EXPLORATORY RESEARCH

sesar JOINT UNDERTAKING

TRANSIT Modelling and Simulation Framework

Deliverable ID:	D5.1
Dissemination Level:	PU
Project Acronym:	TRANSIT
Grant:	893209
Call:	H2020-SESAR-2019-2
Topic:	SESAR-ER4-10-2019
Consortium Coordinator:	Nommon
Edition date:	30 August 2022
Edition:	v02.00.00
Template Edition:	02.00.05

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Authoring & Approval

Authors of the document

Name / Beneficiary	Position / Title	Date
Clarissa Livingston (ETHZ)	Project Team	30/08/2022
Stefano Penazzi (ETHZ)	Project Team	30/08/2022
Geoffrey Scozzaro (ENAC)	Project Team	30/08/2022
Clara Buire (ENAC)	Project Team	30/08/2022

Reviewers internal to the project

Name / Beneficiary	Position / Title	Date
Ricardo Herranz (Nommon)	Deputy Project Manager	30/08/2022
Oliva García Cantú (Nommon)	Project Team	30/08/2022
Jerónimo Bueno (Nommon)	Project Team	30/08/2022
Marta García Rodríguez-Conde (AENA)	Project Team	30/08/2022
Nadine Pilon (EUROCONTROL)	Project Team	30/08/2022

Approved for submission to the SJU By - Representatives of all beneficiaries involved in the project

Name / Beneficiary	Position / Title	Date
Ricardo Herranz (Nommon)	Project Coordinator	30/08/2022
Clarissa Livingston (ETHZ)	Project Team	30/08/2022
Clara Buire (ENAC)	Project Team	30/08/2022
Marta García Rodríguez-Conde (AENA)	Project Team	30/08/2022
Nadine Pilon (EUROCONTROL)	Project Team	30/08/2022

Document History

Edition	Date	Status	Name / Beneficiary	Justification
00.00.01	21/12/2021	Draft	Clarissa Livingston	Initial draft
00.01.00	19/01/2022	Submitted to the SJU for approval	Clarissa Livingston	Internal review before submission to the SJU
01.00.00	02/06/2022	Submitted to the SJU for approval	Clarissa Livingston	Reviewed version addressing the SJU comments
02.00.00	30/08/2022	Approved by the SJU	Clarissa Livingston	Use of SESAR 3 template and fixing of minor issues in accordance with SJU comments

EUROPEAN PARTNERSHIP

Co-funded by the European Union

TRANSIT

TRAVEL INFORMATION MANAGEMENT FOR SEAMLESS INTERMODAL TRANSPORT

This deliverable is part of a project that has received funding from the SESAR Joint Undertaking under grant agreement No 893209 under European Union's Horizon 2020 research and innovation programme.

Abstract

This document describes the transport modelling and optimisation tools developed as part of the TRANSIT project, including links to the relevant code repositories and associated technical documentation. Taking as a basis the agent-based transport modelling frameworks C-TAP and MATSim, the project has developed the following enhancements: (i) the C-TAP model of continuous long-distance travel has been updated to render it capable of dealing with multimodal transport networks in an efficient manner; (ii) the MATSim framework for the simulation of daily travel has been enhanced with the ability to model passenger reactions to multimodal passenger information systems; (iii) new optimisation algorithms have been developed to implement two intermodal solutions: the AMAN/DMAN Ground Tool (ADGT), a tactical coordination mechanism that is able to generate a new flight schedule when a disruption on the access mode to the airport occurs, and the Timetable Synchronisation Tool, a strategic coordination tool that synchronises ground and air transport timetables to offer a more robust multimodal schedule. In the next stage of the project, the newly developed framework will be used to implement long distance travel models of Spain and France and evaluate the proposed intermodal solutions and information services in a variety of scenarios

Table of Contents

List of Tables

EUROPEAN PARTNERSHIP

Co-funded by the European Union

List of Figures

Executive summary

This document describes the work completed in Work Package 5 "TRANSIT Modelling and Simulation Framework" of the TRANSIT (Travel Information Management for Seamless Intermodal Transport) project. This modelling and simulation framework consists of enhancements made to the agent-based transport simulation MATSim, the development of the agent-based long-distance travel demand modelling framework J-TAP, as well as the development of two new optimisation tools, the AMAN/DMAN Ground Tool (ADGT) and the Timetable Synchronisation Tool. Before describing these tools, it is useful to introduce the goals of the TRANSIT project as a whole.

The TRANSIT project is focused on challenges currently facing the planning, management, and operation of long-distance passenger travel services and infrastructure within Europe. The goal of TRANSIT is to develop a set of multimodal key performance indicators (KPIs), mobility data analysis methods, and transport simulation tools which together will allow the evaluation of the impact of innovative intermodal transport solutions on the quality, efficiency and resilience of the door-to-door passenger journey.

In Work Package 2, "ATM Role in Intermodal Transport: Opportunities for Innovative Intermodal Concepts and Passenger", the project identified two main types of intermodal concepts based on information sharing and coordinated decision-making between air transport and other transport modes. These are (i) mechanisms relevant for strategic level planning, including integrated ticketing and air-rail schedule coordination and (ii) mechanisms relevant for tactical decision-making, focused on providing a response plan in the case of a transportation disruption. To simulate and evaluate both strategic and tactical concepts and solutions for long-distance, intermodal passenger transport, Work Package 5 of TRANSIT has developed several transport modelling, simulation, and optimisation tools. The basis for these tools were the two existing modelling tools, MATSim and C-TAP.

For evaluating strategic level planning concepts, such as air-rail schedule coordination, the project has developed a framework for modelling long-distance travel demand. This framework is called **J-TAP**. J-TAP is a modelling framework that contains the C-TAP travel demand model, which has been reprogrammed to allow for multimodal trips, for easier use and for better modularity. J-TAP also adds many features and supporting tools that make using C-TAP more practical. J-TAP focuses on modelling how people make decisions on where to go, when, for how long, and how to get there. One of the more notable features of J-TAP is an innovative way to store a multimodal transport network that allows the exploration of the network with simple routing algorithms, while considering transfer constraints. J-TAP also includes tools to help modellers build their scenarios, store and manipulate their networks, their synthetic populations, and use custom behavioural models. J-TAP can be used to evaluate the potential effects of strategic decisions on long-distance travel demand and long-distance transport networks. Examples of such strategic decisions which could be evaluated with J-TAP are, for instance: the coordination of air and rail service schedules to produce more attractive transfer times, the construction of a new high-speed rail station at an airport terminal, or the design of an intermodal ticketing scheme. The primary output of J-TAP are the optimised long-term (months to a year) activity schedules for each agent. These schedules can be used to derive other useful outputs, such as network loading (how many trips are performed on a certain road segment, with a certain air connection, etc.).

For evaluating tactical decision-making tools, concepts or solutions, the existing agent-based simulation MATSim was enhanced. MATSim simulates the movements of synthetic persons on a city- or region-wide transport network as they perform their daily activities. It includes a mesoscopic traffic simulation and thus explicitly allows the synthetic persons to interact on the network, endogenously producing congestion and other phenomena. In the context of TRANSIT, new functionalities have been developed to model passenger reactions to both unplanned disruptions in the urban ground transport network and to multimodal passenger information systems. The output of simulations run with this enhanced version of MATSim can provide a "sandbox" to test various tactical decision-making strategies or tools for supporting such decisions, such as the AMAN/DMAN Ground Tool (ADGT). The enhanced version of MATSim could also be used to test the effects of various passenger information technologies and distribution strategies or the effects of disruption mitigation measures, such as responsive rail-replacement bus services. The main MATSim output is the detailed (second-by- second) activity plans for each agent, which include the modes and routes each agent chose between activities. From these activity plans, important metrics such as modal split can be derived. Another important output of MATSim is the "events file", which contains each "event" produced by the simulation. Such events include, for instance: when an agent enters a transit vehicle, when an agent enters or leaves a roadway segment, when an agent begins an activity, when an agent is allowed to make changes to their travel plans (e.g., when they receive information about a disruption), etc. The events file can be used to generate other useful outputs, such as network loading, passenger arrival distributions, or even videos of agent movements through the city or region over the course of a day.

In addition, the project has developed two optimisation tools that allow the simulation of important aspects of the intermodal concepts:

- On a strategic level, a so-called **Timetable Synchronisation Tool** has been developed. This is a coordination tool that can be used to synchronise ground and air transportation timetables to offer a robust multimodal schedule. Considering independent schedules, this tool assigns slight changes to each schedule by targeting optimal transfer times between modes from a passenger perspective.
- On a tactical level, an **AMAN/DMAN Ground Tool (ADGT)** has been developed. ADGT is a coordination mechanism that is able to generate a new flight schedule when a disruption on the access mode to the airport occurs, with aim of reducing the number of passengers stranded at the airport.

In the next stage of the project, the newly developed J-TAP framework will be used to implement long-distance travel models of Spain and France and the enhanced MATSim model will be used to implement simulations of Madrid and Paris. Together, these models will be used to evaluate the intermodal solutions and information services proposed by TRANSIT (integrated ticketing, the Timetable Synchronisation Tool, the ADGT, etc.) in a variety of scenarios.

1 Introduction

1.1 Scope and objectives

This document describes the work completed in Work Package 5 of the TRANSIT (Travel Information Management for Seamless Intermodal Transport) project, namely the "TRANSIT Modelling and Simulation Framework". This modelling and simulation framework consists of enhancements made to the agent-based transport simulation MATSim, the development of the agent-based long-distance travel demand modelling framework J-TAP, as well as the development of two new optimisation tools, the AMAN/DMAN Ground Tool (ADGT) and the Timetable Synchronisation Tool. Before describing these tools, it is useful to introduce the goals of the TRANSIT project as a whole.

The TRANSIT project is focused on challenges currently facing the planning, management, and operation of long-distance passenger travel services and infrastructure within Europe. The goal of the project is to develop a set of multimodal key performance indicators (KPIs), mobility data analysis methods and transport simulation tools allowing the evaluation of the impact of innovative intermodal transport solutions on the quality, efficiency and resilience of the door-to-door passenger journey.

The specific objectives of the project are the following:

1. Propose innovative intermodal transport solutions based on information sharing and coordinated decision-making between air transport and other transport modes.
2. Develop multimodal KPIs to evaluate the quality and efficiency of the door-to-door passenger journey.
3. Investigate new methods and algorithms for mobility data collection, fusion and analysis allowing a detailed reconstruction of the different stages of long-distance multimodal trips and the measurement of the new multimodal KPIs.
4. Develop a modelling and simulation framework for the analysis of long-distance travel behaviour that allows a comprehensive assessment of intermodal solutions in terms of the proposed multimodal KPIs.
5. Assess the expected impact of the proposed intermodal concepts and derive guidelines and recommendations for their practical development and implementation.

Objective 1 has been addressed in deliverable D2.1 'ATM Role in Intermodal Transport: Opportunities for Innovative Intermodal Concepts and Passenger. These coordination mechanisms are organised into two main categories:

- mechanisms relevant for strategic level planning
- mechanisms relevant for tactical decision-making

Objective 2 is addressed in D3.1 'Multimodal Performance Framework', while Objective 3 is covered by D4.1 'Methodologies and Mobility Analytics Algorithms for the Analysis of the Door-to-Door Passenger Journey'.

This document focuses on Objective 4, reporting the development effort undertaken for the development of the TRANSIT modelling and simulation framework. The goal of this effort was to adapt

EUROPEAN PARTNERSHIP

Co-funded by the European Union

existing, state-of-the-art transport modelling tools to render them capable of dealing with the full variety of new multimodal (long-distance, passenger transport) concepts proposed by TRANSIT. To achieve this goal, we have faced three specific challenges:

1. Update the C-TAP model of continuous long-distance travel to render it capable of dealing with the full variety of multimodal options.
2. Integrate the C-TAP model with the MATSim framework for the simulation of daily travel.
3. Develop the optimisation algorithms required to simulate the proposed intermodal solutions.

1.2 Applicable documents

- Grant Agreement No 893443 ITACA – Annex 1 Description of the Action.
- TRANSIT D1.1 Project Management Plan, Edition 01.01.00, August 2020.
- TRANSIT D1.2 Data Management Plan, Edition 02.00.00, June 2021.
- TRANSIT D2.1 ATM Role in Intermodal Transport: Opportunities for Innovative Intermodal Concepts and Passenger Information Services, Edition 01.00.00, March 2021.
- TRANSIT D3.1 Multimodal Performance Framework, Edition 01.01.00, March 2021.
- TRANSIT D4.1 Methodologies and Mobility Analytics Algorithms for the Analysis of the Door-to-Door Passenger Journey, Edition 00.01.00, May 2021.

1.3 List of acronyms

Table 1: List of acronyms

Acronym	Definition
ABM	Activity-based travel demand models
ABTDS	Agent-based travel demand simulations
ADGT	AMAN/DMAN Ground Tool
AMAN	Arrival Manager
AOC	Airport Operation Centre
ATM	Air Traffic Management
CDG	Charles de Gaulle
C-TAP	Continuous Target-Based Activity Planning
DI	Dependency Injection
DMAN	Departure Manager
DSS	Decision Support systems
GHG	Greenhouse Gas
GPS	Global Positioning System
GTFS	General Transit Feed Specification

Acronym	Definition
GTS	Ground Transportation Suppliers
GUC	Geographic Unit Cells
IoC	Inversion of Control
J-TAP	Java Target-Based Activity Planning
LDT	Long Distance Travel
MATSim	Multi-Agent Transport Simulation
OD	Origin-Destination
OSM	Open Street Map
RTA	Requested Time of Arrival
SA	Simulated Annealing
SEM	Structural Equations Models
TMA	Terminal Manoeuvring Area

1.4 Document structure

This document is structured as follows:

- Section 2 describes the model developments required to satisfy the requirements of the TRANSIT project.
- Section 3 describes the tools developed or evolved within the project, including the chosen design, methods, limitations, and suggested future evolution.
- Section 4 describes where to access more detailed technical documentation, which in the case of J-TAP and the MATSim modifications is intended to be regularly updated to accommodate future changes. It also describes where the code repositories are stored, includes a brief introduction to using said repositories.
- Section 5 presents the main conclusions extracted from the development of the modelling framework.

2 Required model developments

Before describing the required model developments as well as the models themselves, it is useful to define how two key terms are understood in the context of transport modelling. With regards to mode and route choice, *multimodality* refers to the ability to choose between different modes to accomplish a particular trip. However, in each alternative mode only includes one mode – each alternative is *monomodal*. The decision maker can only choose car *or* train *or* bike, not a combination of these modes, for each trip (access and egress are usually assumed to be completed on foot). *Intermodality* refers to the ability to choose between different modal alternatives, whereby each alternative can be monomodal or intermodal for a particular trip. With intermodal mode and route choice, decision maker can choose between car, rail+bike, car+rail, rail, bike, car+bike, etc. for each trip.

This may seem puzzling to those readers who work mostly with long-distance travel, but these definitions stem from the fact that a) most transport models model local, daily travel and thus b) work with multimodal mode and route choice, since intermodal mode and route choice is substantially more difficult to implement and calibrate. This difficulty stems partially from the fact that intermodal travel behaviour is more difficult to gather data on, largely because it is still relatively rare, especially in local and daily travel. Other sources of difficulty have typically been because it is more difficult to create the set of reasonable alternatives, and partially because intermodal routing is more difficult than monomodal routing, since usually different routing algorithms are necessary for each mode. For instance, scheduled modes, like public transit or long-distance rail or air, require more complex algorithms than network modes, like car or bicycle.

With regards to modelling transport networks, the definition of multimodal is slightly different. A *multimodal network* is a network that contains multiple monomodal networks and connects them. A truly multimodal network enables intermodal routing. Many transport models technically already have a multimodal network made up of several monomodal networks (a road network and a public transit network) connected by links that allow transferring between the two, but each network has such different characteristics that routing is only possible for monomodal trips (the transfer links usually only serve to allow the modelling of access and egress to public transit on foot). J-TAP has developed a truly multimodal network that allows one routing algorithm to explore the whole multimodal network. It does this partially by increasing the abstraction of how the scheduled modes (rail, air) are represented, partially providing special links that represent intermodal transfers, and partially through the way the link weights (the cost of moving through the network) are calculated.

