URBANITE

Supporting the decision-making in urban transformation with the use of disruptive technologies

Deliverable D4.6

Final implementation of the recommendation system for policy design

Editor(s):	Maj Smerkol, Gjorgji Noveski, Miljana Shulajkovska, Matjaž Gams
Responsible Partner:	"Jožef Stefan" Institute
Status-Version:	Final - v1.0
Date:	05/07/2023
Distribution level (CO, PU):	Public

Project Number:	GA 870338
Project Title:	URBANITE

Title of Deliverable:	Final implementation of the recommendation system for policy design
Due Date of Delivery to the EC:	31/03/2023

Workpackage responsible for the Deliverable:	WP4 – Algorithms and simulation techniques for decision-makers
Editor(s):	"Jožef Stefan" Institute
Contributor(s):	JSI, TEC
Reviewer(s):	FVH
Approved by:	All Partners
Recommended/mandatory readers:	WP3, WP5, WP6

Abstract:	D4.6 Final implementation of the recommendation system for policy design - OTHER (M30) This deliverable has a twofold goal. On one hand, it will present the final version of the URBANITE traffic flow model, and on the other hand, building upon D4.5, it will provide the implementations of the methods, tools, and mechanisms presented in D4.5. The result of this deliverable will be used in WP5 during the integration activities (T5.3 and T5.4).
Keyword List:	Recommender engine, recommendation system, optimization, multi attribute decision analysis
Licensing information:	
Disclaimer	

Document Description

Document Revision History

Version	Date	Modifications Introduced	
		Modification Reason	Modified by
v0.1	04/03/2023	Draft ToC	JSI
v0.2	23/03/2023	Added section on MCDA	JSI
v0.3	04/04/2023	Added section on Recommender engine implementation	JSI
v0.4	11/04/2023	Added section Recommender System in the Context of Mobility Policy	JSI
v0.5	03/05/2023	Updates to various section, several sections fully drafted	JSI
v0.61	08/06/2023	Final draft ready for review.	JSI
v0.62	20/06/2023	Added section on High-level Recommender Engine (RE), section on Acoustic pollution and annex with the Noise Computation API.	TECNALIA
v0.7	21/06/2023	Internal review done.	FVH
v1.0	05/07/2023	Final version ready for submission.	JSI

Table of Contents

List of Figures

List of Tables

Terms and Abbreviations

EC	European Commission
GIS	Geographical Information System
TS	Traffic Simulation module
DSS	Decision Support System
RE	Recommender Engine
EA	Evolutionary Algorithm
GA	Genetic Algorithm
k-NN	k-Nearest Neighbors
KPI	Key Performance Indicator
ML	Machine Learning
GUI	Graphical User Interface
MCDA	Multi Criteria Decision Analysis
MOO	Multi-Objective Optimization
NSGA	Non-Dominated Sorting Genetic Algorithm
OD	Origin-Destination (matrix)
MBI	Munich Bikeability Index

Executive Summary

This deliverable describes the recommendation system for policy design that is implemented as part of the URBANITE solution. This document describes the recommendation system overall and in relation to other URBANITE modules, the objectives of the system, and details the approach to providing recommendations and methodology used for decision analysis. The deliverable D4.5 Recommendation System for Policy Design described and planned for implementation of the recommendation system. This deliverable explains and provides the rationale for deviations from the initial design. The software system this document describes is available online and deployed in the four pilot cities.

The software system developed and described in this deliverable is part of the key result KR4 URBANITE algorithms, and the direct outcome of tasks T4.2. The recommender engine utilises the traffic simulation module described in the deliverable D4.4 URBANITE Traffic Flow Model, and the decision support system, described in the deliverable D4.3 URBANITE Policy Decision Model. The system developed is used by the work package WP5 as part of the modules integrated in tasks T5.3 and T5.4.

The document contains three main sections. The first, Recommendation System in the Context of Mobility Policy, overviews the related modules and how the recommender system relates to the platform. The second, Recommender Engine, describes the developed recommender engine, how to use it, and details the methods of implementation. The third, Multi-Criteria Decision Analysis, overviews the approach to decision modelling, describes the decision models created and the KPIs that are used as the basis of decision evaluation.

The software system developed utilises multi criteria decision analysis methodology as an evaluation framework to compare different mobility scenarios simulated and to provide simple but useful recommendations on what aspects of mobility to target to further improve scenarios.

1 Introduction

Deliverable D4.6 presents a comprehensive report on the implementation and utilization of the traffic simulation, analysis, recommendation, and visualization methods for the smart urban mobility platform developed within the scope of the URBANITE project. This final report builds upon the intermediate report presented in the deliverable D4.1, which focused on the methods applicable to the URBANITE domain and their use within the project, and the deliverable D4.5, which describes the initial design and plan for implementation of the recommendation system. The present document is a culmination of work package WP4 "Algorithms and simulation techniques for decision-makers" and tasks T4.1 "Methods for Exploratory Data Analysis and User Interaction," T4.4 "Advanced visualizations methods," and additional research performed during the project's execution.

This report highlights the implementation and integration of these methods in the URBANITE ecosystem, their application in real-world urban mobility and policy support scenarios, and the value they provide to end users, particularly city decision-makers.

1.1 About This Deliverable

This document details the data modelling, analysis, and visualization methods implemented within the URBANITE platform, as well as the advancements made since Deliverable D4.2. It discusses the rationale behind their selection, implementation, and integration into the platform and provides examples of their application in urban mobility and policy support contexts.

Furthermore, this document presents the integration of multi-criteria decision analysis using a program called DEXi, the recommender engine, and the Orange3 data mining tool extension for additional analysis and visualization capabilities. It also highlights the advanced visualization techniques employed, including up to 5-dimensional visualization of KPIs for different proposals. The DEXi program is useful for multi-attribute decision modeling and evaluation of options. It supports complex decision-making tasks and works in a hierarchical structure where a problem is decomposed into subproblems. This makes a good fit for our problem.

1.2 Document Structure

This document is organized into the following sections:

- Section 1 "Introduction" explains the rationale of this document, its relationship to Deliverable D4.2, and details the document structure.
- Section 2 "Recommendation System in the Context of Mobility Policy" provides a high-level description of the platform, its components, and features.
- Section 3 “Recommender Engine” describes the developed recommender engine, its benefits and limitations, and provides reasoning for deviating from the initial plan.
- Section 4 “High-level Recommender Engine” describes a recommender that complements the previous one, and is designed to support the initial steps of the planning, gaining the advantage of previous experiences on decisions.
- Section 5 “Multi-Criteria Decision Analysis” describes the methodology, developed custom decision models, and key performance indicators used in the analysis.
- Section 6 “Source Code” provides the links to the source code in the repository.
- Section 7 "Conclusions and Future Work" summarizes the achievements, challenges, and limitations of the project and outlines the path towards further improvements.
- Section 8 “References” provides the list of references
- Finally, section 9 describes the noise computation API.

2 Recommender System in the Context of Traffic and Mobility Policy

This section provides a high-level description of the smart urban mobility platform developed within the URBANITE project, focusing on its components and features designed to support decision-makers in the realm of urban transport and mobility policy. The platform comprises several interconnected modules, each tailored to address specific aspects of the decision-making process. These components include the data platform, the traffic simulation engine, the decision support system, advanced visualizations, and the exploratory data analysis module.

