible at any moment. In order to facilitate the access to this database without having to install extra software, the User Interface offers a tool that allows selecting any table contained in the database and downloading it in a CSV file. This tool is similar to the one used in the expert mode (see chapter 3.4). Tables downloaded in this way can be easily explored using any spreadsheet editor such as MS Excel.

The tool offers the option of selecting the database containing the results we want to explore and then three possible search fields to look for the information we are interested in. The search is dynamic and will present results as we type.

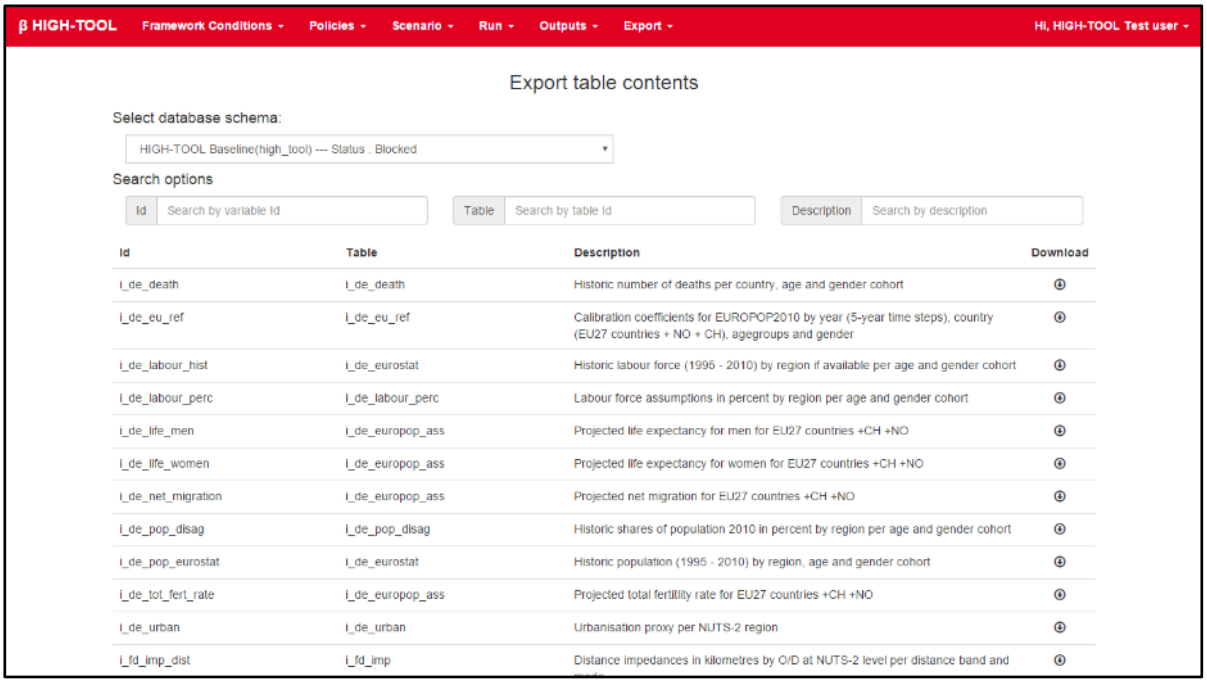


Figure 29: Export table utility in the interface of the HIGH-TOOL model

In order to obtain any table:

- Find it on the list by using the search fields
- Download the table by clicking on the icon ⊕
- Open the downloaded CSV file with a spreadsheet editor (please refer to the Data Stock documentation to obtain a list of the codes used on each table).

5 Stand-Alone Version

HIGH-TOOL runs from a server and is accessible via web browser. However, there is also the possibility to run it on a local computer.

There is a specific document explaining the steps needed to install this stand-alone version, please refer to it for more detail. This technical procedure requires some expertise, although the instructions try to be as easy to follow as possible.

The stand-alone version works in the same exact way from the point of view of the user, so the contents of chapters 3 and 4 apply to it.

The only difference is that the interface runs from Electron and not a web browser, so the actual appearance of some elements like the policy lever sliders could differ slightly due to the rendering (see Figure 30).