2.1 Mode and route choice modelling for intermodal long-distance travel

The objective of this task was to extend the C-TAP model [1] to include an explicit network model of available intermodal travel options and the corresponding intermodal mode choice and route choice models. The goal was to provide a framework that allows the development of a large-scale model able to deal with the full range of long-distance travel services available at the level of a country or region. In service of this goal, it was planned to implement modifications of the C-TAP library to allow the simulation of the new intermodal concepts and passenger information services, with special attention given to the computational speed of the implementation.

The C-TAP model [1] is an activity-based, long-distance travel demand model that uses optimisation to evaluate and modify agent activity chains, which are called "plans" and which include generalised travel costs in the optimisation function, thus implicitly including mode and route choice. The optimisation function also considers the generalised costs of performing activities that can be location and seasonally specific, thus implicitly including destination choice. It's typical inputs including simplified, monomodal, point-to-point transport networks, a synthetic agent population, agent plans, and destinations at which activities such as "vacation" or "visiting friends/family" can be performed. The destinations need to include an "attractiveness" attribute, which can be dependent on seasonality. C-TAP's typical output includes optimised agent plans (when, where, for how long, and how often an agent travels and for which purpose) and origin-destination matrixes of trip flows (how many people travel from A to B during the time frame specified for the model, and which mode they used).

The objective of the C-TAP model is to find the optimal plan of an agent. An agent represents an individual in a population. Each agent has a set of characteristics (e.g., age, gender, income, members of his family, job type, and home location), which are used by the C-TAP model as input parameters to define his long-term optimal plan. An agent plan is defined by a sequence of activities and trips. The C-TAP model optimises the duration of activities and trips based on the specific needs and constraints of the agents. These needs and constrains are expressed as "targets". For instance, each agent needs to complete a certain amount of each activity type during the selected time horizon – this is represented by the "Percentage of Time" target, and represents that an agent might want to spend four weeks out of each year on vacation, two weeks into total visiting friends and family, and the rest at home. Each agent also has different preferences for how long each instance of an activity type should be (perhaps a week for a vacation, but only three days for a visit), how much it should cost, and how well performing an activity at a particular location will satisfy their need for that particular activity. These components also are represented by targets and/or weighting functions within the optimisation function. When the C-TAP model is run for many different alternative plans for a particular agent, an approximation of the overall optimal plan – sequence, duration, frequency, and location for each activity being considered – can be made by selecting the plan with the lowest discomfort value (the optimisation function that forms the core of C-TAP minimises discomfort). The time horizon for an agent plan considered in the model is usually one year.

The C-TAP library was developed to test and use the C-TAP simulation model. Besides the C-TAP simulation model, which applies only to a single agent, the library provides a rudimental form of an agent-based model, which aggregates the optimal plans of the agents in order to predict the overall long-distance travel demand to each possible destination. After a first attempt to extend the old C-TAP library with the necessary features to satisfy the objectives of the project (e.g., multimodal network), we concluded that implementing a new library would bring more advantages. The choice has been taken based on the following analysis.

Table 2: Analysis of C-TAP strengths and weaknesses in the context of TRANSIT

C-TAP advantages	C-TAP disadvantages
• It supports multiprocessing. This is easy to replicate in other languages. • It is written in C++, which usually guarantees better performances than other programming languages	• It is written with a 10-year-old version of C++ • It does not follow a precise development framework, which makes it hard extend it or plug-in new features. • Exception handling and information logging are not completely implemented, making difficult to detect errors. • It does not make use of unit testing, which makes very complicated validating the compatibility of new features. • It is insufficiently documented.

The technical team implemented a new library that overcomes the aforementioned limitations of the C-TAP library. The new library is called J-TAP, which stands for Java Target-Based Activity Planning. The library is written using a modern development framework in Java language that makes it easy to extend it even for non-professional developers. In addition, it also includes many tools to make the generation of model inputs easier.

J-TAP also includes a new implementation of C-TAP, which allows for the evaluation of intermodal trips and the use and output of actual routes, not just point-to-point simplifications (see 3.1). This allows for more detailed output. For instance, since the new implementation of C-TAP can route intermodal trips across a multimodal network, the output can be processed to produce plots of network loadings (how many agents used a particular link in the network during a specified timeframe), not just OD matrixes of flows. Thus, an analysis of which routes agents take or which modes they combine, and which intermodal connections they use, can be made.

2.2 Modelling of the interaction of long-distance travel with the urban transport system

The first objective of this task was to develop the required interface to integrate the C-TAP long-distance travel model (which was re-implemented and enhanced for TRANSIT and is now simply the default optimisation model included within the long-distance travel demand modelling framework J-TAP) with a model of regional or national everyday travel developed with the MATSim simulation software. The second objective was to implement the modifications of the MATSim code required to allow the simulation of the new intermodal concepts and passenger information services.

2.2.1 Interfacing J-TAP and MATSim

MATSim (Multi-Agent Transport Simulation) is an agent-based transport simulation model. In essence, it creates a simplified, "digital twin" of a city or region and simulates how people move in a typical day. Essential inputs include a synthetic population of agents (abstract representations of people), a multimodal transport network, and facilities (locations at which agents perform activities). Each synthetic agent represents a person and has a plan, which consists of a chain of activities and trips between those activities. The trips must initially specify at least the transport mode, a departure time, and a starting and endpoint. When MATSim runs, it will then adjust the details of these trips, performing mode and route choice, and depending on the implementation, also adjusting the departure times of the trips. The main outputs of MATSim are the detailed (second-by- second) activity plans for each agent, which include the transport mode and the associated link-by-link route chosen by each agent. From these activity plans, important metrics such as modal split can be derived, and "heat maps" of agent travel patterns can be produced, illuminating which parts of the network are most heavily used. Another important output of MATSim is the "events file", which contains each "event" produced by the traffic simulation. Such events include, for instance: when an agent enters a transit vehicle, when an agent driving a car enters or leaves a roadway segment, etc. The events file can be used to generate other useful outputs, such as network loading, passenger arrival distributions, maps of congestion at different times of day, or even videos of agent movement throughout the day. During the development of what is now known as J-TAP, it became clear that an integration of J-TAP and MATSim makes sense only in very specific cases, and that the results of J-TAP, which models long-distance travel behaviour, are more useful for city and regional simulations such as MATSim than the results of the city and regional simulations are for J-TAP. This is due to the differences in scale and level of detail, as well as data availability and computational limitations.

2.2.1.1 Inputs from MATSim to J-TAP

An agent (an abstract representation of a person) in J-TAP considers a (large) set of the available destinations when making decisions. Therefore, a true integration with regional models only makes sense if all the destinations have regional models of the same type and scope. Otherwise, the level of information about each destination will be very different across destinations. This would add yet another layer of uncertainty to the modelling process and thus the results. Since MATSim models are computationally expensive, data hungry, and require specialised modellers to build, run, and maintain them, it is unreasonable to expect that all regions will have such a model. Thus, it would be more methodologically consistent not to use MATSim models for any of the available destinations, than to use it for a small subset of the available destinations. However, it was agreed that despite the drawbacks just mentioned, the output of the regional models of Ile de France and Madrid would be used in the national J-TAP models of France and Spain, respectively. The decision to push forward with integrating the national J-TAP models with the capital city MATSim models was largely motivated by the desire to provide a proof of concept of how to integrate J-TAP and MATSim. This integration will consist in providing J-TAP with any changes in the travel time required to access long-distance modes from within the city that a proposed innovative multi-modal solution might produce. Since this is a very simple output, there is no need for an elaborated integration of the two models in this direction; ensuring this single output of MATSim is in a format compatible for J-TAP's methods suffices.

2.2.1.2 Inputs from J-TAP to MATSim

The J-TAP model produces information about long distance trips: origin and destination, transport mode, frequency and duration of the overnight stays, etc. This sort of information is useful for city or regional models. It can help such models fill a traditional gap in the data, namely about trips made by non-residents. Regional models usually only consider daily travel behaviour, not irregular or long-distance travel behaviour. Thus, many of transport models in use today ignore tourists, business travellers, and other trips generated by non-residents or irregular and long-distance trips generated by residents. Using the output from J-TAP, however roughly estimated such results would currently be, can help improve regional models by providing a quantitative way to estimate not only the number of trips generated by non-residents, but also their detailed attributes (access and egress modes, purpose, etc.). This is again simply a question of taking J-TAP output and formatting it such that the MATSim scenario generation process can use it.

2.2.1.3 Iterating J-TAP and MATSim in a loop

By ensuring that each model provides outputs in formats the other model can use, there is the possibility of running them iteratively in a loop. The runtimes of both models are non-negligible, so such an iterative process would be both time consuming and computationally intensive, and would thus likely be restricted to a single digit number of iterations. As a result, for now, the writing of special scripts to automate such a process is superfluous. As both tools mature further and become more computationally efficient or the price of computing continues to fall, such a script may become sensible to create in the future.

2.2.2 Modifying MATSim to better model intermodal concepts and passenger information systems

The enhancement of the MATSim simulation model to model intermodal concepts and information sharing systems will be useful for operational studies beyond the TRANSIT goal of coordinating air-land services in case of disruptions. In particular, it will be useful for modelling passenger information systems in general and also their effects on both individual passengers and the transport system as a whole, and will allow simulations of dynamic services, such as taxi fleets, to consider more realistic passenger behaviour during the traffic simulation itself. These modifications to MATSim also open up the possibility of integrating the AMAN/DMAN departure-delay optimiser in the future: indeed, we think this might be one of the most promising combinations of the TRANSIT tools for providing a "sand box" for testing tactical operational coordination between air and land transport operators.

2.2.2.1 Passenger information systems

The focus of the work in relation to the modelling of passenger information systems was to enable the simulated persons (i.e., the agents) to make route and mode choice decisions during the simulation. Although the already existing collection of MATSim code named "Within-day Replanning" has been used in the past for within-simulation route choice decisions by automobile drivers [2], there were no known studies or projects in which MATSim agents were able to make mode choice decisions during the simulation, which is the key new feature developed by TRANSIT.

2.2.2.2 Intermodal concepts

The intermodal concepts proposed by TRANSIT focus on intermodality of long-distance trips. This means the focus was on the choice between using just one long-distance mode (rail, car, air) to complete the long-distance legs of a long-distance trip and using two or more modes to complete these long-distance legs. Of particular interest was the combination of air and rail in the context of replacing feeder flights with high-speed rail.

Now, long-distance travel modes generally require access and egress legs: the passenger must get to the rail station or the airport before they can use the long-distance mode. Note that these *local, short-distance* access and egress legs are not truly part of long-distance intermodality, because they are a basic requirement of using most long-distance modes.

For this reason, and because of needing to reduce the complexity of the network and mode choice, how passengers get to the rail station or airport or highway (if using a car) is generally not explicitly modelled in J-TAP: J-TAP aggregates the origins and destinations into "city nodes" representing a metropolitan area or region, and by default assigns one generalised access travel time and cost between any one city node and the rail, rode, and air nodes it is connected to. The focus on J-TAP is on the long-distance legs of the trip (i.e., the flights, the high-speed rail service, the highway route chosen), not local access and egress legs.

MATSim, however, focuses exclusively on the local access and egress legs. Indeed, the case studies that will be investigated in WP6 focus on passengers accessing the CDG or MAD airports from their respective metropolitan regions. Such access and egress legs are rarely in and of themselves intermodal. Consider what local intermodality means: it means combining bicycling with public transit, bike-sharing with public transit, or using car to access public transit via park-and-ride. Note that transfers between public transport modes such as bus, tram, and subway, are not considered "intermodal" by any planner other than the transit agencies themselves. Accessing public transit by foot is also not considered intermodal since it is a basic requirement of using public transit. To the best of the author's knowledge, using park and ride to access public transit to then access an airport with public transit is rather rare, and combining the use of a personal bicycle or bike-sharing with transit to access an airport is a behaviour that the authors have not seen mentioned in any of the literature. Thus, *local* intermodality (car + public transit via park-and-ride, bicycle + public transit, bike-sharing + public transit, etc.) was not considered. The focus, instead, was on ensuring the city or regional simulation could depict how the long-distance intermodal concepts, such as integrated ticketing or operational procedures to help guarantee ground-air transfers, affect passenger behaviour while accessing or egressing from long-distance modes. Such concepts can be reflected in the MATSim model as follows:

- *Integrated tickets*: these agents have additional inertia to switching to modes that are not included in their tickets in the case of disruption. Perhaps they even have access to special modes, such as shuttles. These concepts can be expressed in MATSim by, for example, by explicitly making the modes available to the agent at the re-planning moment and making the parameters used in the mode choice model dependent on their status as an integrated ticket holder.
- *Ground-Air Operational Coordination*: agents can be programmed to react to unplanned disruptions as well as different levels of prior information about the disruption using the modified MATSim Within-day Replanning framework. This was the focus on the development of WP5 and the eventual case studies in WP6. Theoretically, agents could also be programmed to know that their connection is essentially guaranteed in the case of minor disturbances. Their knowledge could be translated into special behaviour by creating additional modifications to the modified MATSim Within-day Replanning framework, such as finding new routes that arrive "late" but within the window of transfer guarantee, instead of striving to find routes that still arrive "on time" but have undesirable attributes, such as excessive amounts of local transfers (as may be the case in public transit systems) or very high costs (taxis). However, this additional functionality would also require behavioural research, and thus was not performed. It would, however, be an interesting avenue for future research on disruption management.

2.3 Modelling of the new intermodal solutions and passenger information systems

The aim of this task is to develop the optimisation algorithms required to simulate new coordination mechanisms. An algorithm at strategic level and another one at tactical level have been designed for supporting collaborative decisions of two or more transport modes.

2.3.1 AMAN/DMAN ground tool

The AMAN/DMAN ground tool is a coordination mechanism at tactical level to handle a disruption on the ground side. For instance, when a train or a subway (providing access to the airport) breaks down, several passengers are delayed and are likely to miss their flight. Thus, delaying departure aircraft at the gate would help them catch their flight and would minimise the number of stranded passengers at the airport. However, delaying all the departure aircraft is likely to induce massive congestion at the airport. Information sharing between passengers, ground transportation suppliers and airport operation centre would make it possible to target and delay the subset of departure aircraft having the highest number of passengers affected by such disruption. In parallel, decisions on arrival aircraft would also be made to mitigate airport congestion. In order to implement such mechanism, an optimisation problem is set by making decisions on departing and arriving flights (departure/arrival time, departure/arrival runway). Capacity constraints on the airside and runway throughput are also considered. Constraints related to air-connected passengers are also added to the optimisation problem. The problem is NP-Hard, is applied on a large instance (hub airport), and, since the mechanism is at tactical level, the algorithm computation time needs to be low. Thus, a meta-heuristic called Simulated Annealing is used.

2.3.2 Timetable synchronisation between air and rail

Timetable synchronisation between air and rail is a coordination mechanism at strategic level. Currently, schedules are developed by transportation suppliers without consideration of other transportation mode schedules. This results in a suboptimal overall multimodal schedule. Considering two independent schedules for air and rail, respectively, the goal of such mechanism is to synchronise both schedules by applying small changes on them. This mechanism can consider more than two modes. Thus, long distance-bus schedules could also be incorporated into the optimisation problem.

First, the synchronisation of air and rail timetables is done at an airport by optimising the air/rail passenger transfer times. Aircraft turnaround time constraints are considered. Train/aircraft departure and arrival time changes are limited in range to not de-structure the initial schedules. This coordination mechanism can also be implemented at a national scale by considering a set of airports. In this case, the problem is much harder because synchronising two modes at an airport is likely to desynchronise these modes at another airport. Hence, a highly combinatorial problem can be generated; again, a Simulated Annealing is implemented to tackle this problem.

3 Description of the tools

3.1 J-TAP

Traveling long distances at a low cost has become accessible and affordable to anyone these days. It is estimated that in Europe, about 50% of all kilometres travelled come from trips beyond 100 km [3]. Therefore, it becomes crucial for transport planners and policy makers to obtain decision support systems to tackle new challenges, such as intermodal transportation, integrated ticketing, and reduction of greenhouse gas (GHG) emissions. Activity-based travel demand models (ABMs) take into account the link between activities and travel for an individual but also for different people within the same household[1]. More attention is thus paid to time and space constraints, bringing significant advantages over simple aggregated trip-based travel demand models. Critical elements of ABM are activity generation, activity scheduling, and mobility choices. Due to several challenges listed below, long-distance travel demand is not usually modelled by ABM, despite the tremendous benefits it can bring. Firstly, long-distance travel cannot be analysed focusing on a single day, the typical time frame for most ABMs, because long-distance journeys usually consume more time and are also planned well in advance. Long time periods, many different and distinct types of possible activities and destinations, as well as a complex and extensive transportation network, are the main barriers to using ABM to model long-distance travel demand. This section introduces J-TAP, which is a Java framework developed within TRANSIT for implementing ABMs to model long-distance travel demand. It provides an overview of the J-TAP architecture, several examples for using specific library packages, as well as the default models and solvers.