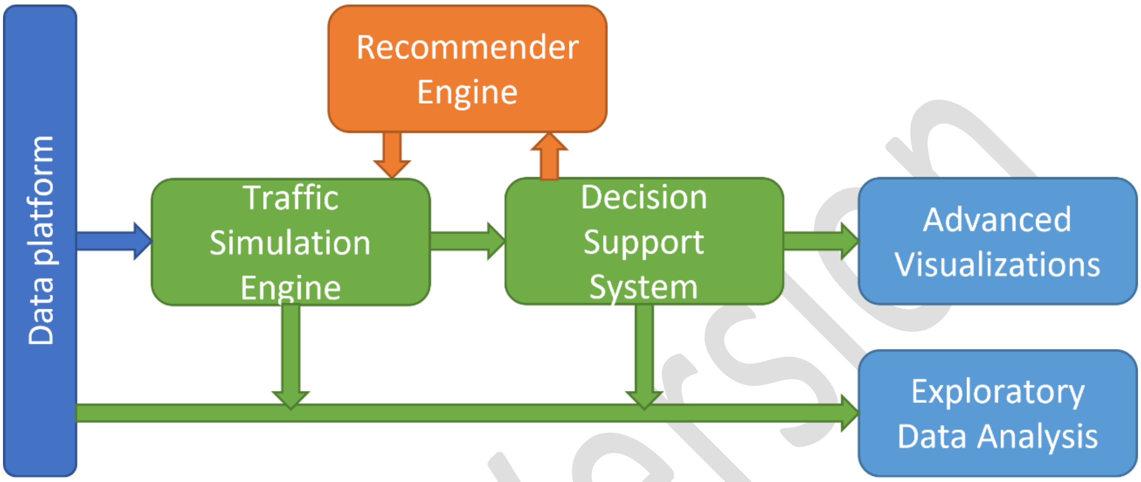

Figure 1: Overview of the components of the system.

2.1 Data Management Platform

The Data Management Platform serves as the foundation for the smart urban mobility platform, collecting, storing, and processing data from various sources, including urban mobility data, GIS information, and socio-economic data. This component ensures the availability and accessibility of high-quality data to support the subsequent modules, such as the traffic simulation engine and the decision support system. The Data Management Platform itself is the result of work package WP3.

2.2 Traffic Simulation Engine

The traffic simulation engine, main part of the policy simulation and validation module, implemented using the open-source Multi-Agent Transport Simulation (MATSim) [1] software, is responsible for generating realistic and accurate traffic simulations. By leveraging the data provided by the Data Management Platform and additional static datasets, the simulation engine can create detailed models of urban mobility scenarios, enabling city decision-makers and specialists to analyse the potential impacts of different policy interventions and infrastructure changes. The traffic simulation is a micro-simulation, meaning that each vehicle is simulated, not aggregated traffic flows. There are several advantages to micro-simulations in comparison to meso- and macro-simulations, such as higher precision, ability to model heterogenous driver behaviour, ability to simulate interaction between different transport modes such as cycling and using public transport. However, there are several disadvantages to micro-simulations as well, such as requiring a lot of computing power and specific, high-quality data.

The Traffic Simulation Engine is described in detail in the deliverable D4.4 URBANITE Traffic Flow Model.

2.3 Decision Support System (DSS)

The decision support system is a crucial component of the smart urban mobility platform, providing a common simulation evaluation framework and integrating Dexi [2] for multi-criteria decision analysis. This module assists decision-makers such as traffic and urban planners and other stakeholders in comparing and evaluating various policy options based on a set of pre-defined criteria and performance indicators. By combining the output of the traffic simulation engine with the MCDA methodology, the decision support system offers valuable insights and recommendations to help city officials make informed decisions regarding urban mobility policies. The decision support system uses preferential decision models, developed for each pilot city in collaboration with the city stakeholders.

The DSS and the decision models developed are detailed in the deliverable D4.3 URBANITE Policy Decision Model. Discussion of the DSS as related to the RE follows in Section 4 Multi-Criteria Decision Analysis.

2.4 Advanced Visualizations

Advanced visualizations are an essential aspect of the smart urban mobility platform, enabling city stakeholders to better understand and interpret the results of the traffic simulations and decision analysis. The visualization component incorporates map-based visualizations, and visualizations enabling decision support, including up to 5-dimensional visualization of key performance indicators (KPIs) to represent complex data sets in an easily digestible format. These visualizations help facilitate data-driven decision-making and enhance communication among stakeholders.

The advanced visualizations module plan is outlined in the deliverable D4.1 Strategies and Algorithms for Data Modelling and Visualizations, the implementation is described in the deliverable D4.2 Implementation of Strategies and Algorithms for Data Modelling and Visualizations. The relevant visualizations are also mentioned in the deliverables D4.3 URBANITE Policy Decision Model and D4.4 URBANITE Traffic Flow Model.

2.5 Exploratory Data Analysis Module (Orange3)

The exploratory data analysis module, which was developed using the Orange3 data mining tool [3], empowers users to further analyse and investigate the simulation results. This component allows traffic and urban planners and other technical staff to conduct additional data processing, exploration, and visualization tasks, customized to their specific needs and objectives. By providing a versatile and user-friendly interface, the Orange3 module encourages in-depth analysis of urban mobility policies and fosters a more comprehensive understanding of the potential outcomes and implications of various policy proposals.

The EDA module is described in the deliverable D4.2 Implementation of Strategies and Algorithms for Data Modelling and Visualizations.

2.6 Objectives

The recommendation system developed within the URBANITE project is designed to support urban decision-makers and traffic planners in the field of urban mobility policy by offering a comprehensive, data-driven, and user-friendly platform. The system's primary objectives are described in the following sections.

2.6.1 Enable Multi-Criteria Decision Analysis (MCDA) for Selection of Best Proposal

The recommendation system facilitates MCDA, allowing users to compare various policy proposals or evaluate them against a baseline, such as the current situation. By considering multiple KPIs, which may sometimes conflict, MCDA provides a holistic approach to policy evaluation. For example, a proposal might reduce air pollutant emissions but also result in longer travel times for residents. In such cases the city's decision support model shows which solution is preferable, but also provides the detailed comparison by specific KPIs and aggregated values by KPI type (e.g., air pollutant emissions aggregate the values of specific pollutants, public transport aggregates the average speed of public transport and usage of public transport).

2.6.2 Recommend Proposals or Suggest Optimizations

The system empowers users to explore further improvements to policy proposals by leveraging further analysis of the results of MCDA. This feature encourages decision-makers to consider alternative possibilities or identify potential optimizations to enhance policy effectiveness. The effectiveness of the proposals to provide such empowerment is not obvious as the proposals are not directly actionable.

2.6.3 Support Data-Driven Decision Making

The recommendation system is built on a foundation of real data and calibrated using known traffic flow measurements or OD matrices from participating cities. This data-driven approach ensures that simulated policy proposals are grounded in reality, differing from real-world traffic simulations only in the specific aspects of the proposal (e.g., new bike infrastructure or road closures). By providing accurate and reliable information, the system promotes informed and evidence-based decision-making.

2.6.4 Enable Robust KPI Estimation

The KPIs used in the recommendation system have been carefully designed in collaboration with pilot city representatives to address specific urban mobility sub-domains relevant to each participating city [4]. These sub-domains include general mobility for Bilbao, public transport for Messina, bicycle traffic for Amsterdam, and harbour vehicle flows for Helsinki. By offering robust KPI estimation tailored to each city's unique needs, the system ensures that decision-makers, traffic and urban planners have the necessary information to make informed choices in their specific context.

3 Recommender Engine (RE)

The recommender engine is a core component of the URBANITE project's smart urban mobility platform, designed to assist decision-makers in identifying optimal urban mobility policy proposals based on their specific requirements and objectives. By analysing the input data and simulation results, the recommender engine generates suggestions on what aspect of urban mobility to focus on to further improve the scenario, ensuring that city officials have access to a diverse set of options to address their urban traffic challenges.