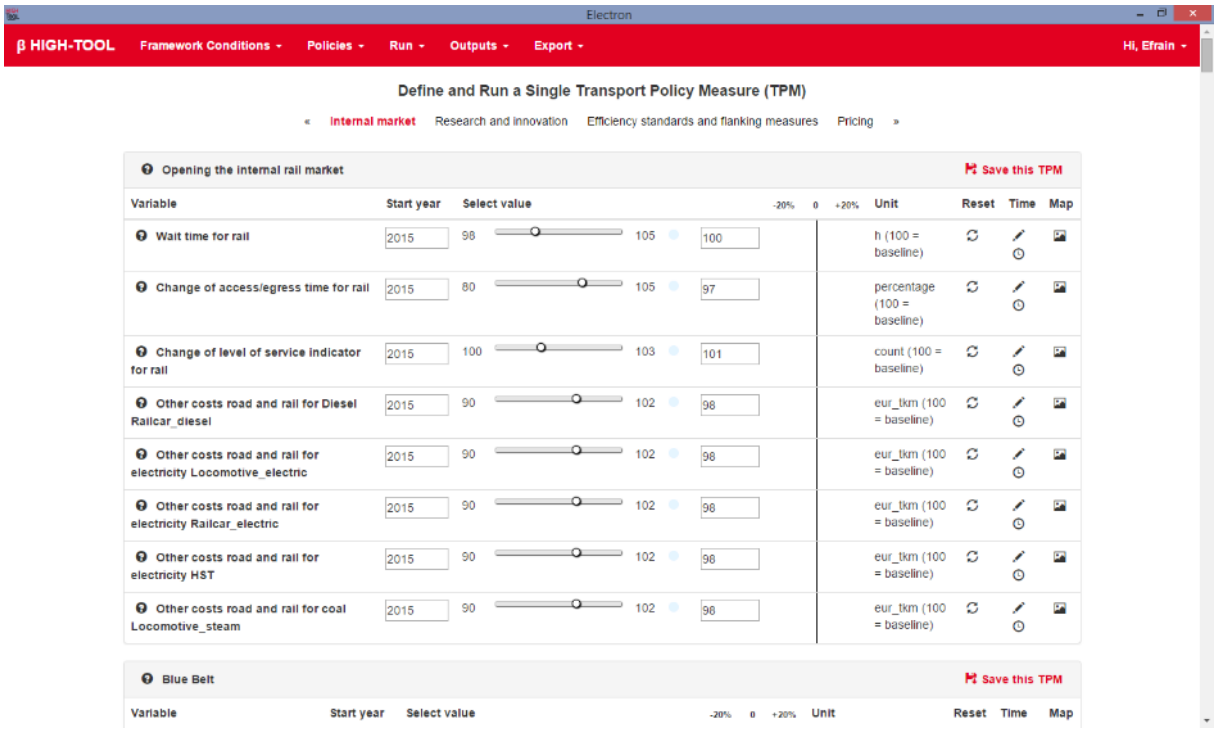


Figure 30: Stand-alone HIGH-TOOL interface

6 Future Improvements of HIGH-TOOL

During the development of HIGH-TOOL, the consortium has found a number of possible enhancements or add-ons to the model that could be desirable on future developments.

Development of a transport network interface: This feature would allow the implementation of transport investments interactively based on the hyper-network at NUTS-2 level. Any changes in the hyper-network will result in modified impedances applying a shortest path algorithm. This network feature is already implemented in a simplified version. It should be extended to the Freight module (now it only works with Passenger) and should allow adding new links. Also, traffic assignment routines could be implemented to show simplified traffic maps.

Consistency check: A simple pre-assessment tool could be implemented to test the magnitude of transport impacts obtained from selected inputs at an aggregated EU level. The tool would aim at providing the user with an order of magnitude or simply the direction of impacts so that he can adjust policies before performing a complete run.

Comparison of packages: The possibility that the User Interface can check and compare differences between two different simulation packages has been explored. It will identify which variable fields have different values between two packages, so that users can quickly apprehend main differences in packages.