3.1.1 Context

Long-term planning and design of transportation infrastructure and new transportation services depend on complex and interacting economic, social, environmental, and technical factors. Knowledge of transportation demand for future scenarios becomes a crucial decision factor for transport planners. Travel demand models predict the number, destination, and modal choice of trips generated by a population considering both their frequency and duration. Travel demand can be divided into urban travel demand, such as daily short-distance commuting, and long-distance travel (LDT) demand. In general, LDT is defined based on one or more of the following criteria: spatial distance, travel time, frequency and regularity, and overnight stay. LDTs arise from activities of national and international scope and are usually a combination of different transportation modes. Forecasting these activities with models represents a challenge for transport planners and policy makers.

In recent decades, a sharp increase in global passenger travel distances has been observed in most developed countries due to economic growth, technological progress, and increased accessibility of

[1] Please note that J-TAP is a framework. This means that if the modeller has the appropriate data upon which to base a quantitative model of how groups of people plan their trips together, J-TAP can accept this model and integrate it into the larger modelling framework. However, the development of such a quantitative behavioural model for group planning behaviour was not part of the scope of TRANSIT. The development of such a model would require a dedicated activity focused on data collection and analysis of such group planning behaviour and subsequent estimation of quantitative behavioural models, such as discrete choice models.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

faster motorised modes of travel [4]. The shift from low-speed to high-speed transportation will likely accelerate even more in the future [5]. To illustrate the share of total person-kilometres travelled generated by LDT, consider that while approximately 75% of trips in the United States are less than 15 km, they account for only 28.9% of vehicle-miles travelled [6]. On the contrary, trips greater than 150 km account for 1% of trips, but represent 15.5% of vehicle-miles travelled. Similarly, 50% of all passenger kilometres in Europe correspond to trips over 100 km [3].

LDT is crucial to understanding current and future mobility in an increasingly connected society. However, despite their importance, long-distance trips have received less attention in travel behaviour models and data collection than daily trips. LDT is captured to some extent in conventional household travel surveys but, being a "rare event" for most individuals, it does not fit into the short reporting period of most travel diaries. Surveys of national long-distance travel have been conducted in several countries. Collecting data on long-distance travel has proven to be a major challenge in the past primarily due to recall limitations and the inability of the participant to accurately estimate travel distances and the lack of collection of personal variables to facilitate prediction models. Despite these difficulties, several studies in the literature take into account sociodemographic factors to model long-distance individual travel. These include age [7], gender [8], occupation [9], income [10], and number of children [10] among others. An overview of the main person-level socioeconomic measures and trip information that should be considered in long-distance travel modelling can be found in [11]. Other important factors are the amount of time spent on vacation travel during the year [12], the vacation destination attractiveness [13], the accessibility, the transport mode comfort, and the activities' purpose, among others.

In the last few years, the introduction of new sources of big data, e.g., travel data collection using passive and non-passive mobile devices, has opened up the possibility of developing complex models that have the potential to make reliable predictions of long-distance travel demand. Studies of long-distance travel demand in the literature can be divided in two categories. Firstly, there are those models that are more general with regards to the estimation of long-distance demand as a required component for large-scale models (e.g., country wide); secondly, there are those models that are focused exclusively on specific corridors, such as high-speed rail lines [14], or certain demand or supply segments, such as vacation trips.

Models of long-distance travel demand can be divided into two different types: trip-based models and activity-based models. Trip-based models are the simplest version of a statistical model for travel demand. Trip-based models mostly ignore the cause of travel, namely activities. One reason for this approach is the lack of information about the purpose of travel. This is the case if the underlying data source is passive data collection such as GPS or mobile network data. However, a crucial assumption in travel demand modelling is that travel is induced by the desire to perform certain activities in certain locations. This assumption has led to the development of activity-based models, and this assumption also applies to the modelling of long-distance travel.

Activity-based models consider a trip as the result of activities in different locations. Activity-based models have been used to explain all the components of long-distance travel demand. These components include, for instance, what activities are performed, when, where, for how long, for and with whom, and the travel choices people will make to complete them. The activities in these models usually induce home-based tours. For instance, binary choice models for long-distance tours were used in Ohio [15] to explain the motivations for long-distance travel. At a later stage, mode choice models instantiating nested-logit models have been implemented on several occasions in Europe [3] or in the US ([16], [17]). In California, Structural Equations Models (SEMs) as well as Latent Class Analysis were used to estimate miles travelled per mode and length of long-distance tours [18]. By summing all activity-based models, a complete picture of long-distance travel demand can be generated. For instance, a sequence of interdependent sub-models representing choice of tour frequency, tour destination, travel mode, and other related choices [19]. This approach has been further enhanced to develop a national tour-based model in the US [20].

However, "there's no such thing as a free lunch". Indeed, more complicated models, to which activity-based models belong, require new types of data, which are usually not considered in national long-distance travel surveys. In addition, defensible travel forecasting is a challenging endeavour regardless of the modelling framework, and the most complex models will be more difficult to use than simpler models from a variety of perspectives. Despite the aforementioned difficulties, activity-based modelling in travel forecasting has the ability to provide a full range of quantitative dimensions to represent travel-inducing activities and choices.

In a trip-based model, first aggregate estimates of the demand are predicted. Next, each subsequent step in the model further disaggregates the overall aggregate estimates of the demand. Once all disaggregation steps have been performed, the demand can be entered into the network assignment model (routing trips on the transport network and calculating how many trips will use specific elements of that network). In contrast, in an activity-based model, disaggregate estimates of the demand are predicted first, and then these estimates are aggregated by geographic area and time for input into the network assignment model. The demand side of the problem (activity-based models) and the supply side of the problem (either assignment or simulation models) are usually decoupled since the activity models typically compute probabilities for a large number of alternatives, which requires an explicit choice set. To account for such alternative sets in assignment or simulation procedures for real size networks would result in very high computation times [21].

Agent-based travel demand simulations (ABTDS) simulate the (travelling) behaviour of virtual agents individually, e.g., by using the output of an activity-based model. Agent-based models are able to capture emergent phenomena resulting from interactions among agents and between agents and the external environment, e.g., transport supply. Furthermore, agent-based modelling provides a natural description of a system, since each person and their behaviour are individually modelled by a single agent. Finally, agent-based modelling is flexible because the modeller can easily add or modify agents in the system without manipulating the core of the simulation. One of the well-known approaches for agent-based simulations of travel behaviour is MATSim [22]. In MATSim, agents choose a daily schedule for their behaviour and execute it. The results of the execution are reported and agents can re-plan their schedule based on the results of all agents. This procedure is repeated until a stochastic user equilibrium with consistent travel demand is reached.

The agent-based simulations of travel behaviours that are currently available lack several features that are crucial to accurately study LTD. For instance, they usually consider a time horizon of a single day or at most a week [23] **,** which is not sufficient for modelling long-distance travel demand. The rapid growth and significant policy implications of LDT require the development of agent-based simulations of travel behaviours that can help transport planners and policy makers model the future LDT networks. In TRANSIT, this gap in the modelling toolbox available to travel demand forecasters is addressed by the development of J-TAP.

3.1.2 Research motivations

Moving from a trip-based model to an activity-based and agent-based model for estimating LDT demand is not straightforward and presents several challenges including:

- data limitations;
- longer computation time;
- big data manipulation;
- increased efforts to calibrate the model;
- lack of confidence in this new modelling paradigm;
- lack of experience with activity-based models;
- models need to be agile to be prepared to analyse a large range of scenarios (including unforeseen scenarios);
- models need to be fast enough to allow for multiple model runs;
- lack of established software packages

J-TAP aims to fill the gap by providing a Java library that contains packages intended to:

- simplify the creation and manipulation of complex transport networks;
- simplify the creation and manipulation of a synthetic population;
- simplify the use of efficient routing algorithms;
- simplify the use of a wide range of graph algorithms;
- support the creation of new activity-based models (e.g., continuous target-based activity models);
- simplify the use of optimisation solvers;
- run agent-based simulations combined with activity-based models;
- run multiple scenarios in parallel;
- simplify calibration processes.

3.1.3 Intended use cases

Real cases in which forecasting LDT demand results in a critical task and an activity-based model would bring advantages over a traditional trip-based model are listed below. In all the following cases J-TAP can provide precious help to simplify and speed up the development of activity-based models for LDT at different levels during the process.

- test national policies (e.g., airport/railway planning);
- measure the performance of the transport network along economic, social, and environmental dimensions;
- measure the benefit of multimodal transport systems;
- evaluate the impacts of private-sector decisions;
- evaluate the impacts of new infrastructures using multi-criteria methods;
- predict the effects of building high-speed rail corridors;
- predict the benefits of simplified and integrated ticketing.

3.1.4 Design

This section is aimed at technical readers who wish to gain an overview of how and why J-TAP was designed, and how it is intended to be used. Such technical explanations are essential components in ensuring the tools and methodology developed in TRANSIT will continue to be reproducible and useable after the project ends.

J-TAP is intended to be an open-source library. As with any open-source project, a "building together" approach to software design is crucial to growing a large community that can support future developments. Extensibility is a cardinal point. An extensible product must be designed from its earliest stages for customisation and enhancement. This makes it easy to expand feature sets and enrich current functionalities. Maximising extensibility is one of the main goals through all aspects of J-TAP. The architectural guidelines in the J-TAP software development are:

- **replacing or extending** the core code rather than modifying; this supports efforts to maintain the integrity of the tested code provided by the core, while allowing for broad customisation;
- **dependency on known architectural and programming structures** that can reduce the learning curve for new J-TAP developers;
- **designing self-contained modules** organised by feature, in order to reduce external dependencies on each module and allow for replacement without affecting other areas of the code.

J-TAP is designed following the SOLID principles. SOLID principles are an object-oriented approach that is applied to software structure design so that it is easier to scale and maintain. Robert C. Martin conceptualised this concept [24]. SOLID principles ensure that the software is modular, easy to understand, debug, and refactor. SOLID stands for

- S: Single Responsibility Principle (SRP)
- O: Open closed Principle (OSP)
- L: Liskov Substitution Principle (LSP)
- I: Interface Segregation Principle (ISP)
- D: Dependency Inversion Principle (DIP)

Below we describe J-TAP's core architecture and its main packages. The classes in these packages are designed to be easily extended. J-TAP relies on Google Guice, an open-source framework that provides support for dependency injection (DI) using annotations to configure Java objects. DI aims to address separation of construction concerns, increasing readability and code reuse.

3.1.4.1 Architecture

J-TAP is a library containing different packages that can be used independently or jointly to develop an activity-based model to predict LDT demand. Each step of the model development is supported by a different package or a set of them that aim to simplify the specific task. Figure 1 shows an overview of the architecture, dividing the packages into the J-TAP core (right) and the external sources (left), which are used by the core packages to store and manipulate data. Since J-TAP is in an early stage of development, the architecture may change slightly in the future. In the following sections, each package is presented in more detail, highlighting its functionalities and possible uses at different stages of the model development.

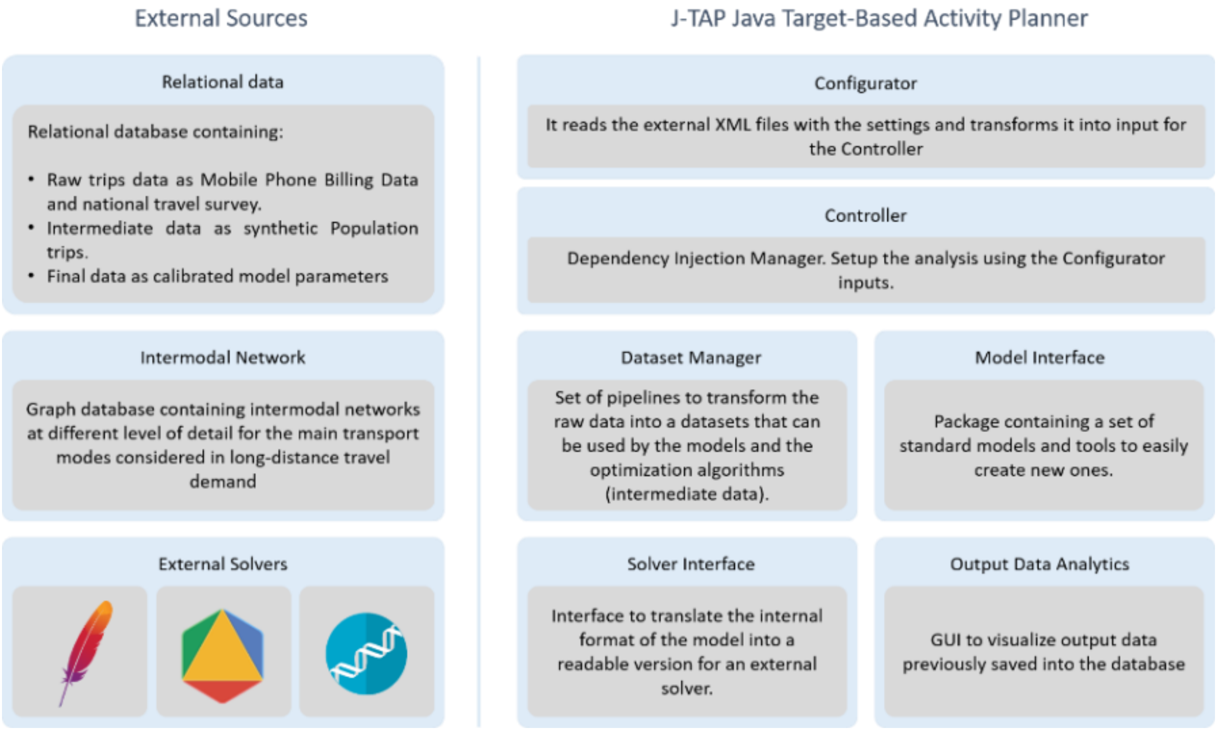

Figure 1: J-TAP Architecture

3.1.4.2 Multimodal network

Agents must interact with their environment. The environment in J-TAP is a network. Agents move through the network from one node to another using links. In order to exploit the maximum potential (efficiency and reusability) of the network, a graph database is used to create, store, and query the network. We decided to use Neo4j to perform these tasks.

Some advantages of using a graph database and specifically the Neo4j implementation are:

- Graph data structures are easily represented (e.g., transport network).
- It provides efficient algorithms to retrieve/traverse/navigate through connected data (e.g., routing algorithms).
- It represents semi-structured data very easily.
- Neo4j provides CYPHER, a query language in human readable format.
- Complex Joins queries are not required to retrieve connected data.

J-TAP contains packages that simplify network creation in the graph database. These packages contain classes that allow for converting various sources that are stored in non-graph data structures into graph data structures (e.g., OpenStreetMap (OSM) data and General Transit Feed Specification (GTFS) data). In addition, these packages provide several methods for connecting different transport mode networks together, thus creating a multimodal network.