3.1 Use of the Recommendation System

The recommendation system has a very simple to use and easily understandable UI, shown on Figure 2 below.

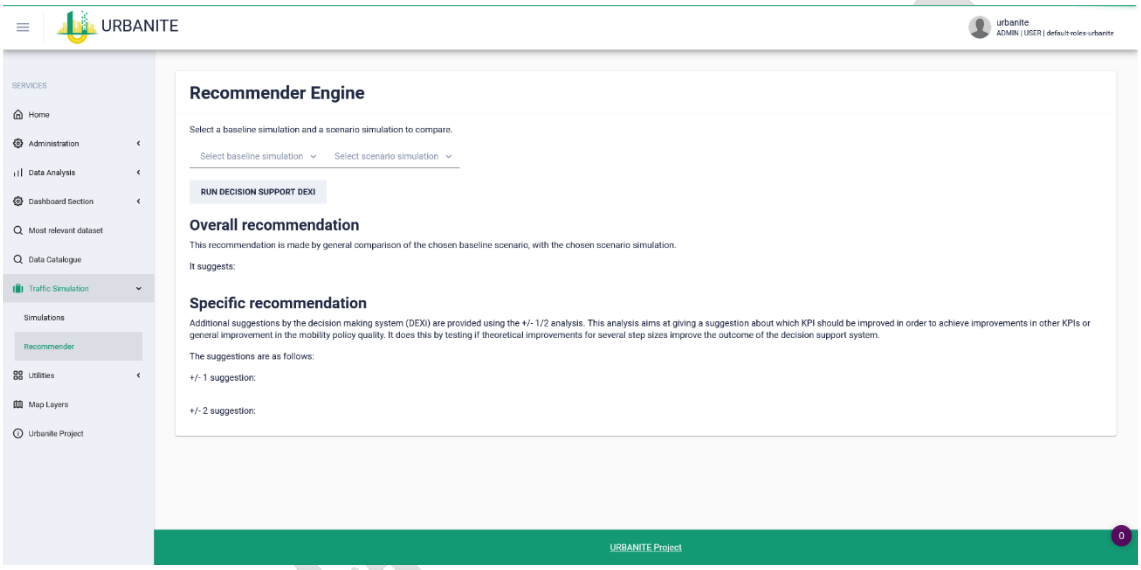

Figure 2: The recommendation system user interface. The user must select a baseline simulation and a scenario simulation from those available.

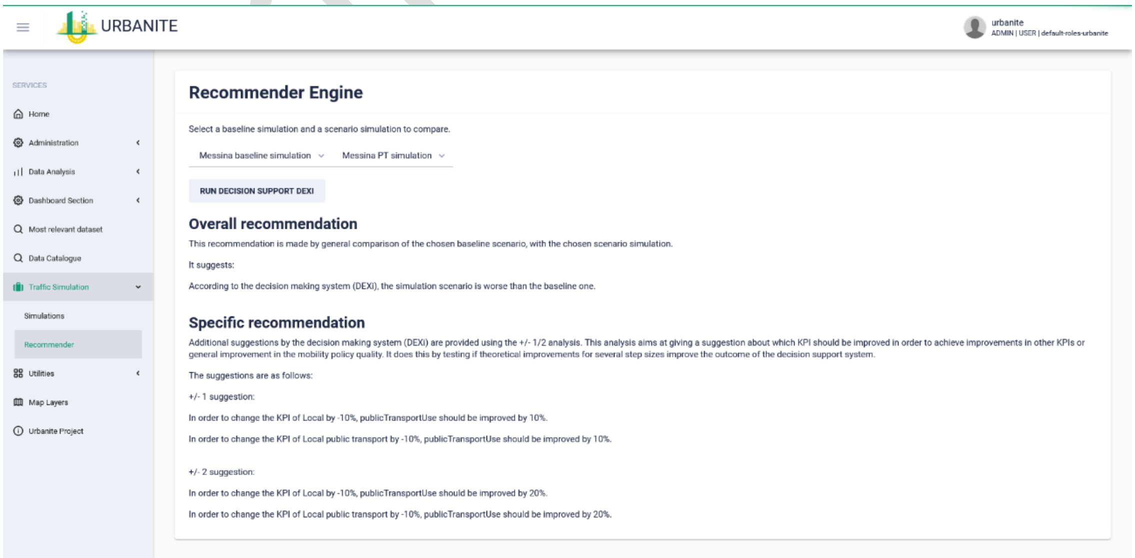

Figure 3: The UI of the recommendation system presents overall recommendation and specific recommendations.

The use of the RE is simple and requires minimal user input. First, the user selects the baseline simulation and a simulated scenario. The baseline is required to evaluate the scenario simulation and provide relative comparison of specific KPIs.

After that, the user clicks the “Run decision support Dexi” button, which causes the RE to run the analysis and return results. This process is fast, and the user does not need to wait for the results long.

The overall recommendation is an evaluation of the scenario, it tells the relative change in overall quality of the simulated scenario and provides recommendation about whether to implement it.

The specific recommendation is split into two sections, the +/-1 suggestion and the +/-2 suggestion. The first provides insight into how to improve the overall quality by slightly improving a specific aspect of the city, such as air quality, or public transportation. The second one provides the same information, but assumes larger improvements, or multiple improvements, to improve the overall quality of the scenario in a more significant way.

3.2 Plus/Minus One Analysis

The recommender engine employs a plus/minus one analysis [2] approach, which systematically explores variations of existing proposals by incrementally adjusting their parameters. This method enables the system to identify potential improvements or trade-offs in the proposal that may better align with the decision-maker's goals and preferences. By evaluating these variations, city officials can refine their proposals and achieve a more comprehensive understanding of the potential impacts of different policy options.

3.3 Algorithm and Implementation

The underlying algorithm of the recommender engine leverages machine learning techniques and optimization algorithms, to generate and evaluate proposal variations. By analysing the KPIs and other relevant metrics, the engine can identify proposals based on their performance, enabling decision-makers to focus on the most promising options. The recommender engine is integrated with the other components of the smart urban mobility platform, ensuring seamless interaction between the data platform, the traffic simulation engine, and the decision support system.

3.4 Benefits and Limitations

The recommender engine offers several benefits to urban mobility decision-makers and other stakeholders. It streamlines the proposal evaluation process, provides a set of policy options, and facilitates data-driven decision-making. By providing aspects of mobility that if improved may further improve scenario, the engine encourages innovative thinking and allows decision-makers to consider alternative solutions they may not have initially identified.

However, the recommender engine has certain limitations. The quality of its recommendations depends on the accuracy and completeness of the input data, the calibration of the traffic simulation models, and the relevance of the KPIs used for evaluation. The proposals generated are not directly actionable, and only offer possible directions for further consideration. It is essential to recognize these limitations and consider them when utilizing the recommender engine within the decision-making process.

3.5 Changes from the Initial Design

The initial design of the recommender engine, detailed in the deliverable D4.5 Recommendation System for Policy Design, underwent several changes. Aim of the changes was to ensure a faster and more robust implementation while still providing valuable insights to urban mobility decision-makers. The primary motivation for these changes was to streamline the recommendation process and reduce computational overhead, as evolutionary algorithms (EAs) [5] required considerable time and computational resources for generating and evaluating multiple solutions. After initial experiments, we have estimated that to achieve expected results, thousands of scenario evaluations are needed, with each one taking a couple of hours on available hardware [6]. The high requirements of using EAs for such optimization techniques were known at time of initial design and planning, but the expectation was that by using heuristics and appropriate crossover and mutation operation would help make the design feasible, which experiments proved to not be true. The modified design aims to deliver output within a more reasonable time, while preserving limited computational resources, also used by other components.