Online graphics set: Adding a visualisation library would allow showing automatic graphics for the output results on the User Interface.

Changing the framework conditions: HIGH-TOOL could offer a set of predefined framework conditions prepared from future scenarios developed by third parties. Maybe even having a catalogue of the successive EU Reference Scenarios to test how they affect the policy outcomes.

Cloud computing: HIGH-TOOL runs now on a server or a local computer. In the future, it could be set up to run on the cloud, so that the hardware costs could be cut, also ensuring that it always runs on the fastest computer possible.

Warning system: the system will send the user an email informing that the run has been completed, that the expert mode is ready or that there has been an error.

7 References

European Commission (2009): Impact Assessment Guidelines, SEC(2009) 92, Brussels.

European Commission (2011a): White Paper, Roadmap to a Single European Transport Area – Towards a competitive and resource efficient transport system, COM(2011) 144 final, Brussels.

European Commission (2011b): Impact Assessment to the White Paper – Roadmap to a Single European Transport Area. Commission Staff Working Paper, SEC(2011) 358 final, Brussels.

European Commission (2011c): A Roadmap for moving to a competitive low carbon economy in 2050, COM(2011) 112 final, Brussels.

European Commission (2011d): Commission Staff Working Document, Accompanying the White Paper – Roadmap to a Single European Transport Area, SEC(2011) 391 final, Brussels.

European Commission (2012): The 2012 Ageing Report: Underlying Assumptions and Projection Methodologies. European Economy 4/2011, DG ECFIN, Brussels. http://ec.europa.eu/economy_finance/publications/european_economy/2012/pdf/ee-2012-2_en.pdf

European Commission (2013a): HIGH-TOOL, Kick-off meeting, FP7 Call TPT.2012.2-1. Strategic high-level transport model, presentation 18 April 2013, Brussels.

European Commission (2013b): EU Energy, Transport and GHG Emissions Trends to 2050, Reference Scenario 2013. http://ec.europa.eu/transport/media/publications/doc/trends-to-2050-update-2013.pdf.

European Commission (2014): The 2015 Ageing Report: Underlying Assumptions and Projection Methodologies. European Economy 8/2014, DG ECFIN, Brussels. http://ec.europa.eu/economy_finance/publications/european_economy/2014/pdf/ee8_en.pdf.

Eurostat (2011): Eurostat Population Projections 2010-based (EUROPOP2010): Methodology and results of a long-term scenario of demographic convergence. http://epp.eurostat.ec.europa.eu/portal/page/portal/population/publications/population_projections.

Mandel, B., Kraft, M., Schnell, O., Klar, R., Ihrig, J., Szimba, E., Smith, R., Laparidou, K., Chahim, M., Corthout, R., Purwanto, J. (2016): Final Structure of the HIGH-TOOL Model, Deliverable D2.2 of the HIGH-TOOL project, co-funded by the European Commission under the 7th Framework Programme. Karlsruhe, Germany.

Purwanto, J., Szimba, E., Mandel, B. (2015): HIGH-TOOL Deliverable D1.2, Users' Feedback on the HIGH-TOOL Model (Pre-Final Version), Co-funded by the European Commission under the 7th Framework Programme, Karlsruhe, Germany.

Szimba, E., Ihrig, J., Kraft, M. (2013): Inception Report, Deliverable D10.2 of the HIGH-TOOL project, project co-funded by the European Commission under the 7th Framework Programme. Karlsruhe, Germany.

Tognazzini, B. (2013): AskTOG, referenced on 17 June 2013. http://www.asktog.com/basics/firstPrinciples.html

Vanherle, K., Corthout, R., Szimba, E., Meyer, C., Kiel, J., Ulied, A., Biosca, O. (2013): HIGH-TOOL Deliverable D1.1 (Draft), User Requirements, Co-funded by the European Commission under the 7th Framework Programme, Karlsruhe, Germany.

Neither the European Commission nor any person acting on behalf of the Commission is responsible for the use that might be made of this information. The views expressed in this publication are the sole responsibility of the author(s) and do not necessarily reflect the views of the European Commission.