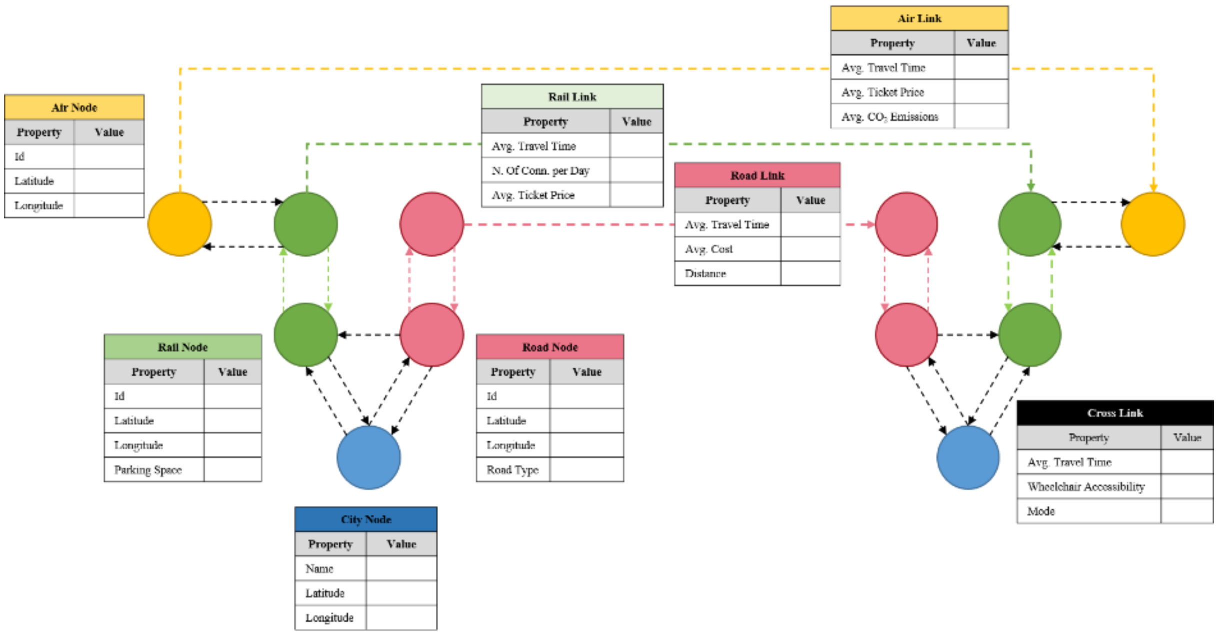

Figure 2: Example of multimodal transport network

Figure 2 shows the data structure of an example multimodal transport network generated in Neo4j using J-TAP network packages. In this example, three types of networks are connected together: road network, rail network, and air network. The different colours, which are used to depict the nodes and connections, indicate that the nodes and connections belong to different networks. Each network is connected to other networks via *Cross Links*. The *Cross Links* can be created using a J-TAP class that provides methods that create new connections between existing nodes using different measures, e.g., minimum distance between nodes belonging to different networks.

Each node and link in the network contains a map of properties that can be used to query the network. In addition to transport networks and their connections, J-TAP provides a set of classes and methods for inserting and connecting what we call geographic unit cells (GUCs). GUCs are fundamental components because they represent the spatial granularity of the model, e.g., countries, regions, cities, neighbourhoods, facilities. Figure 3 shows how this approach allows for a high degree of scalability.

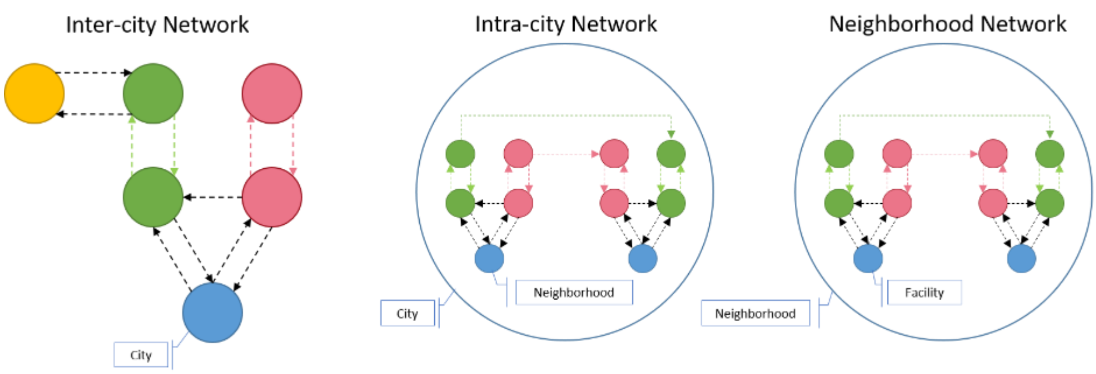

Figure 3: Transport network scalability

J-TAP routing packages rely on the graph database to generate the dataset containing the routes required by a model, such as C-TAP. All routes are usually precomputed and saved in a hash map to make them quickly accessible during the optimisation process. Different networks might be needed based on the study to be performed, the transport modes that should be available for a particular agent, the link weights used to represent different components of travel utility or disulity, the possible destinations to be considered, and the acceptable computation time. This means that the complete graph generated in the graph database may be not entirely necessary or adequate for a specific study. Therefore, it is critical to have a mechanism to easily manipulate the original graph to create and use a safe and efficient projection of it. The Neo4j *graph data science library* provides the *graph catalogue*, which can handle the aforementioned task. The graph algorithms are executed on a graph data model, which is a projection of the Neo4j property graph data model. A graph projection can be considered as a materialised view on the stored graph, containing only relevant analytical, potentially aggregated, topological and property information. Graph projections are stored entirely in-memory using compressed data structures optimised for topology and property lookup operations. J-TAP routing packages provide a set of methods to easily create the graph projections and execute routing algorithms to generate the routes dataset for the optimisation model, while hiding the necessary Cypher queries from the user.

3.1.4.3 Configurator

J-TAP can be configured using the config file. The config file builds a bridge between the user and J-TAP. The config file consists of an XML file containing a set of parameters that the user can modify to set up the study. New features can be entered and configured by defining new tags in the config file. The tags need to be bound with the properties of a new class that hosts the information and make them available at run time in J-TAP. The config file is used to initialise the Controller.

3.1.4.4 Controller

The controller is a crucial component as it represents the interface that the user can use to set up all the components needed for a specific study. J-TAP relies on Google Guice to achieve Inversion of Control (IoC) between classes and their dependencies. A fundamental step in IoC is binding ("wiring", in case the reader is more familiar with Spring) the interfaces with the classes that implement them. Therefore, the biding procedure in Google Guice defines how to inject dependencies into a class. The J-TAP controller defines the methods that the user can use to override the default bindings, defining their own classes and then customising the J-TAP components. In addition, the controller is responsible for the creation of the Injector, which is responsible for building the graphs of the objects that make up the specific J-TAP application. The injector keeps track of dependencies for each type and uses bindings to inject them.

3.1.4.5 Dataset

The dataset package provides several tools to manipulate, store, and integrate into the external dataset of J-TAP databases. Activity-based models and agent-based models require a large and heterogeneous dataset. Thus, a central database is needed. This central database is designed to channel all the data needed for the model results into a powerful tool. This tool should be able to simplify multi-scenario analysis, ensure reproducibility (obtaining consistent computational results using the same input data), study replicability (obtaining consistent results across studies aimed at answering the same scientific question), and simplify data conversion to several formats. Below, we show pipelines in the dataset package that simplify data integration into the graph database.

3.1.4.5.1 Facilities

Facility location choice models can be integrated into activity-based models, bringing many advantages. In this case, it becomes critical to assign a set of facilities for each GUC in order to assign an attractiveness rate to the GUC based on, for instance, the number of restaurants in the area, the number of parking spaces, etc. J-TAP provides a pipeline that uses OpenStreetMap data to assign facilities to a specific GUC based on the minimum distance between each facility and all the GUCs in the scenario. Figure 4 shows an example, where the circle in blue represents a GUC (e.g., a city) and the circles in orange represent different facilities assigned to that GUC. Each facility has a map containing a set of properties that can be used by the attractiveness function of the GUC.

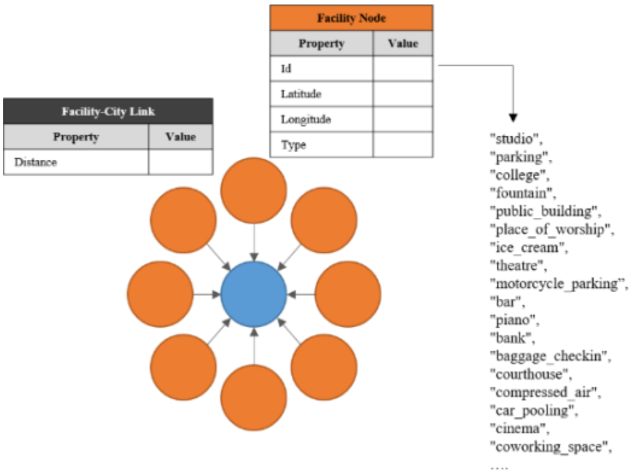

Figure 4: Facilities

3.1.4.5.2 Synthetic population

Each agent has different characteristics. These characteristics are critical in determining the agent's choices and interactions with other agents and the environment in which they live, e.g., age, gender, income, occupation, household, travel budget, disabilities, etc. J-TAP provides a pipeline to integrate the synthetic population into the graph database. The integration allows for the creation of a multi-layered network in which the multimodal transport network is connected with the agents, which allows, for instance, the agent to be linked to its city of residence or for an agent's perceptions of a destination at different times of year to be stored. It also allows a social network representing the relationships among the agents to be represented, and connected to the multimodal network. Social networks have an import impact on the long-distance travel demand [25]. However, developing improvements to C-TAP that could incorporate social network effects were out of the scope of the TRANSIT project: here, the idea was simply to make it possible for future work.

Figure 5 shows an example, where the circle in blue represents a GUC (e.g., a city) and the circles in green represent categories of agents. Each category is connected to one or more GUCs if at least one agent belonging to that category lives in the GUC. The link connecting the agent category to the GUC is called Agent-Resident Link and can contain information such as the number of agents in that category living in the GUC, the average accessibility rate to public transport, etc. In addition, agent categories can also be connected one to another, creating a kind of abstract social network representing how different agent categories are likely to interact. The current implementation of J-TAP foresees social networks that have two types of links: one including links to categories representing family members and the other including links to categories representing friends.

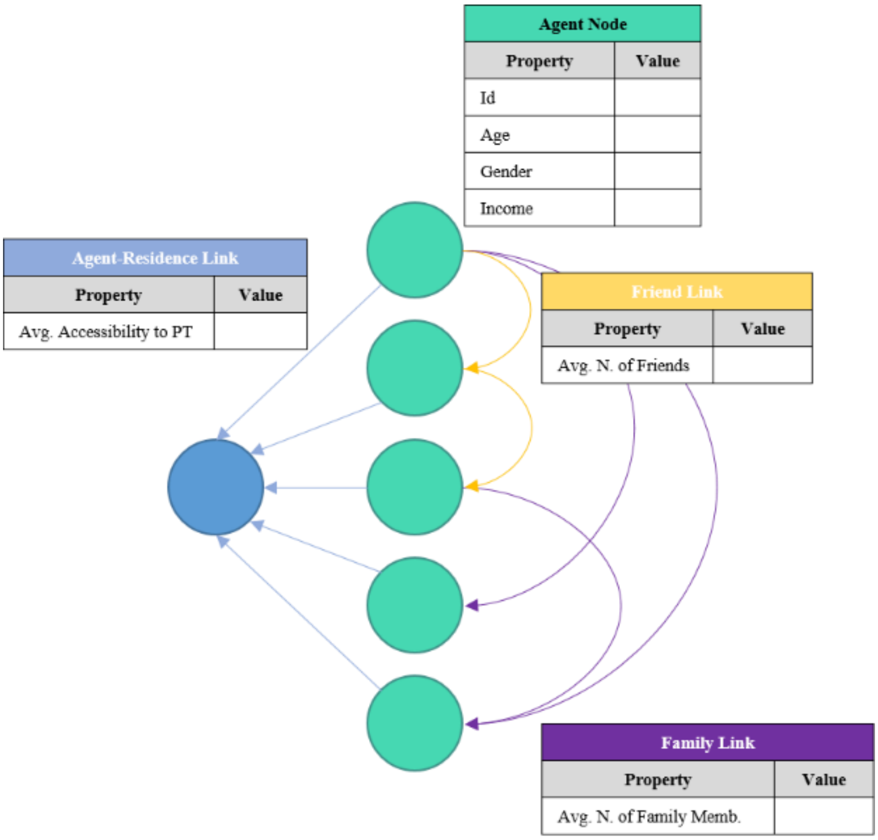

Figure 5: Schematic of a synthetic population example

3.1.4.6 Model

Activity-based models are often optimisation models involving the optimal scheduling of the activities, their locations, their duration, the transport mode used to move from one activity to another, etc. Optimisation models can be solved using different algorithms based on the model type (linear, non-linear, integer, mix-integer, constrained, etc.). The optimisation modelling lifecycle is depicted in Figure 6. An integrated lifecycle promotes rapid development and reliable results. In order to provide an integrated optimisation modelling lifecycle, J-TAP provides a package that can be used to easily create a model (objective function and constraints) and integrate it into a pipeline. The pipeline may, for instance, include pre-processing the input data, converting the model to the format required by the solver, and analysing the result of the optimal solution.

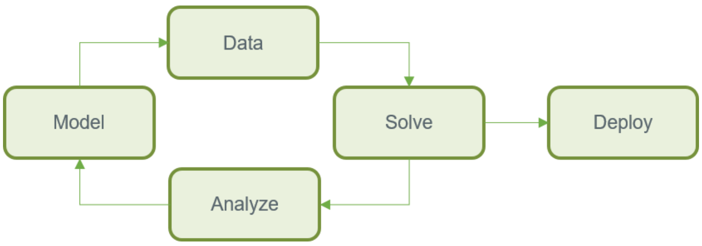

Figure 6: Optimisation model lifecycle

3.1.4.7 Solver package

J-TAP can make use of external solvers to solve optimisation problems (e.g., Apache common). Different solvers and algorithms accept different model formats (e.g., genetic algorithms). The J-TAP solver package provides a blueprint for creating methods that convert the J-TAP model format into a format that can be accepted by the selected solver. In addition, there are some solvers implemented in J-TAP itself that are used by the C-TAP model.

3.1.5 Activity target-based models in J-TAP

J-TAP provides a package containing a default activity target-based model, namely the Multimodal C-TAP Model, that is ready to be used. This model can be used as is or can provide a basis for implementing other models. In this section, this model is described.

3.1.5.1 Intermodal C-TAP model

Below, we present a multimodal transport version of C-TAP. C-TAP, which stands for Continuous Target-Based Activity Planning, is an optimisation model that was initially developed by [26] and extended to consider transport modes between locations by Janzen in his doctoral thesis [1]. The aim of the model is to predict the long-distance travel demand of an agent over a period of one year. The agent is defined by its behavioural targets. The agent satisfies its targets by executing corresponding activities. In contrast to iteration-based models, a continuous planning model does not iterate to a steady-state but continuously generates an activity schedule without any replacing. Below we show the C-TAP mathematical model for long-distance travel demand, highlighting the transport options the agent can choose between consecutive locations in his plan.

3.1.5.1.1 Decision variables

The decision variables are represented by two vectors:

- t^s vector containing the start time of each activity in the agent plan
- t^e vector containing the end time of each activity in the agent plan

Therefore, the vectors' length is equal to the number of activities in the agent plan *N*. The activity sequence of the agent's plan is an input data of the model. In the following chapters we refer to the start time of the activity at index *i* as t_i^s and the end time as t_i^e.

3.1.5.1.2 Parameters and functions

All the parameters and functions (Table 3) are specific for an agent category; this creates heterogeneity in the synthetic population when optimised plans are aggregated to generate the long-distance travel demand. Parameters and functions are generally created in an early step of the pipeline that feeds the model.

Table 3: Parameters and Functions that define Agent Categories and their behaviour

$\alpha \epsilon A$	Activity type
$\lambda \epsilon L$	Location
$\mu \epsilon M$	Transport mode
$i \epsilon N$	Activity index in the agent plan
α_i	Activity type at index i in the agent plan
τ_α	Calibration constant of the state value increasing function
κ_α	Calibration constant of the state value decreasing function
ε_λ	Agent perception of location λ ($\varepsilon_\lambda \in [0.5,1.5]$)
$S_\lambda(u)$	Attractiveness per unit time per location λ
$c_D(\lambda_i, t_i^s, t_i^e, \alpha_i)$	Cost of the non-business activity at index *i*
$c_M(\mu_i, \lambda_{i-1}, \lambda_i)$	Cost of travelling between the location of the activity at index i-1 (λ_{i-1}) and the location of the activity at index *i* (λ_i) using the transport mode μ
$c_v(t_0^s, t_n^e)$	Cost of the time budget consumed during the entire plan
B_C	Monetary budget
B_V	Time related budget
γ_b	Budget discomfort
$\omega^{time}(\lambda_i, \lambda_{i+1})$	Travel time between the location of the activity at index *i* of the agent plan and the location of the activity at index *i+1*

Several further concepts important to the model are described below.