Reasons for changes:

- Policy Encoding Wizard

The originally proposed policy encoding wizard was significantly altered. The initial concept of the policy encoding wizard was not implemented due to its high complexity and the inability to meet the required time constraints. Instead, a more straightforward alternative was chosen. The new approach enables users to create simulation scenarios by modifying a limited number of simulation parameters related to each city's use case, limiting the diversity of possible simulations but allowing for exploration of relevant policy options within constraints.

- Simulation of Proposals and Variants

The simulation of proposals and variants was completely dropped for two reasons. The first is, this functionality as defined in the deliverable D4.5 Recommendation System for Policy Design, was not implemented due to time constraints brought on by delays in the implementation of the traffic simulation module. Similar functionality is provided by enabling the users to change previously mentioned simulation parameters, while reducing the complexity and computational demands. The second reason is the lower priority with regards to other requests from use cases validation sessions. Initial design included variations in simulated weather conditions and differences between weekday and weekend traffic patterns. While relevant source data were harvested and are available in the data management platform, we finally were unable to aggregate these into appropriate datasets due to lack of time and other priorities set by the use cases' participants.

These changes from the initial design have resulted in a more efficient and user-friendly recommender engine, while still providing valuable support for urban mobility policy decision-making, urban and traffic planning. Despite the simplification of certain aspects, the modified design maintains its core functionality and continues to facilitate data-driven decision-making for city officials. Likewise, other alternatives have been explored and implemented, exploiting metaheuristic and collaborative optimization methods to complement the recommender engine, exploiting, in the long term, the domain knowledge and criteria of mobility technicians and planners.

4 High-level Recommender Engine (RE)

This recommender complements the previous one, and it is designed to support the initial steps of the planning, gaining the advantage of previous experiences on decisions.

Based on the city's base decision model, different recommendations are provided: in the first case, given a decision model, it identifies the combination of KPIs that maximize the general indicator (Mobility Policy Quality) defined for each of the pilots. The second recommendation provides a list of actions that maximize a cost function based on the selection and weighting of indicators defined by the council technician. Finally, a third option presents a list based on previous experiences, both from the council and from the rest of the pilots.

4.1 Optimization

Through optimization, the different solutions in the spaces for two different applications are explored:

- Given a decision model, the objective is to identify the values for the independent indicators in order to optimize the overall KPI. Three improvement criteria can be applied: minimization of the cost necessary to reach a solution in the form of weighted distance. As the nodes are classified according to relative values [>15%, >5%, 0, <5%, <15%] (see section 5.1)] and having a cost estimate based on the variation, the distance is defined by Cij*KPI relative value, improvement of the global indicator, in this case, % improvement or weighting of them.
- The second area of optimization starts with a portfolio of urban solutions and their characterizations in terms of parameters and impact on each of the reference indicators in the use case. For these parameters, enumerated types are assumed that allow the exploration method responsible for creating urban solutions:
    - Closure of an area o set of streets (area id, restricted vehicles)
    - Update of public transport Line (new frequency, stops)
    - Traffic Lights Regulation (cycle)
    - Speed constraints on an area or streets (area_id, speed)
    - Tunnel Construction (ends, speeds, direction)
    - Bike Line Construction ends, itinerary)
    - Pedestrian Line Construction (ends, itinerary)

The application of this algorithm implies the following definitions:

- Structure of the candidate solution to the problem.
- How its elements are used to calculate the objective functions that serve to evaluate the quality of the current solution.

4.1.1 Algorithm and Implementation

In URBANITE project, the approach employs metaheuristic techniques for solution exploration; specifically, it is proposed to use NSGA II (Elitist Non-Dominated Sorting Genetic Algorithm), which is one of the most popular Pareto-based approaches whose basic idea is to find a set of non-dominated individuals in the population and whose general structure is shown in Figure 3.

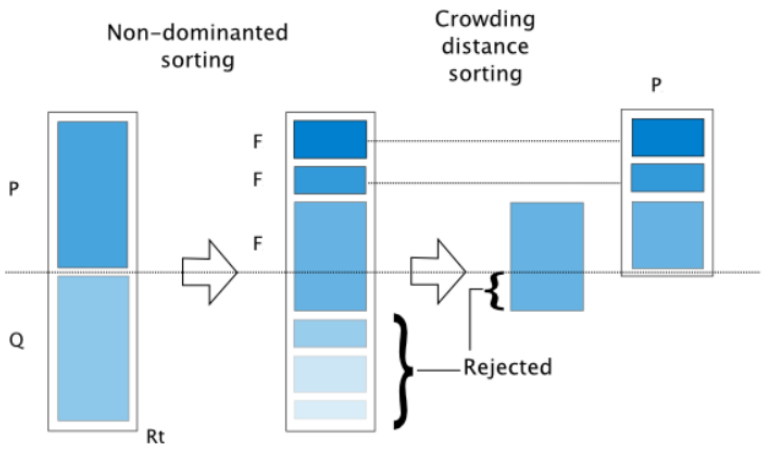

Figure 4: Logic scheme for the NSGA II algorithm (https://pymoo.org/algorithms/moo/nsga2.html)

In this case, we are going to consider a hybrid chromosome: combining a binary vector (A, where A is the number of actions) where each element identifies whether the actuation is applied or not; on the other hand, a vector of natural numbers (R, the number of potential solutions) which represents the order of the different solutions to be applied and the value of the different parameters, within the range of its definition.

4.1.2 Benefits and Limitations

As mentioned in the previous point, the exploration of the solution space requires time, which is why it is not viable in the interactive system. In particular, for the second approach, simulation-based optimization deals with developing efficient sampling schemes supported by surrogate models; however, to generate these simplified models, it would have been necessary to have a relevant volume of simulations available.

4.2 Collaborative Filtering

Collaborative filtering systems aim to predict a user's interest in items by leveraging the preferences of other users. These systems identify similar users to generate personalized recommendations, relying solely on past user-item interactions or historical preferences rather than item metadata (unlike content-based filtering). There are two classes of collaborative filtering algorithms [7]: memory-based and model-based. Memory-based algorithms employ heuristics to make predictions using the entire database. The recommendation values for an item are calculated by aggregating the records of other users who have shown interest in the same item. On the other hand, model-based algorithms construct a model from the database and make predictions based on that model. The key distinction between model-based algorithms and memory-based methods is that model-based approaches do not rely on heuristic rule, instead, utilize learned models to provide recommendations.

URBANITE adopts a memory-based collaborative filtering approach using two methods: user-based and item-based. User-based filtering recommends items based on the similarity between users, while item-based filtering suggests new items based on their similarity to previously selected ones.

The following search criteria have been taken into account:

- k-NN recommendation based on a similarity item-to-item metrics, the recommendation will provide those solutions closest to the previous selection.

- Based on preferences, either from the selection of the objective values of a series of attributes or from their weighting (from -50 to +50).
- Bayesian approach [8], able to directly calculate the probability of the technician's possible interests without defining a similarity or distance metric, only based on the conditional probability of a solution being a good option for the user whose previous selections are known.

4.2.1 Algorithm and Implementation

The different filters have been implemented as stored procedures of the database.

4.2.2 Benefits and Limitations

Again, the idea of this approach is to avoid dependence on a large volume of data for the training and adjustment of analytical models, in which case, the quality of predictions is solely dependent on the quality of the model built. However, as main disadvantages, we find that a Memory-based filtering recommender does not scale well, taking a while to find similarities; for huge datasets, this technique is close to being impractical and also has the "cold-start" problem, it's hard to get recommendations based on new criteria.

5 Multi-Criteria Decision Analysis

Multi-Criteria Decision Analysis (MCDA) [9] is a widely used decision-making methodology that allows decision-makers and other stakeholders to evaluate and compare different policy options based on multiple criteria or key performance indicators (KPIs). In the context of urban mobility policy design, MCDA provides a holistic approach to evaluating proposed interventions that may have a significant impact on various aspects of urban mobility, such as transportation equity, air quality, or accessibility.

This section of the deliverable focuses on the MCDA methodology and its integration within the URBANITE project's recommendation system. We will discuss the fundamental principles and techniques used in MCDA, such as preference modelling, weighting, and aggregation, as well as the challenges and limitations of applying MCDA to urban mobility policy design. We will also explore the integration of MCDA within the recommendation system and its role in providing valuable insights and recommendations to decision-makers.

5.1 Multi attribute models

For each partner city, a decision model is created which has various attributes and rule sets. Some of the attributes are discretized KPIs values which are explained in a later section. The other attributes are subjective and represent an aggregated decision defined by the policy maker. The scale of the discretized KPI attributes is [>15%, >5%, 0, <5%, <15%], meaning, how different is an attribute between two simulations. The order (increasing/decreasing) of the values in this range is important.

The scale of the subjective aggregated attributes is [Better, Worse, Same]. A value from this subjective range is obtained according to previously mentioned, defined rule sets.

5.2 KPIs

As part of the URBANITE project, specific KPIs were designed in collaboration with the pilot city representatives to address the unique urban mobility sub-domains of each participating city. Following is the overview of designed KPIs for each city.

5.2.1 General

5.2.1.1 Acoustic pollution

A specific component has been deployed related to the car traffic works for the four use cases considered within the project, that allows to compute the noise produced by the cars in a given area (it is not possible to calculate for the whole medium cities in a single step). The component has two different modes of operation: the first, takes a set of roads and produces estimates for what is the most likely flux of vehicles at each of the links of the navigational map considering only static characteristics of the road, i.e., number of lanes, size, type of the road, etc, from those flux estimates a typical noise pattern within the city is computed. The second mode of operation uses a Matsim [1] traffic simulation output as the input for the noise computation. This output is more flexible being able to characterize a wider class of situations and based to the actual simulated traffic. The module is constructed on top of Noise Modelling[1], which is a library developed by acousticians and GIS (Geographic Information Science) and that can be freely used for research, education or professional use. The results and visualization are given in baud rate (bdms).

[1] Noise-Planet -Scientific tools for environmental noise assessment (https://noise-planet.org)

Both, an API (described in the 9 APPENDIX: Noise Computation API) and a user interface, whose use was described in D5.9, are provided for its integration with other components of the platform.

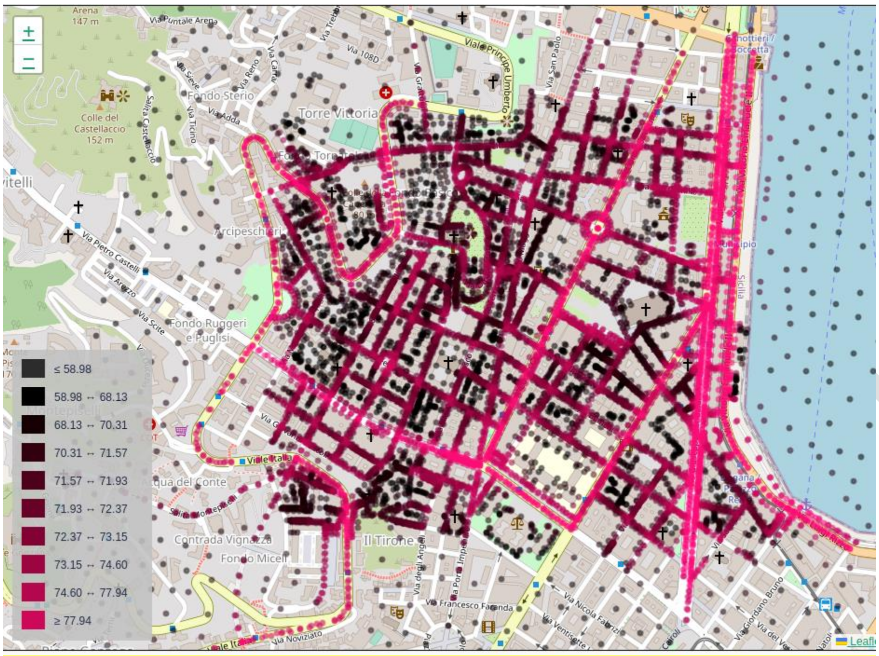

Figure 5: Detail of the resulting noise for the use case of Messina considering the static characteristics of the roads. Different colours denote different dBm of intensity at 63Hz.

5.2.2 Amsterdam KPIs

This chapter describes the KPIs supporting the Amsterdam use case. A visualization of the values of KPI Bike Congestions is shown on Figure 6.

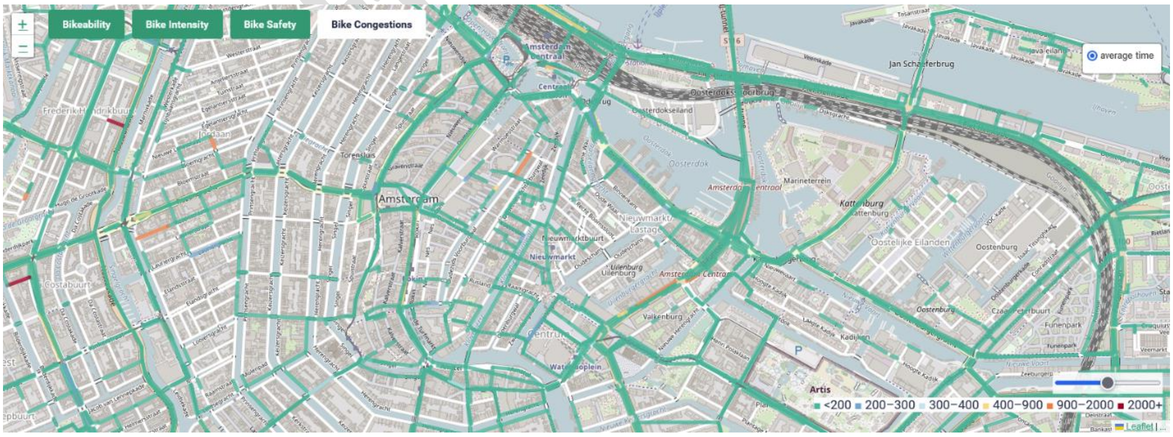

Figure 6: Visualization of values of the Bike Congestion KPI in center of Amsterdam.

5.2.2.1 Bike Infrastructure

The KPI for bike infrastructure measures the extent and quality of the infrastructure available to support bicycle transportation. This includes factors such as the number of bike lanes, bike parking facilities, and the quality of road surfaces.

Each road segment in the city's road network is assigned a score based on the data, available in the simulation. The scores partially follow the Munich Bikeability Index [10] infrastructure score. The infrastructure scores are reassigned based on the MBI values but relegated to street types available on OSM. The score for different types of road infrastructure and speed limits are listed in Table 1.

Table 1: Score values for types of road segments as tagged on OSM.

Road Segment OSM Highway Tag	Value	Comment
Cycleway	10	
Path	7	
Living street	7	Pedestrians have legal priority over cars and other vehicles.
Pedestrian	5	
Footway	5	Only those where cycling is allowed considered.
Unclassified	5	Least important roads, often link villages and hamlets. Does not mean type unknown.
Track	5	Roads for agricultural or forestry uses.
Service	4	
Residential	4	
Tertiary	3	
Secondary	2	
Primary	1	
Motorway	0	Cycling not allowed.
Steps	0	Cycling very dangerous for cyclists and pedestrians.