- **State value**. The state value, v_i, is dynamic and it is calculated recursively using the state values of previous activities in the agent plan. v_i is represented by two different functions depending on if at position *i* of the agent's plan the activity is performed or not.

$$v_i = \begin{cases} \hat{v}_i(t_i^s, t_i^e, \alpha_i, v_{i-1,\beta}) & if\ the\ activity\ \alpha\ is\ performed\ at\ i \\ \check{v}_i(t_i^s, t_i^e, \alpha_i, v_{i-1,\beta}) & otherwise \end{cases}$$

- **State value increasing function**. Every time an activity is performed the state value for that activity increases, thus reducing the need to perform that activity in the future.

$$\hat{v}_i(t_i^s, t_i^e, \alpha_i, v_{i-1}) = 1 + (v_{i-1} - 1) \cdot \exp(-\tau_{\alpha_i} \cdot \phi_\lambda(t_i^s, t_i^e) \cdot (t_i^e - t_i^s))$$

- **State value decreasing function**. Every time an activity is not performed the state value for that activity decreases, thus increasing the need to perform that activity in the future.

$$\check{v}_i(t_i^s, t_{i+1}^s, \alpha_i, v_{i-1}) = v_{i-1} \cdot \exp(-\kappa_{\alpha_i} \cdot (t_{i+1}^s - t_i^s))$$

- **Pull factor.** Some of the variables described above reduce the probability to visit a specific destination λ. In order to model this probability, we introduce a pull factor, $\phi_\lambda(t_i^s, t_i^e)$. The pull factor combines the attractiveness of a location, $s_\lambda(u)$, and the agent's perception of a location, ε_λ. Moreover, the attractiveness is a function of time in order to represent seasonal influences.

$$\phi_\lambda(t_i^s, t_i^e) = \varepsilon_\lambda \cdot \int_{t_i^s}^{t_i^e} s_\lambda(u) \frac{du}{(t_i^e - t_i^s)}$$

3.1.5.1.3 Optimisation problem definition

Objective function

$$\min_{\alpha,\lambda,t^s,t^e} D_{pot} + D_{dur} + \gamma_b \cdot D_{bud}$$

Instead of computing a heuristic value, which is based on the discomfort reduction of a single activity, the full discomfort value is used as the driver for the activity planning. All the activities and their targets are considered at the same time.

- **Percentage of time discomfort**

$$D_{pot} = \sum_{\alpha \epsilon A} (T^{\alpha}_{pot} - v_n(t^s_n, t^e_n, \alpha_n, v_{n-1}))^2$$

- **Duration discomfort**

$$D_{dur} = \sum_{i=1}^{N} (T^{\alpha_i}_{dur} - v_n(t^s_n, t^e_n, \alpha_n, v_{n-1}))^2$$

- **Budget discomfort**

$$D_{bud}(\lambda, \mu, t^e_n) = \frac{(\sum_{i=1}^{n} c_D(\lambda_{i,} t^s_i, t^e_i) \cdot c_M(\mu_{i,}\lambda_{i-1}, \lambda_i))^2}{B^2_C} \cdot \frac{c_v(t^s_0, t^e_n)^2}{B^2_V}$$

Constraints

- The end time of the activity in the agent plan at position *i* cannot be smaller than the start time of the same activity.

$$t^e_i - t^s_i > 0$$

- The sum of the costs of all the activities in the agent plan and the travels made to pass from one location to another cannot be greater than the available monetary budget.

$$\sum_{i=1}^{n} c_D(\lambda_{i,} t^s_i, t^e_i) + c_M(\mu_{i,}\lambda_{i-1}, \lambda_i) \leq B_C$$

- The cost of the time budget consumed during the entire plan cannot be greater than the time related budget.

$$c_v(t^s_0, t^e_n) \leq B_V$$

- The end time of the activity at the position *i* in the agent plan plus the travel time between the location of the activity in position *i* in the agent plan and the location of the activity in position *i+1* must be equal to the initial time of the activity in position *i+1*

$$t^e_i + \omega^{time}(\lambda_i, \lambda_{i+1}, \mu_{i,}) = t^s_{i+1} \ \forall\, i \,\epsilon \{1, \dots, n-1\}$$

3.1.5.2 Pipeline

The multimodal version of C-TAP presented in the previous section requires an extensive data set. The optimisation lifecycle of the model requires several analyses and a complex calibration process. In this chapter, we present step-by-step how to create the pipeline using the J-TAP components, which are designed to simplify the optimisation lifecycle. Figure 7 shows the main components of the pipeline.

The output of the pipeline is the optimal plan of the agent in terms of activity type, activity location, activity duration and transport modes.

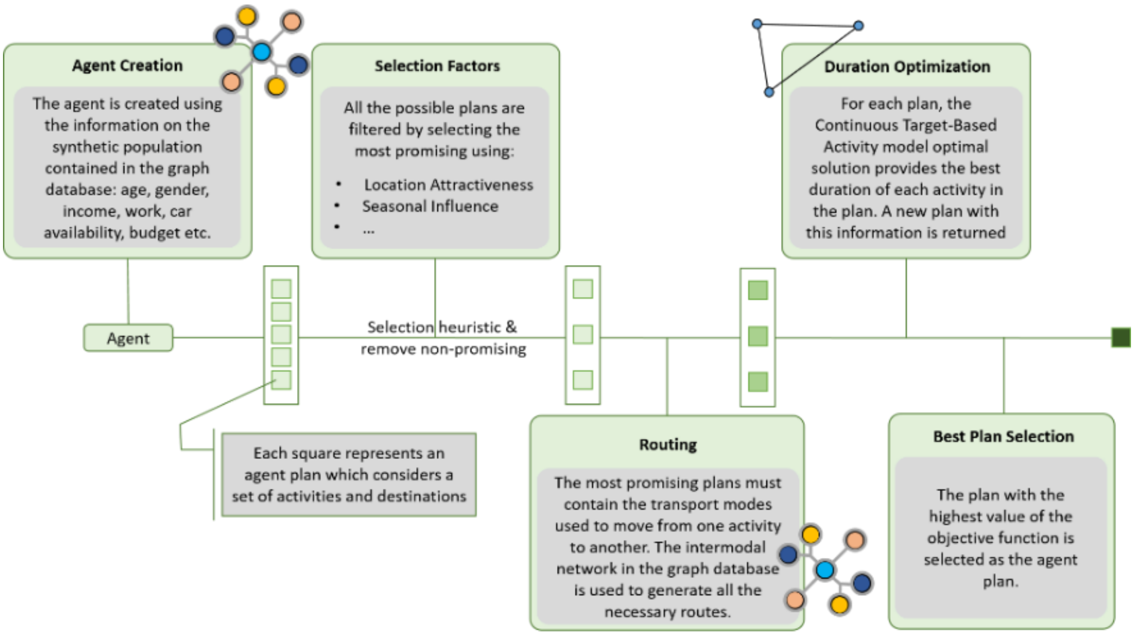

Figure 7: J-TAP pipeline

3.1.5.2.1 Agent Creation

Because C-TAP is an agent-centric model, the parameters of the model should be customised to the agent category. Age, gender, income, and the other agent properties can affect its decisions. For example, in C-TAP the location attractiveness plays a fundamental role in the agent's decisions. This can be customised using J-TAP. Moreover, the data regarding the agent can be easily retrieved from the graph database. These data are then used in a function to generate a location attractiveness tailored for the agent. Below an example of a customised location attractiveness function is shown.

$$S_{\lambda,\beta}(u) = LS_\beta(u) - C_\beta^{nf}(u) + f\big(\sigma_{\lambda,\beta}(u)\big)$$

$$\sigma_{\lambda,\beta}(u) = \sum_{i\,\epsilon\,A_{\lambda,\beta}} C_i^s(u) + \mu_{i,\beta}(u)$$

$A_{\lambda,\beta}$	Agent's activities subset in location λ
$LS_\beta(u)$	Lost salary per unit time agent category β [$/t]
$C_\beta^{nf}(u)$	Fixed living costs per unit time per agent category β [$/t]
$f\big(\sigma_{\lambda,\beta}(u)\big)$	Location combined activities attractiveness per unit time [$/t]
$\sigma_{\lambda,\beta}(u)$	Location activities attractiveness per unit time [$/t]
$C_\beta^s(u)$	The average cost of activity i per unit time
$\mu_{i,\beta}(u)$	The extra value of activity i at location λ perceived by agent category β [$/t]

Moreover, this first part involves the generation of the first set of plans (alternatives) that the agent can perform. A single plan in the set is composed of a sequence of activities and locations.

3.1.5.2.2 Selection factors

Optimising each alternative would result in an infeasible amount of time due to the enormous number of possible alternatives in terms of sequence of activities in the agent plan and possible activity locations. A selection of the most promising alternatives is necessary. This can be achieved using the agent parameters computed at the previous step of the pipeline. J-TAP provides efficient tools to generate a set of alternatives and filter them. For example, alternatives that cannot be afforded by the agent, alternatives with a low pull factor and alternatives with impossible travel modes are discarded.

3.1.5.2.3 Routing

Intermodal routing is a key aspect of this C-TAP version. C-TAP needs a set of optimal routes between the activities of the agent plan for use as inputs in the model. Optimal routes in a multimodal network can be computed using the J-TAP network package presented above. An optimal route could be mono-modal or intermodal. The routing algorithm, even simple ones such as Djikstra, can explore the full multimodal network as if it were a single network. As we have already highlighted, C-TAP is an agent-centric model, this means that also the agent transport choices could and should be customised based on an agent's characteristics. Using J-TAP, it is possible to adjust the network links weights based on the agent characteristics before computing the origin-destination (OD) matrix of the optimal intermodal routes necessary to feed C-TAP. Below the function we used for this specific pipeline to create the weights.

$$WL_{o,d,\beta,m} = \eta\left(\Delta T^{con}_{m,o,d}, \vartheta, V^{frt}_{\beta,m}\right) + \delta\left(V^{wt}_{\beta,m}, E\left[WT_{o,d,m}\right]\right) + \gamma\left(V^{tt}_{\beta,m}, E\left[TT_{o,d,m}\right]\right) + \phi\left(V^{ivt}_{\beta,m}, E\left[IVT_{o,d,m}\right]\right) + \beta\left(TK_{o,d,m}\right) + \varphi\left(EC_{m',m}, u\right) + f\left(R_{o,d,m}\right)$$

o	Departure location(node)
d	Arrival location(node)
β	Agent category
m	Transport mode
$\Delta T^{con}_{m,o,d}$	Interval of time between two consecutive connections of type m between the node o and d
$V^{frt}_{\beta,m}$	The monetary value of the perceived reliability based on the service frequency of transport mode m by the agent category β [$/t]
$\rho_{o,d,m}$	$= \begin{cases} 1 \ \ if\ exists\ a\ scheduled\ based\ transport\ service\ m\ between\ o\ and\ d \\ 0 \ \ otherwise \end{cases}$
$\chi_{o,d,m}$	$= \begin{cases} 1 \ \ if\ a\ transfer\ time\ using\ a\ transport\ mode\ m\ between\ o\ and\ d\ is \\ 0 \ \ otherwise \end{cases}$
$V^{wt}_{\beta,m}$	The monetary value of time spent by the agent category β waiting for transport mode m [$/t]
$WT_{o,d,m}$	Waiting time between o and d using transport mode m
$V^{tt}_{\beta,m}$	The monetary value of time spent by the agent category β to transfer from o to d using mode m [$/t]
$TT_{o,d,m}$	Transfer time between o and d using transport mode m
$IVT_{o,d,m}$	The in-vehicle time between o and d using transport mode m

$V_{\beta,m}^{ivt}$	The monetary value of "in-vehicle" time. Time spent by the agent category β on a transport mode m [$/t]
$TK_{o,d,m}$	Ticket price from o to d using the transport mode m [$]
$EC_{m',m}$	External intermodal costs (e.g., train station parking cost per hour) [$/t]
$R_{o,d,m}$	The monetary value of the perceived reliability not depending on the service frequency of transport mode m by the agent category β [$/t]

$$\eta\left(\Delta T_{o,d}^{con}, \vartheta, V_{\beta,m}^{frt}\right) = \begin{cases} \min\left(\frac{\Delta T_{o,d}^{con}}{2}, \vartheta\right) * V_{\beta,m}^{frt} & if\ \rho_{o,d,m} = 1 \\ 0 & otherwise \end{cases}$$

$$\delta\left(V_{\beta,m}^{wt}, E[WT_{o,d,m}]\right) = \begin{cases} V_{\beta,m}^{wt} * E[WT_{o,d,m}] & if\ \rho_{o,d,m} = 1 \\ 0 & otherwise \end{cases}$$

$$\gamma\left(V_{\beta,m}^{tt}, E[TT_{o,d,m}]\right) = \begin{cases} V_{\beta,m}^{tt} * E[TT_{o,d,m}] & if\ \chi_{o,d,m} = 1 \\ 0 & otherwise \end{cases}$$

$$\phi\left(V_{\beta,m}^{ivt}, E[IVT_{o,d,m}]\right)\ V_{\beta,m}^{ivt} * E[IVT_{o,d,m}]$$

$$\beta(TK_{o,d,m}) = \begin{cases} TK_{o,d,m} & if\ \omega_{o,d,m} = 1 \\ 0 & otherwise \end{cases}$$

$$\varphi(EC_{m',m}, u) = \int_{t1}^{t2} EC_{m',m}(u)\, du$$

This function considers several aspects we believe are crucial in the agent choice of using intermodal transport services, especially for long distance trips.

- $\eta\left(\Delta T_{o,d}^{con}, \vartheta, V_{\beta,m}^{frt}\right)$ represents the agent perception of reliability of a connection between the origin and the destination using the transport mode *m* in monetary terms. The average time between two services is used as the maximum amount of time that an agent would consider arriving at the access point prior to departure of the scheduled transport service. A high number of connections between the origin access point and the destination egress point reduce the average waiting time, increasing the reliability of the connection and thus reducing the value of the link weight.
- $\delta\left(V_{\beta,m}^{wt}, E[WT_{o,d,m}]\right)$ represents the average waiting cost incurred by the agent β at the origin for the connection between the origin and the destination with the transport mode *m.*
- $\gamma\left(V_{\beta,m}^{tt}, E[TT_{o,d,m}]\right)$ represents the average cost incurred by the agent β to transfer from the origin *o* to the destination *d* using the transport mode *m.*
- $\phi\left(V_{\beta,m}^{ivt}, E[IVT_{o,d,m}]\right)$ represents the average time cost incurred by the agent β to travel from the origin *o* to the destination *d* using the transport mode *m.*
- $\beta(TK_{o,d,m})$ represents the cost incurred by the agent β to travel from the origin *o* to the destination *d* using the transport mode *m* (e.g., ticket price).

3.1.5.2.4 Duration optimisation

This component of the pipeline exploits both the model and the solver packages to find the optimal values of the decision variables in C-TAP, returning the optimal plan of the specific agent.

3.1.6 Limitations and future work

Although J-TAP provides useful and innovative tools for the development of long-distance travel demand models, it is still in an early stage of its development. Several other features are necessary to provide a useful and simple toolkit for transport planners. The current version is intended to be the backbone for future developments in a community of academics and practitioners.

J-TAP provides a set of tools that can be used in the development of an activity-based model based on a deterministic optimisation model. However, transportation demand models can be both deterministic and stochastic (usually the nature of the agents' choices is stochastic). A well-known stochastic model is the discrete choice model, which specifies the probability that a person will choose a particular alternative. Stochastic models present several challenges, especially for LDT demand models, which are not yet tackled in J-TAP (e.g., unchosen alternatives). Stochastic models are highly data-driven and require dataset structures that are not yet implemented in J-TAP, as well as efficient algorithms to assign probabilities. Another limitation in J-TAP concerns the possibility of letting the agents interact at a very low level (e.g., traffic microsimulation). This is not strictly necessary to predict long-distance travel demand, but it becomes crucial, for example, to study the agents' interactions on the road network.

With the current version, we recommend using J-TAP exclusively to develop long-distance travel demand models based on deterministic optimisation. Future version should also explore the possibility to extend J-TAP to stochastic models.

In the next months, we will test the C-TAP model integrated in a pipeline, which was developed using J-TAP. Two case studies involving the assessment of intermodal transport modes both on the infrastructure and service side at a country level (France and Spain) will serve as a test bench for the features implemented so far.