5.2.2.2 Bike Speed Limit

The KPI for bike speed limit refers to the maximum speed limit for bicycles on specific roads or bike lanes. This KPI is important for ensuring the safety of cyclists and other road users and promoting sustainable mobility by encouraging more people to cycle. This KPI follows the MBI [10] speed score.

Table 2: Score values for different speed limits on road segments.

Speed Limit of Road Segment	Value	Comment
≤30 km/h	10	
≤50 km/h	7	Accidents mostly cause minor injuries.
>50 km/h	0	Accidents may often cause major injuries or death.

5.2.2.3 Bikeability

The KPI for bikeability is a comprehensive metric that assesses the overall quality of the cycling environment. This KPI is combined from bike speed limit KPI and bike infrastructure KPI. This KPI follows the MBI [10] but is changed to be fit for evaluating simulated data (infrastructure scores are based on OSM tags instead of custom values, removed bicycle parking due to no data availability in simulations and intersection scoring due limited traffic light data availability in the simulations).

The bikeability score is the average of bike infrastructure KPI values and bike speed limit KPI values.

5.2.2.4 Bike Intensity

The KPI for bike intensity measures the volume of bike traffic on a specific road or bike lane. This KPI is essential for understanding the usage and popularity of cycling as a mode of transportation and can help identify areas where improvements are needed to support increased bike traffic.

Bike intensity is implemented as the number of bicycles using a specific road segment during the simulated day.

5.2.2.5 Bike Congestion

The KPI for bike congestion measures the level of traffic congestion experienced by cyclists on specific roads or bike lanes. This KPI is important for understanding the quality of the cycling experience and identifying areas where infrastructure improvements or traffic management strategies may be necessary to reduce congestion and improve safety for cyclists.

Bike congestion is implemented as the combined length of all bikeable road segments. For each road segment it is determined whether it is congested based on calculated traffic flow values of cyclists.

5.2.3 Bilbao KPIs

This chapter describes the KPIs developed for the Bilbao use case. Shown on Figure 7 is the visualization of amounts of emitted CO2 in the simulated day.

Figure 7: Visualization of emitted CO2 in the center of Bilbao. Amounts of different air pollutants emitted can be shown.

5.2.3.1 Share of bikes

This KPI measures the proportion of trips made by bicycle. It provides insights into the prevalence and effectiveness of cycling as a mode of transportation, which can have significant impacts on urban mobility, air quality, and public health.

The share of bikes is calculated as the ratio between the number of bicycles and the number of all vehicles on a road segment throughout the simulated day.

5.2.3.2 *Share of cars*

This KPI measures the proportion of trips made by cars. It provides insights into the prevalence and effectiveness of car use as a mode of transportation, which can have significant impacts on urban mobility, air quality, and congestion.

The share of cars is calculated as the ratio between the number of cars and the number of all vehicles on a road segment throughout the simulated day.

5.2.3.3 *Share of public transport*

This KPI measures the proportion of trips made by public transport vehicles, such as buses, trains, and trams. It provides insights into the prevalence and effectiveness of public transport as a mode of transportation, which can have significant impacts on urban mobility, accessibility, and air quality.

The share of public transport is calculated as the ratio between the number of public transport vehicles and the number of all vehicles on a road segment throughout the simulated day.

5.2.3.4 *Acoustic pollution*

This KPI measures the level of noise pollution in a selected area, which can have significant impacts on public health, quality of life, and urban mobility. Elevated levels of noise pollution can contribute to stress, sleep disturbance, and hearing loss.

The acoustic pollution KPI considers the simulated traffic situations and the geometry of buildings and other large objects in the city.

5.2.3.5 *CO2, PM10, NOx*

These KPIs measure the levels of carbon dioxide, particulate matter, and nitrogen oxides, which can have significant impacts on air quality, public health, and climate change. Elevated levels of these pollutants can contribute to respiratory problems, cardiovascular disease, and other health issues.

The air pollutant emissions KPIs consider simulated traffic situations and are based on the measured emission factors and vehicle fleet composition as provided by The Handbook of Emissions Factors (HBEFA).

5.2.3.6 *Average pedestrian trip time*

This KPI measures the average time it takes for pedestrians to complete a trip. It provides insights into the accessibility and quality of the pedestrian infrastructure, which can have significant impacts on urban mobility, safety, and public health.

Note that the map-based visualization shows only long pedestrian trips, which can represent a lack of alternative mobility options.

5.2.4 Helsinki KPIs

This chapter describes the KPIs developed for the Helsinki use case. Figure 8 shows the visualization of the Congestions and bottlenecks KPI on the map.

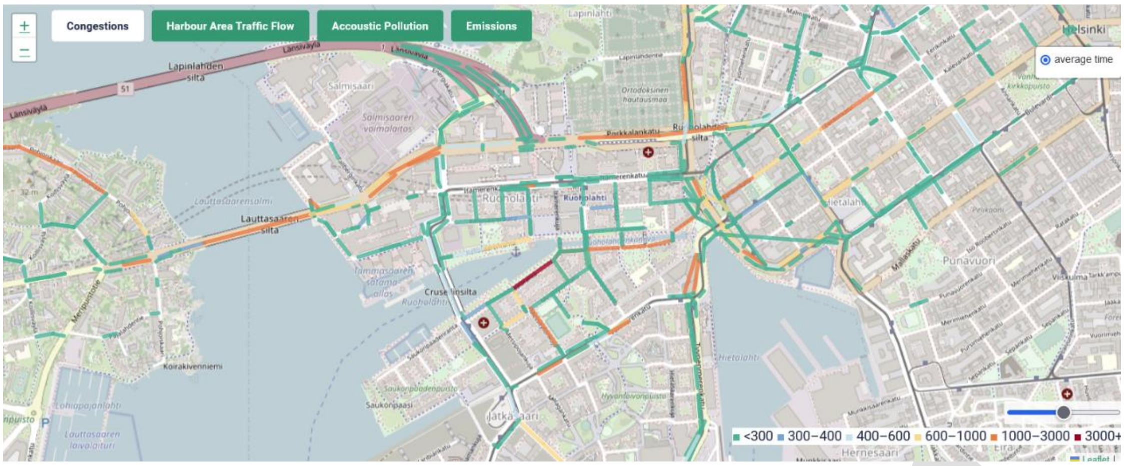

Figure 8: Visualization of congestions and bottlenecks KPI in Helsinki, coloured by average time required to pass a road segment during the hour of highest congestion in seconds.

5.2.4.1 CO2, PM10, NOx

These KPIs measure the levels of carbon dioxide, particulate matter, and nitrogen oxides, which can have significant impacts on air quality, public health, and climate change. Elevated levels of these pollutants can contribute to respiratory problems, cardiovascular disease, and other health issues.

The air pollutant emissions KPIs consider simulated traffic situations and are based on the measured emission factors and vehicle fleet composition as provided by The Handbook of Emissions Factors (HBEFA).

5.2.4.2 Acoustic pollution

This KPI measures the level of noise pollution in a selected area, which can have significant impacts on public health, quality of life, and urban mobility. Elevated levels of noise pollution can contribute to stress, sleep disturbance, and hearing loss.

The acoustic pollution KPI considers the simulated traffic situations and the geometry of buildings and other large objects in the city.

5.2.4.3 Congestions and bottlenecks

Congestions and bottlenecks are key performance indicators that help evaluate the efficiency of the urban mobility system. High levels of congestion result in increased travel times, decreased accessibility, and reduced economic productivity. By monitoring and analysing the levels of congestion and bottlenecks, decision-makers can identify areas in which traffic management interventions, such as lane restrictions or public transportation improvements, may be necessary.

Congestions and bottlenecks KPI is implemented as the combined length of all congested road segments. For each road segment it is determined whether it is congested based on calculated traffic flow values.

5.2.4.4 Harbour area traffic flow

Harbour area traffic flow is a critical KPI in evaluating the efficiency of cargo transportation in urban areas. High traffic flows indicate flowing traffic and good conditions, but low traffic flow may mean either a congestion or low utilisation of the roads. By monitoring and analysing the traffic flow in the harbour area and considering also the Congestions and bottlenecks KPI, decision-makers can identify areas where improvements to cargo transportation may be necessary.

This KPI is implemented by adding virtual traffic flow sensors to simulations at the locations of existing real sensors, focused on the Jätkäsaari Smart Junction in Helsinki.

5.2.5 Messina KPIs

This chapter describes the KPIs developed for the Messina use case. Figure 9 shows the visualization of public transport usage KPI.

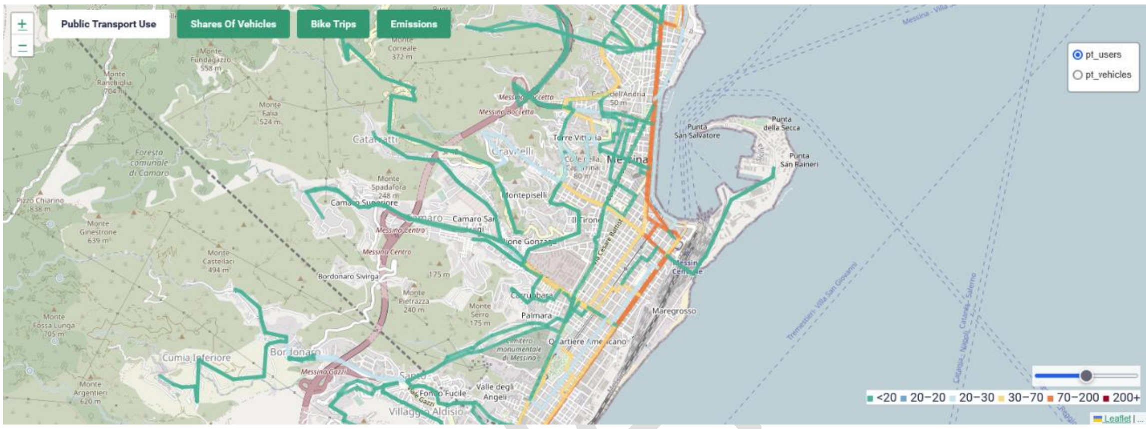

Figure 9: Visualization of the number of daily passengers using public transport in Messina.

5.2.5.1 Public transport usage

This KPI measures the number of passengers using public transport services in a specific period. It is a critical metric for urban mobility decision-makers, traffic planners and researchers as it provides insights into the demand for public transport services and helps identify potential opportunities to improve service quality and coverage to meet the needs of the public.

Public transport usage KPI is implemented as the number of passengers of all public transport vehicles and visualized as the number of passengers of all public transport vehicles on specific road segment.

5.2.5.2 Average speed of public transport

This KPI represents the average speed of public transport vehicles. It provides insight into the efficiency and reliability of public transport services, as well as the effectiveness of traffic management policies. Improving the average speed of public transport can reduce travel time and encourage more people to use public transport.

The average speed of public transport is calculated using the following algorithm:

1. The speed is calculated for each public transport vehicle traveling each road segments.
2. The speeds are averaged for each road segment. This data is used to visualize the average public transport speed on the map, enabling the identification of road segments that cause largest delays.
3. The speeds are aggregated for all road segments, weighted by length of road segment. This is the final value of the KPI.

5.2.5.3 Number of bike trips

This KPI measures the number of trips made by bicycles throughout the simulated day. It is a critical metric for urban mobility decision-makers who aim to promote sustainable and healthy transportation alternatives. By encouraging more people to use bicycles, cities can reduce traffic congestion, improve air quality, and promote physical activity.

Number of bike trips is implemented by counting all bike trips made in the city. Note that this KPI is visualized on the map-based visualization as the number of bike trips that use a specific road segment.

5.2.5.4 Share of public transport

This KPI measures the proportion of trips made by public transport vehicles, such as buses, trains, and trams. It provides insights into the prevalence and effectiveness of public transport as a mode of transportation, which can have significant impacts on urban mobility, accessibility, and air quality.

The share of public transport is calculated as the ratio between the number of public transport vehicles and the number of all vehicles on a road segment throughout the simulated day.

5.2.5.5 Share of cars

This KPI measures the proportion of trips made by cars. It provides insights into the prevalence and effectiveness of car use as a mode of transportation, which can have significant impacts on urban mobility, air quality, and congestion.

The share of cars is calculated as the ratio between the number of cars and the number of all vehicles on a road segment throughout the simulated day.

5.2.5.6 Share of bikes

This KPI measures the proportion of trips made by bicycle. It provides insights into the prevalence and effectiveness of cycling as a mode of transportation, which can have significant impacts on urban mobility, air quality, and public health.

The share of bikes is calculated as the ratio between the number of bicycles and the number of all vehicles on a road segment throughout the simulated day.

5.3 Preparing input to multi criteria decision analysis

Multi criteria decision analysis, performed using the decision support tool DEXi [9], is the main part of the recommender engine. To perform multi criteria decision analysis, we need to calculate some relative values between two simulations. We obtain these values from the corresponding KPIs. Firstly, relative values are calculated for each KPI between a baseline simulation with no changes in the network, and a scenario simulation that has some changes. This calculated number indicates percentage amount of change. For example, KPI 1 in the scenario simulation is 7% higher than the same KPI in the baseline simulation.

After obtaining the relative change, we discretize it into indexes ranging from 1 to 5. The indexes correspond to the ranges in the scale of discretized attributes. For example: If a KPI has a relative change of 21%, then it is discretized to index 1 (changes larger than 15%). If it has a relative change of 11%, then it is discretized to index 2 (changes larger than 5%), and so on. When the relative change decreases, the index number increases and vice versa. This is true for all KPIs. This is the reason the order of the scale is important. A KPI that indicates a beneficial occurrence, like bike safety, will have a decreasing scale, while a KPI that indicates a detrimental occurrence has an increasing scale.

6 Source Code

The source code is available in Tecnalia's Gitlab:

- DSS repository: https://git.code.tecnalia.com/urbanite/private/wp4-algorithms-and-simulation/dss
- High-level recommender engine: https://git.code.tecnalia.com/urbanite/private/wp4-algorithms-and-simulation/recommendation-engine-alternative
- The front end is part of the URBANITE UI and available at: https://git.code.tecnalia.com/urbanite/private/wp5-integration-and-devops/urbanite-ui-template/
    - Recommender engine in folder '/app/pages/simulation-wizard/recommender-page',
    - imports some other files from the simulation-wizard folder '/app/pages/simulation-wizard'.

7 Conclusions

The recommender engine is developed on top of the other URBANITE modules. It utilises the traffic simulation engine, powered by the data gathered in the data platform developed in work package WP3, and the decision support system, built on top of the traffic simulation engine. The workings of the recommendation system are explainable and there are no black box algorithms required, which may help increase the trust in technology among the stakeholders, which is among the objectives of the project.

The main feature of the recommender engine is to provide insight into which aspect of urban mobility should be focused on to further improve the simulated scenario. The recommender engine developed deviates from initial design, since it was discovered that the initial design was not feasible for any real-world usage due to large drain on resources and requirement of extremely high computational power. The final implementation has several advantages over the initial design, as well as some drawbacks. The advantages over the initial design are lower requirements on computational power, leading to faster results, the provided information is simple to understand, explainable, and transparent, while still being useful to the decision makers, and simplified user interface and usage of the tool. On the other hand, the results are less detailed and do not provide directly actionable information, as was initially planned.