3.2 MATSim Within-day Replanning with Mode Choice

3.2.1 Context

MATSim is an agent-based modelling framework that simulates the movements and interactions of a large number of individual travellers on a network (roads, public transit), called "agents", as they perform their daily activities. In most of its implementations, MATSim relies on a loop (see Figure 8) that cycles through the mobility simulation (Mobsim), in which the agents' movements throughout one day are simulated using a mesoscopic traffic simulation, and modules that evaluate the agents' experience (Scoring) and modules that model agents' decision-making processes (Replanning).

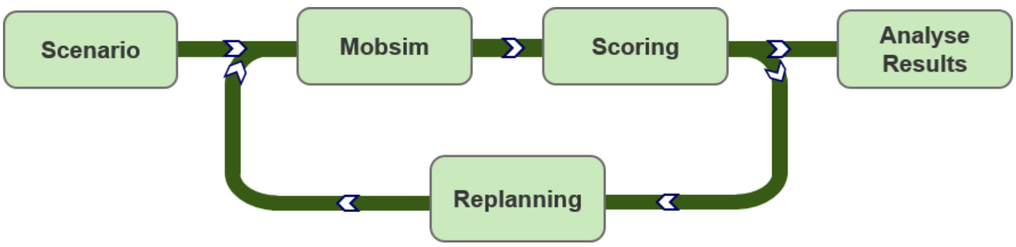

Figure 8: Traditional MATSim loop

Each completion of one cycle is called an "iteration" and can be loosely understood as one simulated day, in which the agents test their chosen plans (which include their schedule of daily activities and their planned modes of travel between their activities) and subsequently evaluate their experiences and perhaps make changes to their plans based on those experiences. In the next iteration, this process repeats. Similarly, to real persons learning how to move around a new city, the agents "learn" how to travel through the simulated city's transport infrastructure (the "network") with route and mode choices that are personally satisfactory. Thanks to the mobility simulation, the agents' travel decisions take into consideration the interactions with other agents (congestion, and in some implementations, crowding). After a sufficient number of iterations have passed, any individual agent cannot find a better way to improve the average value of its scoring function, which means that the scoring function value derives from a stationary process. This stable state is what is often called stochastic equilibrium for MATSim scenarios. Once a MATSim scenario has reached this stable state, it can be analysed.

One important detail to note is that in these standard MATSim runs (a "run" refers to running the MATSim loop through enough iterations to reach the stable state) agents do not change their plans within the mobility simulation: they only change their plans outside the mobility simulation, when the decision-making module(s) are run. During the mobility simulation, an agent will patiently wait in line to pass through a congested roadway segment and will not reroute, even if they must wait hours. If the MATSim implementation simulates delays in public transport, agents in a standard MATSim run will wait for a delayed bus indefinitely. There is also no parking search, and if taxi fleets are implemented, an agent will wait indefinitely for a taxi or be unable to complete their trip if their request is denied. This inflexibility of agents during the mobility simulation is not necessarily a problem: the situations described above are penalised heavily in the modules evaluating agents' experiences and thus agents are very unlikely to continue to use a plan that consistently puts them in the described situations. By the time the stable state has been reached, very few agents on average, if any, will be still performing plans which such illogical behaviour. Thus, for studies concerned with local daily travel behaviour under typical network conditions, the standard MATSim loop is sufficient.

However, agent-based modelling frameworks such as MATSim have the potential to simulate non-typical conditions and dynamic agent behaviour. Indeed, this is one of the more interesting potential use cases for such models. In order to harness this potential, agents must be able to make decisions within the mobility simulation: they have to be able to make decisions "within the simulated day". In the MATSim community, this is known as "within-day replanning" (WDR).

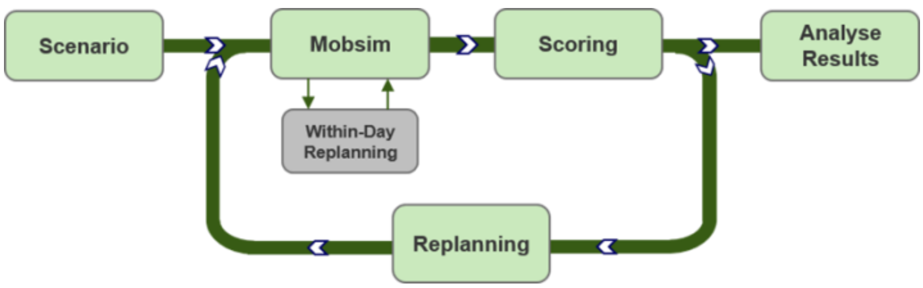

Figure 9: The traditional MATSim Loop with Within-day Replanning (WDR)

The desire to be able to simulate dynamic agent decision making within the mobility simulation is not new. Indeed, there are WDR modules within the core of MATSim (the "core" is composed of the stable, "standard" elements of the very large, constantly changing, open-source code base that composes MATSim). This "core" WDR code includes several basic modules that theoretically enable the developer to implement different kinds of passenger information systems, such as radio broadcasts during an emergency or roadway signs warning of congestion ahead. However, this core WDR code has largely been used in the past to enable road network rerouting decisions for agents driving motor vehicles during exceptional events, such roadway closures or evacuation procedures. It does not include any "replanners" that enable agents to reroute through a transit system, nor does it include any replanners that would enable agents to perform mode choice during the simulation.

Recently, several researchers have worked on creating a new WDR implementation that allows agents to reroute through a transit system [27] [28]. They also worked on implementing "agent selectors" which could theoretically simulate how different kinds of passenger information systems deliver information to passengers, allowing passengers to react to this information within the mobility simulation. However, as of the writing of this report, this code has not been added to the MATSim GitHub repository and is only available upon request from the researchers themselves. It has not yet been adjusted to be compatible with the current versions of MATSim, either. This makes it very difficult to use and to integrate with the current version of MATSim. These researchers also did not implement any replanners that enable within-day mode choice decisions to be made: their simulations only allowed agents already using transit to re-route their transit journey: agents could not decide to abandon their planned transit route and instead take a taxi or utilise shared mobility options, such as bike or e-scooter sharing. Thus, for the larger MATSim community, the problem of re-routing transit agents during the mobility-simulation remains unsolved. This is one of the gaps that TRANSIT is attempting to address.

Furthermore, one very large gap in the MATSim framework preventing a fuller, more realistic spectrum of passenger reactions to passenger information systems from being modelled is the lack of any code enabling within-day mode choice decisions. This gap is largely due to how the "traditional" evaluation and decision-making modules in MATSim work.

In the "traditional" MATSim implementations, agents' daily plans are scored based upon their experiences within the mobility simulation. This scoring process utilises, in one form or another, the concept of econometric utility. Depending on which scoring modules are used, agents will evaluate the econometric utility of their performed trips and also their performed activities. Which parameter values are assigned to the attributes of the trips (such as travel time and comfort) and activities (the necessity of working or the pleasure of leisure)[2] is case-study specific and entirely up to the modeller. Generally, the starting values of the attributes' parameters for trip attributes are based on the values estimated using traditional choice models, and then calibrated such that the MATSim runs can reasonably reproduce real observations from the system[3]. However, the sheer number of attributes

[2] Although the types of attributes that are relevant for calculating the econometric utility of transport options are fairly well known, as are reasonable values for the attributes' associated parameters, the attributes and parameters for activities are not. Thus, even though MATSim can – from a software point of view - endogenously allow agents to change the starting and ending times of their activities, it does not mean that a modeller should use this feature. Properly addressing this research gap would require extensive behavioural research and is thus out of the scope of this work package as well as the TRANSIT project as a whole.

[3] This practice is also methodologically problematic, since discrete choice models are probabilistic, whereas the utility equations in MATSim scoring are (largely) deterministic. Again, there are many implementations of MATSim, some of which may have attempted to address this inconsistency. However, in this report only the core, standard usage is being addressed, since this is what most potential users will encounter first.

typically incorporated into the scoring process, and thus the large number of necessary parameters, makes traditional MATSim implementations difficult to calibrate. Furthermore, the plans are scored as a whole, meaning that the standard scoring process is not designed nor calibrated to evaluate a single trip without the context of the entire plan. This means that is not methodologically consistent to use the standard scoring modules within the mobility simulation to evaluate single or even partial trips. The previously mentioned implementations of WDR largely avoided this problem by simply re-routing agents using the current network conditions and finding the least-cost path (which is often, but not always, the shortest path), to minimise their delay, instead of allowing for more complex decisions, like mode choice.

The difficulties, both practical and methodological, of using the traditional MATSim scoring motivated several researchers working with MATSim to replace these scoring modules with discrete choice models [29] [30]. Discrete choice models are a well-established type of models used to predict and understand decision-making behaviour both in transportation and beyond. In transport planning and modelling, discrete choice models are generally formulated to evaluate the available options for individual trips. This makes the Discrete Mode Choice Extension of MATSim and the associated eqasim framework [31] good candidates for combining with the WDR code in the MATSim core to enable within-day decision-making that is consistent with the whole loop. The eqasim framework is a new modelling framework that incorporates discrete choice models into MATSim (see Figure 11). Thus, this was the approach chosen for modifying MATSim to be able to model the innovative coordination mechanisms involving passenger information systems and providing a “sand box” to test dynamic operational optimisation tools, such as the AMAN/DMAN Ground Tool (ADGT) described in section 3.3.

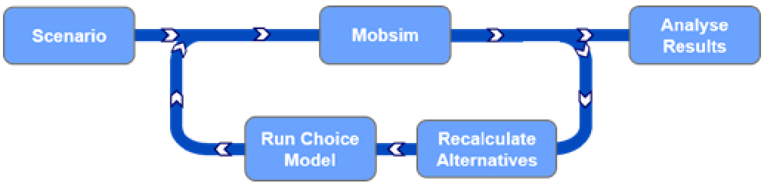

Figure 10: The eqasim loop

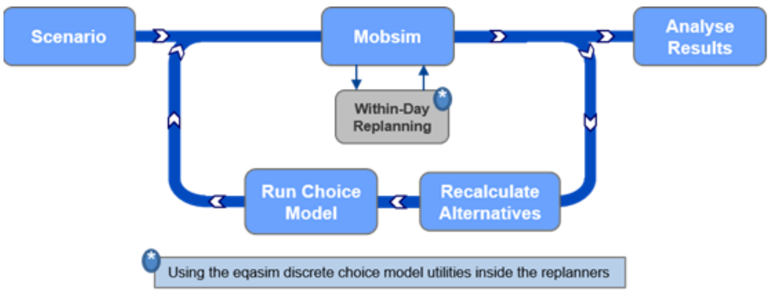

Figure 11: The eqasim loop with WDR using discrete mode choice models

Lastly, in reviewing the existing code and literature available on the subjects of using agent-based simulations to (a) model dynamic (within-day) agent decision making within the mobility simulation and (b) passenger information systems and their effect on passenger experiences and network states, it became clear that there is a distinct need for precise language to describe *how* passenger information systems and traveller’s behaviour in response to these systems could be and is being modelled. Thus, the task of evolving the MATSim framework also aimed to address this gap.

3.2.2 Research motivations

Many of the potential innovative coordination mechanisms proposed in deliverable D2.1 involve highly dynamic passenger information systems, which either provide live information on passenger movements to system operators, provide live service status information to passengers, or both. This reflects a general trend not just in managing the interface between long distance passenger transport services and local passenger transport services, but in all passenger transport services (public transport, intelligent driver information systems, ride sharing, bike sharing, automated taxi services, etc.). Thus, being able to model realistic passenger responses to various passenger information systems and information distribution strategies dynamically is a broadly relevant research goal. It is also a challenging undertaking for which agent-based modelling frameworks, such as MATSim, are well suited. The implementation of such functionalities faces challenges both from a technical point of view and from a communicative point of view.

The communicative (specification) challenge is the lack of a standard terminology for describing the various types of passenger information systems and information types and when, where, and how passengers receive it in a manner that model developers can consistently – and without critical misunderstandings - translate into specific programming tasks. Such standard terminology would also ensure that descriptions of what a model can and cannot do are consistent across different models and projects, ensuring that the necessary precision in communicating exactly what a model does and does not do. Such precise communication is a key element for:

1. correctly interpreting model results,
2. deciding on which model to use for a specific question or to model a specific passenger information system or strategy, and
3. ensuring an effective scientific peer review of work done with such models is not entirely dependent on the peers having an intimate, code-level understanding of the specific model used in a specific paper.

To address this gap, we have proposed a standardised linguistic framework.

The technical challenges include:

- long computation times due to more and more complex processes being run during the mobility simulation;
- formulating a reasonably realistic abstraction of real behaviour;
- maintaining consistency between how decisions are modelled between iterations and within the mobility simulation, yet allowing for the flexibility to allow these two processes to be modelled differently should the results of behavioural science/validation procedures demand this;
- keeping the model agile, to allow as-of-yet not-considered or foreseen types of behaviour and information systems to be modelled;
- ensuring the existing community of MATSim users can easily understand and use the new modifications;
- working with existing and often unwieldly open-source code bases;
- gathering data that is spatially and temporally precise enough to enable the validation and calibration of such models.

The MATSim modifications aim to fill these gaps by:

- taking advantage of already parallelised code available in the MATSim core package called “Within-Day Replanning” as well as the eqasim framework, which requires fewer iterations to reach a stable state than the traditional, standard MATSim loop;
- keeping replanning logic simple for the moment, but allow for future complexity;
- using same discrete mode choice utilities for both the decision-making process within the mobility simulation and the one between iterations, but making sure to take advantage of the inversion of control to allow future developers to more easily use different ones in the future;
- taking advantage, insofar possible, of inversion of control, which allows for more agile programming and modular design of the code;
- building upon well-established elements of the MATSim core and the new eqasim framework

3.2.3 Intended use cases

The modifications of the MATSim Within-Day Replanning code are intended to allow for the simulation of effects of passenger information systems on how agents react to disruptions.

3.2.4 The linguistic framework in brief

Modelling passenger information systems requires being able to precisely describe not only the passenger information systems themselves, but also how passengers react to that information. In support of the evolution of the MATSim framework, the initial literature review presented in D2.1 was expanded and synthesised into a collection of terms.

3.2.4.1 Describing the passenger journey

During the literature review, a particularly useful dissertation entirely focused on passenger information systems for public transit systems was found [32]. For TRANSIT, the most useful aspect was the development of a set of terms that precisely and flexibly describes the passenger journey. Hörold’s terms were translated from the original German and slightly adjusted to make them somewhat more universal. The result is the “trip timeline” and the “trip locations” (see Figure 12). These can be used to describe travel with any mode (e.g., for a car, the access point is where the car is parked). Especially in dense cities, this is not always the starting point of a journey.

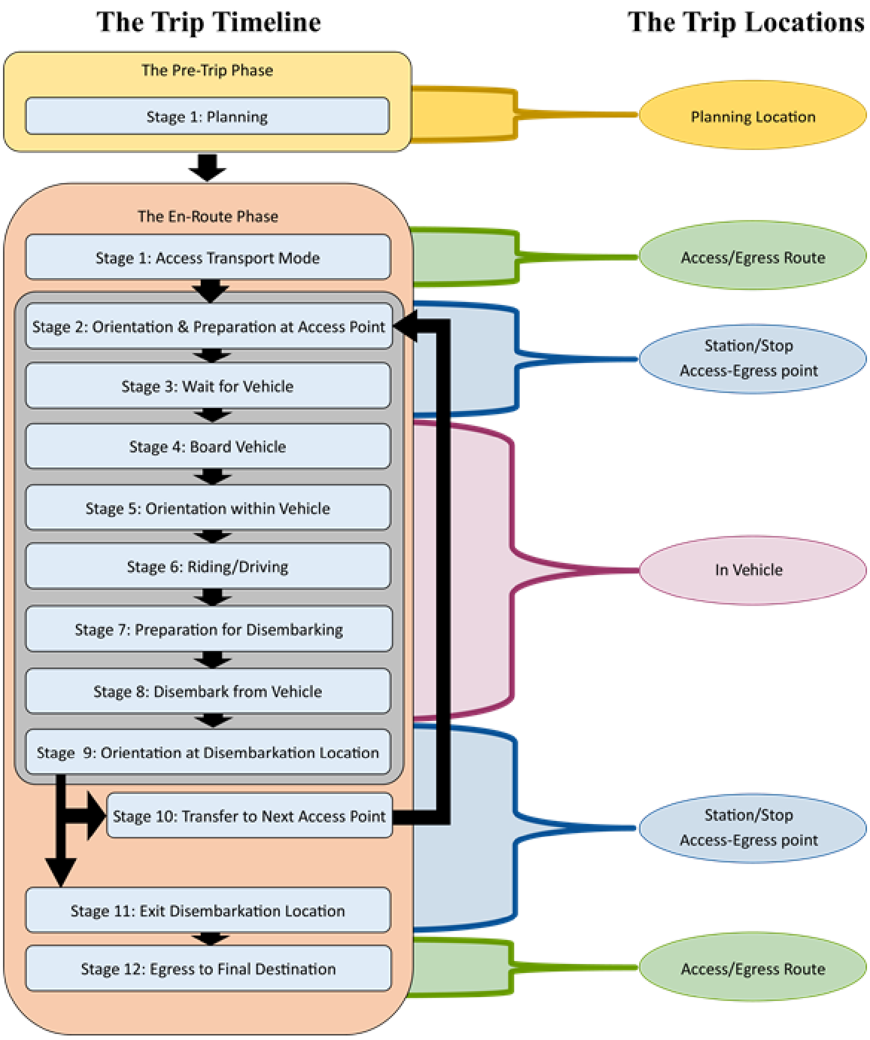

Figure 12: The Trip Timeline and the Trip Locations

3.2.4.2 Categorising the information provided

Also, from [32] comes a useful categorisation of passenger information types. This has again been translated from the original German and slightly modified to make it more universal and clearer for specifying these information types in models.

- **Spatial information**, such as maps or route instructions
- **Time-based information**, such as timetables or travel time estimates
- **Ticketing & Cost information**, such as booking systems or tariff overviews
- **Disruption information**, such as delay notifications, congestion alerts, etc.
- **Advertisement** of events and offers
- **Supplemental information**, such as crowding forecasts, seating, and on-board amenities.

3.2.4.3 Describing how passengers perceive information

Passenger information services exist to help passenger navigate through their journey, or inform them of goods and services. When attempting to model how passengers react to such information, it is necessary to describe the information in terms that refer to how passengers perceive that information and process that information. [33] provided many very useful concepts for this. The following terms are either copied or paraphrased from [33], with a few new terms being added to allow modellers to more precisely describe what their models model.

- **Historical information:** "Information which describes the state of the transportation system during previous time periods"
- **Current or real-time information:** "the most up-to-date information about current traffic conditions"
- **Predictive information:** "information concerning expected traffic conditions during subsequent time periods when travel can occur"
- **Experiential information:** a person's memory of past experiences
- **Exogenous information:** information received from external sources
- **Perception or information level:** the combination of experiential and exogenous information
- **Updating perceptions:** how a person combines their experiential information with exogenous information to form their perception of the system.
- **Experiential uncertainty:** experiential information about past disruptions and delays, their frequency, duration, etc.
- **Exogenous uncertainty:** exogenous information that communicates uncertainty, such as a range of possible durations for a disruption

In addition, these modifiers are needed, especially for modelling:

- **Planned:** refers to exogenous information about planned (or assumed, default) states, such as scheduled time-tables or travel times calculated on uncongested networks.
- **Realised:** refers to exogenous historical information on observed time-tables (so records of when vehicles actually arrived or departed, etc.) and observed travel times.
- **Static:** refers to a one-time, real-time information access (e.g., a person checks real-time congestion information before leaving work, but does not have access to updated information while en-route, decisions can only be based upon the initial request for real-time information).
- **Continuous:** refers to constant access to real-time information (e.g., a person riding a bus has a smartphone app that continuously receives updates on all delays in the transit system. This person can always check the current network state to see if they can still make their planned connection, and if not, which other connections might still be possible).

3.2.5 Design

MATSim is an open-source, agent-based transport simulation [22] that was designed to be a modular piece of software. With the publication of Google Guice for Java, MATSim was refactored to take advantage of inversion of control. This makes mixing and matching default and customised functionalities much easier, especially in the open-source, multi-developer environment, but does make the code more difficult to read. Thus, a new user is strongly advised to make sure they understand inversion of control and have at least an introductory knowledge of Google Guice when modelling with or developing new functionalities for MATSim.

MATSim is largely driven by academic researchers. This means that although the “core” is stable and consistent, there are many extensions and customisations that have been independently developed by different teams around the world for different purposes and with different goals. Thus, not all MATSim implementations can do the exact same things, or even do the same things in the same way. Neither can it be assumed that a particular extension will work “out of the box” with another extension. Thus, a new user or a reader of MATSim papers is strongly advised to find out exactly which version of MATSim and which extensions were used for a particular study, and how these were implemented specifically. New users should additionally check to see if the desired extension has been added to the official MATSim repository, which is largely maintained by the team of Prof. Dr. Kai Nagel at the TU Berlin, with occasional support from Prof. Dr. Kay W. Axhausen’s team at the ETH Zurich.

3.2.5.1 The eqasim’s design

Prompted by the fact that MATSim’s co-evolutionary algorithm relies on scoring not just the agent’s trips but also all of their activities and that, at the time, the empirical, quantitative behavioural science on estimating the disutility of travel vs the utility of activities was not advanced enough to properly and robustly inform the parameters of the utility functions used for scoring, two then-PhD students at the ETH Zurich developed the Discrete Mode Choice Extension [30] and the framework now known as “eqasim” [31]. As mentioned, eqasim is a framework. Firstly, it includes a “pipeline” written in Python that structures the process of converting raw data into a MATSim scenario. Secondly, it contains an “ecosystem” of classes integrated with MATSim that make using the Discrete Mode Choice extension easier and more structured and modular. Thirdly, it contains a collection of analyses written partially in Java using MATSim classes (a sort of “pre-processing” of the standard MATSim outputs) and partially written in Python (post-processing and analysis).

Eqasim is also explicitly open source. An entirely open-source example scenario of the Ile de France region, based upon open-source data provided by the French government, is available on GitHub.

Eqasim embraces inversion of control, and thus utilises the customised MATSim Guice functionalities.

3.2.5.2 The WDR design

The Within-day Replanning extension was developed by [34] and eventually incorporated into the MATSim core. Dobler, in the MATSim spirit, designed WDR to be modular. Indeed, the structure is fairly straightforward. Thanks to the dependency injection still used within most of the WDR code, a UML diagram can be generated and makes how WDR classes are related fairly easy to follow.

The central functionality of WDR is to allow agents to modify their plans during the mobility simulation (mobsim). This means that the objects and methods within WDR must be able to interact with the mobility simulation (QSim, in most MATSim implementations). WDR must also provide ways to find the agents to replan and also the methods to replan those agents as required by the research goal of the modelling effort. Several of the most relevant classes/categories of classes that enable this functionality are the following:

- The WDR Controller brings together all the WDR elements. This is currently where Agent Filters and Selectors are combined with Replanners to specify which agents should be replanned and how they should be replanned.
- The WithinDayEngine manages and executes most of the WDR elements during the mobility simulation (in most MATSim implementations, this is QSim). It gets called by QSim before any other simulation engines at the beginning of each time step, ensuring that any specified agents are replanned before the other elements of QSim (moving agents between links, ending activities, etc.) are executed.
- The Listeners interact with QSim to keep track of the states of agents and the network travel times step by time step. These have turned out to contain several bugs, which will need to be addressed in the future.
- The Agent Filters and Selectors (sometimes named “Identifiers” in the code due to incomplete refactoring) utilise the Listeners and customisable logic to find the agents that will be allowed to replan.
- The Replanners are the classes containing the logic by which the agent’s plans are modified. These might also call upon Listeners and other classes for information about the network state and model settings.
- The Withinday Utils (Utilities) are the classes containing methods that are useful for modifying agent’s plans. This is the part of the WDR code that will be most familiar to most MATSim developers, as the methods contained within have often been used for other types of within-day replanning, like dispatching taxis.

The biggest limitations of the WDR code within the MATSim core is that it does not provide any tools for modifying transit trips (transit trips are stored in a different format than network trips). Nor does it provide any tools for allowing agents to perform mode choice (all of the “default” replanners focus on re-routing an agent performing a trip with a network mode, like car).

3.2.5.2.1 The WDR Controller

The WDR controller is perhaps misnamed, since it does not really resemble the MATSim controller. Instead, it contains methods for configuring the MATSim controller and methods for the lazy initialisation of agent filter, selector, and replanner factories and adding these to the WithinDayEngine so the desired agent filters, agent selectors, and replanners can be instantiated as needed during the mobility simulation. The methods within the WDR Controller that configure the MATSim controller mostly ensure that the WDR Controller is called at the right times and the routers evaluate route costs in the desired way (this is separate and distinct from how agents evaluate the utility of routed alternatives via the discrete mode choice model).

The WDR Controller is one of the few classes of the core WDR code that interacts with MATSim. Here, Guice injection can be successfully used. It is from the WDR Controller that many of the objects WDR agent selector factories and replanner factories require must be passed to said agent selectors and replanners using traditional constructors.

3.2.5.2.2 The WithinDayEngine

The WithinDayEngine manages how the agent filters, agent selectors, and replanners are used during the simulation. For instance, it is the class which contains the "doSimStep" method that checks to see if there are any replanners that need to be run, and then runs the needed replanners. It is also the class which manages the classes which enable parallel processing of replanning tasks.

The WithinDayEngine is another "contact point" for Guice. The WithinDayEngine constructor receives its arguments through the @Inject command of Guice.

3.2.5.2.3 The WDR Listeners

To enable the retrieval of agents currently being simulated and other information from the simulation itself, WDR provides several "listener" classes. Note for readers already familiar with MATSim: these WDR "listeners" also implement various event handlers' interfaces. The Listeners are thus able to "listen" to the execution of the classes inside the MATSim core which do the actual simulation tasks, such as the various QSim engines and the EventManger.

The WithinDayControllerListner class is called at the start of the simulation (as it implements the class StartupListener) and creates and initialises the other, more specific listeners. Those listeners that are of most interest for readers of this this text are:

- the MobsimDataProvider, which allows other WDR classes to retrieve information from QSim (the class which runs the simulation), such as the agents being simulated;
- the ActivityReplanningMap, which keeps track of which agents are performing activities and what the end times of those activities are, and provides methods for accessing these agents. The methods in the class are largely reliable, although there is one identified bug. The way it is written means that if an agent starts and ends an activity in the same time step – as is often the case if an agent is running late due to a disruption – this agent is neither registered as performing an activity nor as about to end an activity. Thus, agent selectors looking for agents who are ending their activity will not be able to find agents who start and end an activity in the same time step. This bug will be addressed during the implementation of the developed models in the case studies, in a later stage of the project;
- the LinkReplanningMap, which keeps track of agents which are performing a leg (a trip) and provides methods for accessing these agents.

In essence, the Listeners act as a pre-selector of agents by keeping track of which agents are about to end activities and which agents are performing a trip. Depending on where on the trip timeline a passenger should retrieve information, other listeners could be implemented. For instance, if a passenger has no information, they would only learn of a transit disruption when their planned vehicle does not arrive. Thus, a Listener that keeps track of agents who are stuck or who have been denied boarding could be implemented to find agents who are waiting for a vehicle that is not coming or agents who are driving and stuck in traffic.

3.2.5.2.4 Agent Selectors and Agent Filters

Agent Selectors combine information from the Listeners and the Agent Filters to select those agents that should replan at the beginning of each simulation time step. The Agent Filters are the classes which filter agents tracked by the Listeners. For instance, while the LinkReplanningMap keeps track of which agents are currently performing a trip, an agent filter can search this list of agents for agents who have a certain attribute, are performing a specific type of activity, plan to use a certain mode, etc.

3.2.5.2.5 Replanners

Replanners contain the methods which alter an agent's plan. Replanners alter the plan that the agent is currently executing, and must return it in a form that can be interpreted by the mobility simulation. It is by developing a new replanner that a modeller can simulate different kinds of decisions (re-route, change mode, etc.) that an agent should perform when notified of a disruption.

Each replanner specifies which Agent Selectors it uses to find agents with. By specifying which Agent Selectors a replanner should use to find agents to replan, it is possible to specify which groups of agents – and when, and where – should make certain kinds of decisions.

For the TRANSIT project, two new replanners have been developed so far to allow agents ending an activity and agents performing a trip to perform mode choice instead of just re-routing their trip with their planned mode.

3.2.5.2.6 WithinDay Utils (Utilities)

This category of classes contains methods that are able to splice new legs or activities into plans. However, these methods did not allow for enough customisation. For this reason, new methods have been implemented to take direct advantage of utilities within the MATSim core, such as those contained within TripRouter class, to splice the replanned elements of an agent's plan back into the entire plan. A particular challenge is to splice the elements of a partially executed and partially replanned leg in MATSim, back into the original leg in the original plan. This is particularly tricky for transit legs, since they are not structured in the same way as network legs, such as a route performed with a car. The last details of developing this new "transit leg splicer" are still in progress and will be finished early in the implementation of the case studies.

It is the Replanners that make use of the methods contained within the WithinDay Utils, since the Replanners must return the modified agent plan to the mobility simulation.

3.2.5.2.7 How these classes work together

Figure 13 illustrates how the WithinDayEngine, Replanners, Agent Selectors, Agent Filters, and the Listeners work together to replan agents during a disruption. This figure is a schematic example: the actual number of each type of class will need to be higher to reflect the different passenger information systems used during a disruption and the different options an agent has depending on where they are in their plan and which information they have available and when.

The following process illustrates how the described classes work together to allow agents to replan during the mobility simulation. At the start of each time step:

1. The WithindayEngine checks to see which replanners are active. Replanners can have limited windows of simulation time for which they are active. This is one way to ensure agents only replan in a certain way at a certain time.
2. The WithindayEngine then passes the active replanners to the ParallelReplanner class, which is the class that manages the parallelisation of replanning tasks.
3. The ParallelReplanner does the following for each Replanner (actually each Replanner Factory):
    a. Retrieve the Agent Selectors assigned to the Replanner
    b. Run each Agent Selector to retrieve the agents that need to replan
    c. Add each agent that needs to be replanned using the Replanner as a Replanning Task to a container of Replanning Tasks within the ParallelReplanner.

4. The ParaelleReplanner then executes its method "run"
   a. First, the Replanning Tasks get assigned to a processing thread
   b. Within each thread, replanning tasks are run using the "doReplanning" method of the replanner associated with the replanning task
   c. The success or failure of the replanning task is noted in the log file

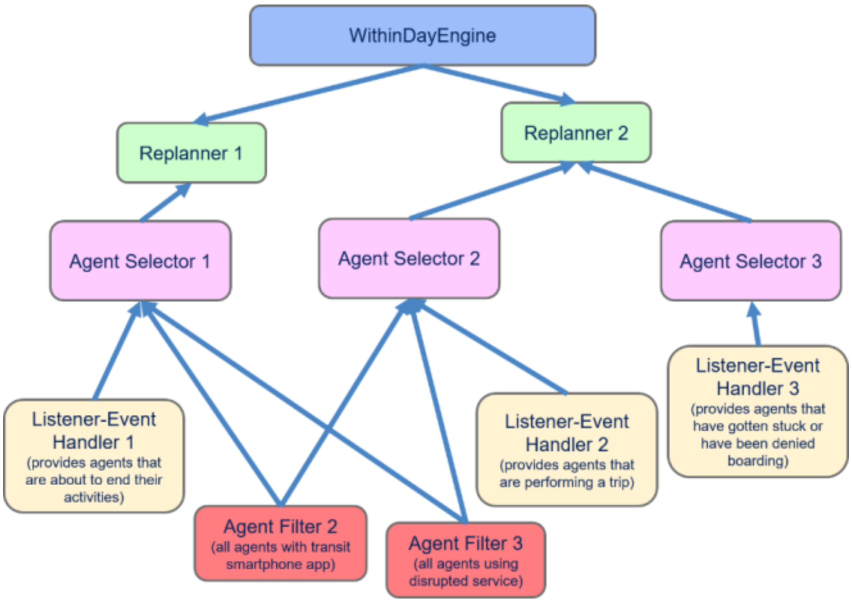

Figure 13: The structure of how the WithinDayEngine, Replanners, Agent Selectors, Listener-Event Handlers, and Agent Filters work together.

3.2.5.3 The combined design

When MATSim was revised to take advantage of inversion of control using Google Guice, the WDR code was only partially updated. This resulted in the WDR code still using standard dependency injection, which makes it less easy to modify or extend than the now "Guice-fied" MATSim and eqasim. This lack of inversion of control in WDR, combined with eqasim's completely "Guice-y" design, made combining the two codebases a relatively challenging task that was not guaranteed to succeed.