The recommender engine is based on the decision support system developed for the URBANITE project and utilises MCDA and +/-1 analysis, expanded to +/-2 analysis.

Besides, the high-level recommender complements the previous one, and is designed to support the initial steps of the planning, gaining the advantage of previous experiences on decisions. It provides three types of recommendations are provided: in the first case, given a decision model, it identifies the combination of KPIs that maximize the general indicator (Mobility Policy Quality) defined for each of the pilots. The second recommendation provides a list of actions that maximize a cost function based on the selection and weighting of indicators defined by the council technician. Finally, a third option presents a list based on previous experiences, both from the council and from the rest of the pilots.

In the future, this method of providing recommendation can be further improved to consider more simulations at once, improvements of the decision support system, and improvements in the quality of the traffic simulations. Further research is required to develop a recommendation system that will provide more actionable results. Plans for doing this include research into intelligent neural agents, that may allow approach using genetic algorithm by providing faster conversion, and building surrogate models that will allow for higher number of evaluations of proposed scenarios faster than full traffic simulations.

8 References

[1] A. Horni, K. Nagel y K. W. Axhausen, The Multi-Agent Transportation Simulation MATSim, London: Ubiquity Press, 2016.

[2] M. Bohanec, «DEXi Suite: Renewing Qualitative Multi-Criteria Decision Modeling,» de *Proceedings of the ICDSST 2023*, Albi, 2023.

[3] J. Demšar, T. Curk, A. Erjavec, Č. Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajner, M. Toplak, A. Starič, M. Štajdohar, L. Umek, Žagar Lan, J. Žbontar, M. Žitnik y B. Zupan, «Orange: Data Mining Toolbox in Python,» *Journal of Machine Learning Research,* vol. 14, pp. 2349-2353, 2013.

[4] J. Al Dakheel, C. Del Pero, N. Aste y F. Leonforte, «Smart buildings features and key performance indicators: A review,» *Sustainable Cities and Society,* vol. 61, 2020.

[5] I. Delgado-Enales, P. Molina-Costa, E. Osaba, S. Urra-Uriarte y J. Del Ser, «Improving the Urban Accessibility of Older Pedestrians using Multi-objective Optimization,» de *2022 IEEE Congress on Evolutionary Computation (CEC)*, Padua, 2022.

[6] C. Zhuge, M. Bitchell, C. Shao, X. Li y J. Gao, «An improvement in MATSim computing time for large-scale travel behaviour microsimulation,» *Transportation,* nº 48, pp. 193-214, 2021.

[7] D. Valcarce, A. Landin, J. Parapar y Á. Barreiro, «Collaborative filtering embeddings for memory-based recommender systems,» *Engineering Applications of Artificial Intelligence,* vol. Volume 85, pp. 347-356, 2019.

[8] K. Wang y Y. Tan, «A New Collaborative Filtering Recommendation,» de *ICSI 2011, Part II, LNCS 6729*, Berlin Heidelberg, Springer-Verlag, 2011, pp. 218-227.

[9] M. Bohanec y A. J. Kulkarni, «DEX (Decision EXpert): A Qualitative Hierarchical Multi-criteria Method,» de *Multiple Criteria Decision Making. Studies in Systems*, Singapore, Springer, 2022.

[10] J. Schmid-Querg, A. Keler y G. Grigoropoulos, «The Munich Bikeability Index: A Practical Approach for Measuring Urban Bikeability,» *Sustainability,* vol. 13, nº 1, p. 428, 5 January 2021.

9 APPENDIX: Noise Computation API

##
#
DEFINITION OF THE NOISE COMPUTATION API:
In order to perform a noise computation a traffic simulation and
a map within the network input for the traffic simulation is needed.
#
##
#

##
#
createNewMap: This method allows to create a new map for the noise computation
A unique name should be passed and a bounding box defining the area.
##
#
TYPE: PUT
URL:
https://bilbao.urbanite.esilab.org/urbanite_noise_computation/webresources/putCommands/createNewMap
PAYLOAD:
{
"name": "abando",
"bbox": [
-2.942697,
43.260752,
-2.93596,
43.265285
]
}
##
#
getMaps: Method in order to get the maps that are available in the platform.
In the response, the bounding box (bbox) and the name are returned, the rest of the information include the the size and the
name of the osm file within the platform and an identification value.
##
#
TYPE: GET
URL:
https://bilbao.urbanite.esilab.org/urbanite_noise_computation/webresources/getData/getMaps
RESPONSE:
[{"loc":"moyua.osm","size":1195829,"bbox":"[-2.937848,43.260721,-2.931326,43.264379]","name":"moyua","id":1},{"loc":"abando.osm","size":1329304,"bbox":"[-2.942697,43.260752,-2.93596,43.265285]","name":"abando","id":2}]

##
#

```
# getTraffics: Method in order to get the traffic simulations that are available in the platform.
# In the response, an array of jsonObjects with the name (for the simulation) and identification id is returned
#
#################################################################################
#
TYPE:    GET
URL:
         https://bilbao.urbanite.esilab.org/urbanite_noise_computation/webresources/getData/getTraffics
RESPONSE:
[{"name":"bilbao","id":1}]

#################################################################################
#
# getModels: Method in order to get the noise simulations that have been run in and stored in the platform.
# In the response, an array of json objects is returned, each json object contains
#     "geojson": escaped version of the geojson defining the zones and the level of noise of each of the zones.
#          Each Feature is of type "Polygon"  has the following "properties"
#               "PK":      52,
#               "ISOLVL":   4,   <-- Integer correlated to the Noise Level (del 0 al 10)
#               "CELL_ID":  0,
#               "ISOLABEL": "50-55" <-- Denotes the value of NOISE
#          The mapping from ISOLVL to ISOLABEL is the following:
#                0: < "35", 1: "35-40", 2: "40-45", ... , 10: ">80"
#
#     "mapa": the name of the osm file
#     "name": name of the simulation performed
#     "id": identification value.
#     "traffic": name for the traffic simulation used.
#     "status": value which denotes the status of the noise computation, 0 means  it has finished, <0 otherwise.
#################################################################################
#
TYPE:    GET
URL:
         https://bilbao.urbanite.esilab.org/urbanite_noise_computation/webresources/getData/getModels
RESPONSE:
  [{
    "geojson": GEOJSON,
    "mapa": "moyua.osm",
    "name": "sim2",
    "id": 2,
    "traffic": "bilbao",
    "status": 0
  }],
```

##
#
createNewTraffic: Method to upload a file
The File should be a zip file that contains the following:
#
##
#
TYPE: POST
URL:
https://bilbao.urbanite.esilab.org/urbanite_noise_computation/webresources/putCommands/createNewTraffic?filename=bilbao.zip
PAYLOAD:(Query String Parameters) bilbao.zip

The zip file should contain the following 2 files with these exact names that are the output from the simulation:

- output_network.xml.gz
- output_events.xml.gz

EXAMPLE WITH CURL:

curl -X POST -H "Content-Type: application/zip" -F "thumbnail=<PATH_TO_ZIP_FILE_IN_LOCALHOST>" https://bilbao.urbanite.esilab.org/urbanite_noise_computation/webresources/putCommands/createNewTraffic?filename=<NAME_OF_ZIP_FILE>.zip

##
#
createModel:
##
#
TYPE: PUT
URL:
https://bilbao.urbanite.esilab.org/urbanite_noise_computation/webresources/putCommands/createModel
PAYLOAD:
{
 "name": "sim1",
 "map": "moyua.osm",
 "traffic": "bilbao"
}