This rather awkward mix of different methods of handling dependencies was addressed by using the "Guice access points" within the WDR code. The relevant objects bound by eqasim – such as the utility calculators, choice models, etc. – were injected into the Withinday Controller and then passed on as arguments to Replanners. It was not clear if this would work, but so far it has not produced any problems.

The next hurdle was that neither the eqasim methods for applying the discrete mode choice models and utility functions to agent plans nor the WithinDay Utils had the necessary granularity needed for the Replanners to enable mode choice within the simulation and divorced from the context of modifying plans outside of the simulation. Thus, new methods were created within the new Replanners to directly access individual methods from eqasim and other parts of MATSim to generate new alternatives (including re-routing the planned or current mode), estimate the utility of each alternative, apply a choice model, and splice the new plan elements back into the agent's plan.

The new Agent Selectors follow the WDR structure (so there are Selectors for agents finishing activities and selectors for agents performing legs). They currently hard-code the list of agents to be replanned. This is a beta-version of the eventual goal of having a set of classes that will import a list of agent ids from a csv file, allowing users to easily specific agent lists – such as those agents that have downloaded a passenger information app, have purchased integrated tickets, or are booked to fly with a particular flight, etc.

To model a disruption, the process illustrated in Figure 14 must be used. First, an undisrupted scenario needs to be run until it reaches stochastic equilibrium, or is "relaxed". Then, the disturbance must be expressed as either a modified network (such as reducing the link capacity of a roadway link to represent a partial lane closure) or a modified transit schedule (such as removing the tours of transit vehicles during a time window during which that particular transit line is not running due to a disruption). Then, one single iteration with the disrupted network and/or transit schedule is run, using the agents with their plans from the relaxed scenario and with Within-Day Replanning active. Afterwards, the results of this one-iteration run can be analysed.

Ideally, to test the robustness of the effects of a particular passenger information strategy for disruption mitigation, many different one-iteration runs using slightly different disrupted networks and/or transit schedule should be performed.

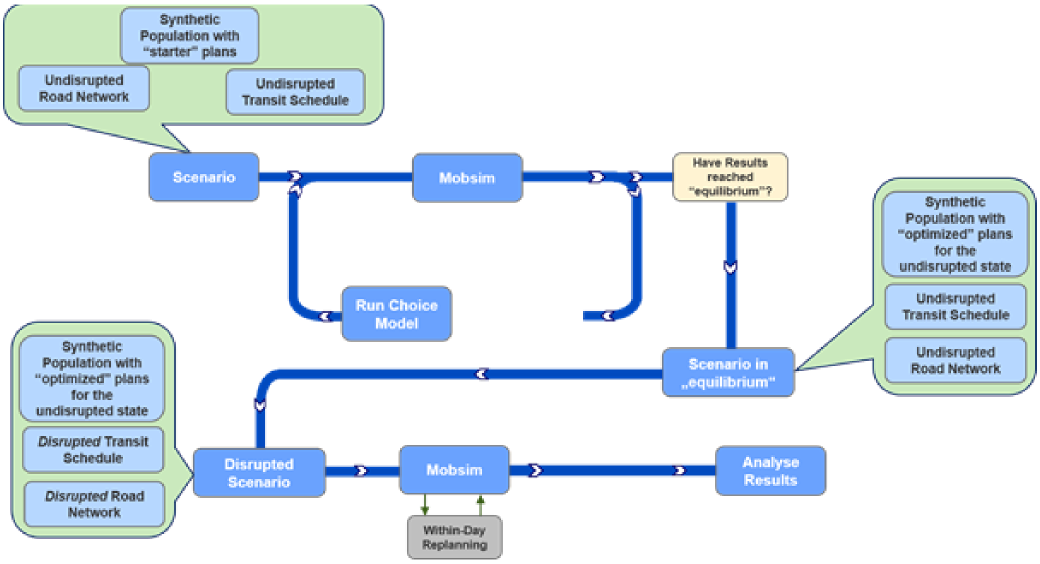

Figure 14: MATSim workflow for modelling a disruption as implemented in eqasim and with WDR

3.2.6 Limitations

The main limitation of the modified WDR code is that it does not alter the activities of agents. This means it cannot model how an agent would decide to leave work early if their planned mode has predicted delays, nor how an agent would adjust its activities following a disrupted leg that has made them late. For example, in the latter case, the agent would simply start performing the activity when they arrive late but then end the activity as planned: the agent would not, for instance, stay late a work to make up for lost time. Addressing this gap properly involves behavioural research, and is outside of the scope of the TRANSIT project.

3.2.7 Future improvements

In the implementation of the models for the case studies, the output of J-TAP – so the number and type of agents travelling in and out of the regions simulated by MATSim – will be utilised to generate "tourist" agents in the MATSim scenarios. These tourist agents will be the ones who have integrated tickets, need to catch flights, etc.

Alternatively to using the J-TAP output, data on the number of flights in and out of the region's airports on a typical weekday (which is the kind of day a MATSim scenario represents), the percentage of booked seats on the flights, and statistics on the characteristics of airline customers will be used to generate the tourist agents.

3.2.8 Concluding comments

The modifications to the Within-Day Replanning code and the descriptions of the relevant elements of said code in this report, combined with the community-maintained technical documentation on GitHub should allow future model developers to model agents' reactions to passenger information systems during disruptions. As a result, the output of these MATSim simulations can be used to test the effectiveness of the DMAN/AMAN Ground Tool, described in Section 3.3.

3.3 AMAN/DMAN Ground Tool (ADGT)

This subsection presents a detailed description of the coordination mechanism 'DMAN/AMAN optimisation' previously introduced in D2.1. This tactical mechanism aims at regulating the aircraft flow at the airport to manage the impact of a ground disruption on passengers' journey.

3.3.1 Context

Li et al. [35] present different incentives around the world on air-rail collaborations. They highlight the existence of different levels of partnership. For instance, AIRail [36] is one of the best intermodal collaboration examples between an airline (Lufthansa) and railway operator (Deutsche Bahn) at Frankfurt airport. The High-Speed Rail (HSR) serves cities such as Cologne, Dusseldorf and Stuttgart. Passengers can check-in their luggage and receive their boarding pass for their flight directly at the train station. Connections between HSR and flights with AIRail are like connecting flights. A lower level of partnership exists for instance between Air France and SNCF [37]. Tickets are booked at the same time, but passengers need to check-in at the airport and no coordination is implemented in case of disruption. If an arriving flight is delayed, passengers might miss their connection and will have to wait for the next train. Rail & Fly in Germany is also a kind of low-level air-rail collaboration ([38], [39]). This partnership involved the Deutsche Bahn and several airlines such as TAP, Lufthansa, or Singapore Airlines. Passengers can book a train ticket when they are purchasing their flight ticket. This train ticket enables passengers to take the train they want during the day of their flight. However, passengers are held responsible in case of missed connections contrary to AIRail collaboration.

Only bi-lateral partnerships between an airline and a train operator have been developed. No global coordination at the airport level has been implemented. Moreover, most of the actual air-rail collaborations do not include coordination of transportation means in case of a delay occurring on the first leg. This lack of coordination might induce missed connections and stranded passengers. Such passengers must be assigned to a new flight/train. This solution is not optimal both for stranded

EUROPEAN PARTNERSHIP

Co-funded by the European Union

passengers, who experience a large delay, and for transportation suppliers, which must reallocate these passengers and even pay additional costs (such as overnight accommodations).

On the contrary, when a massive disruption occurs on the ground side (bomb threat, major road traffic accident, etc.), several airports such as CDG airport have implemented a hard waiting policy to minimise the number of stranded passengers. The latter consists in delaying all departing flights at the gate to wait for delayed passengers. However, such program induces massive congestion at the airport and may disturb the overall air traffic flow management. A ground delay program can be applied in combination with this waiting policy to mitigate congestion by delaying arriving flights directly at their departure airports.

The ADGT finds a compromise between no coordination and hard waiting policies. It would act as an interface between the overall ground transportation system, which provides an access to the airport and the air transportation system. A new flight schedule is generated every time the passengers' location information is updated during the day. This tool aims at finding a balance between the number of stranded passengers and the overall flight delay.

3.3.2 Description of the tool

A coordination mechanism between the Air Traffic Management (ATM) and Ground Transportation Suppliers (GTS) is proposed in case of a disruption on the ground side. A collaboration based on information sharing between ground and air transportation stakeholders is considered in this work. When a disruption occurs on one of the airport access modes (such as a train or a subway shutdown), several outbound passengers are likely to miss their flights. Assume a passenger information service allowing passengers to declare themselves as delayed to GTS. Then the Airport Operation Centre (AOC) would have some knowledge on passengers' status (on-time, delayed) thanks to information sharing with GTS. The AOC would be able to delay specific flights to help passengers affected by the disruption to catch their flights. Assigning pushbacks (i.e., a delay at the gate) to these targeted flights would be done thanks to the Departure Manager (DMAN) tool. However, holding several aircraft at the gate is likely to induce congestion on the airport's airside. Thus, arriving flights need to be regulated through the Arrival Manager (AMAN) tool to mitigate the airport congestion. Information on air passengers would be valuable to not induce missed air connections while regulating the arriving aircraft flow. Figure 15 illustrates the different steps of this coordination mechanism.

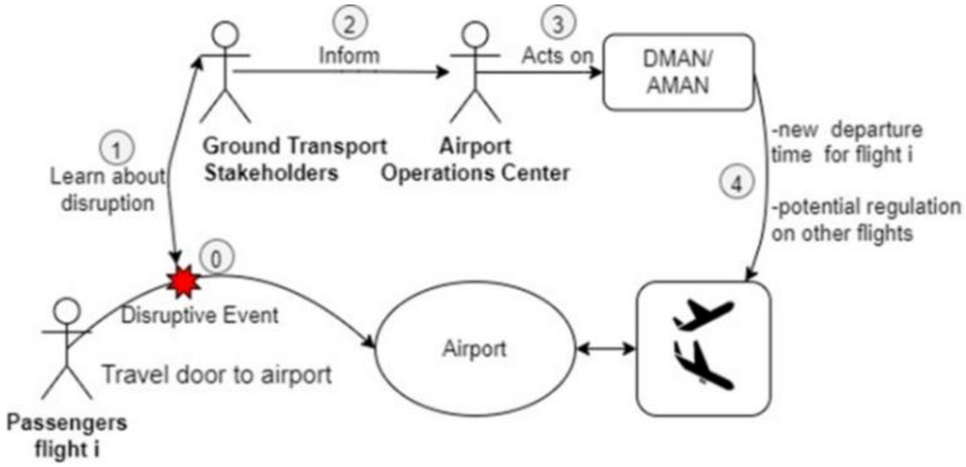

Figure 15: Illustration of the different steps of the AMAN/DMAN coordination mechanism, ADGT

Let us assume a disruptive event such as a shutdown on a subway which goes to the airport (step 0 in Figure 15). Outbound passengers who rely on the subway to reach the airport are likely to miss their flight. If the subway operators know which passengers are impacted by this disruption (step 1), this information could be communicated to the AOC (step 2). Thus, a set of flights can be targeted to wait for passengers affected by the disruption (step 3). At the same time, decisions on other departing and arriving flights need to be taken to mitigate airport congestion. The DMAN and AMAN tools can assign pushback to the targeted departing flights and air-speed regulation or delay to arriving flights respectively (step 4).

The presented mechanism proposes an extension of the coordinated AMAN/DMAN tool using passengers' location information on the ground side. Hereinafter, this mechanism will be referred as AMAN/DMAN Ground Tool (ADGT). In this work, the airport is seen at macroscopic level. Several constraints on the airside must be considered to implement the ADGT. Figure 16 illustrates the airport model retained.

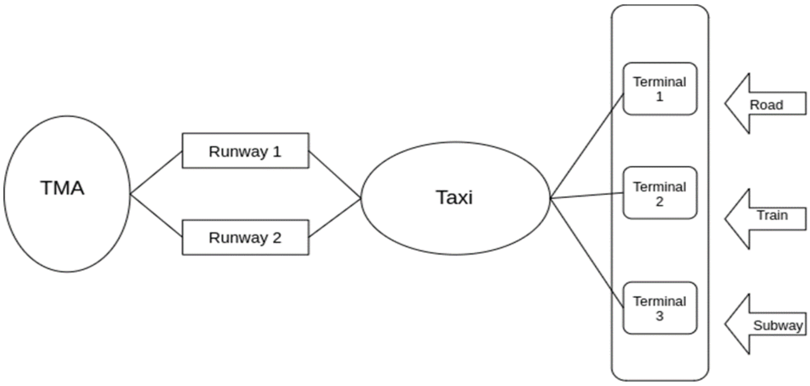

Figure 16: Modelling of airport and access modes

On the ground side (right side of the figure), several access modes (road, train, subway) enable passengers to access the airport. This one is composed of several terminals. Each of them has a maximum aircraft capacity depending on the number of operating gates (or stands). The taxi network, which is referred to 'Taxi' in Figure 16 connects terminals to runways and has also a limited capacity. This means that the number of aircraft in movement within the taxi network is constrained. Finally, a set of runways enables aircraft to enter into /exit from the Terminal Manoeuvring Area (TMA) and each runway has a maximum throughput.

3.3.3 Intended use cases

This coordination tool would be implemented at the AOC. This tool is dynamic and would be launched every time there is an update on outbound passengers' location. For instance, let us consider a day when the subway undergoes a disruption and assume that every 30 minutes GTS communicate to the AOC the location of passengers who are affected by this disruption. AOC could then forecast passenger arrival times at the gate and use this new information as an input for the ADGT. This tool would be run again and a new set of decisions for the flights would be designed. These decisions would be either on the departing flight time (assign a pushback at the gate), on the arriving flight time (assign a speed regulation or a delay at TMA entrance), or on the selection of the departure or the arrival runway. Since decisions are taken at tactical level, a low computational time to provide the solution is required (less than one minute). Furthermore, the forecast of passenger arrival times at the airport is only reliable for flights operated in the following hours of the GTS's communication. Thus, taking decisions during the morning on evening flights is not relevant. In the same manner, delaying a flight which is supposed to land in five minutes is not a good option. Decisions must be taken on a set of flights which are included in a specific time window depending on when the tool is run. According to Paris airports' website, outbound passengers should arrive at the airport between two and three hours before their flight [40]. Thus, a two-hour time window which would start one-hour after the run of the algorithm is fixed. Therefore, flights of passengers related to the last GTS's communication are likely to be operated within the time window targeted.

Example: A shutdown occurs on RER B (subway that links Paris city centre to CDG airport) at 7:45am. Passengers who are likely to miss their flight inform GTS about their current location. This information is provided by GTS to the AOC at 8am. AOC forecasts passenger arrival times at the gate and the ADGT updates decisions on departing and arriving flights between 9am and 11am. An update on passengers' locations is communicated by GTS at 8:30am. The ADGT is run again with this information and decisions on flights operated between 9:30am and 11:30am are updated.

3.3.4 Design

3.3.4.1 Mathematical modelling

A mathematical formulation is proposed to model the operational problem described before.

The following assumptions are made:

- The time scope considered is of one operational day. This time scope is discretised and noted as $\mathcal{T} = \{t_1, .., t_{|\mathcal{T}|}\}$.
- Flights can be assigned to any gate within their associated terminal.
- The flight time duration between one entering point of the TMA and one runway is computed by considering a constant deceleration or acceleration within the TMA.
- For each couple (runway, terminal), the taxi-in and taxi-out durations are assumed constant since no consideration is made on the gate associated at the terminal.
- In this model, aircraft and flights are mixed up. The term 'flight' is used for aircraft through misuse of language to lighten the mathematical modelling.

Data, decision variables, constraints and the objective function are defined below.

3.3.4.1.1 Data

The dataset used within this model is composed of:

- a set of flights (or aircraft) $\mathcal{F} = \mathcal{D} \cup \mathcal{AD} \cup \mathcal{A}$ where $\mathcal{D}, \mathcal{AD}, \mathcal{A}$ refer to a set of departing flights, arriving/departing flights and arriving flights respectively. The first ones are located within the airport at the beginning of the day while the latter will sleep at the airport at the end of the day.
- for each arriving or arriving/departing flight *f* the following characteristics are known:
  - initial Requested Time of Arrival (RTA) T_f^0 at the entering waypoint of TMA,
  - initial speed V_f^0 at the entering waypoint of TMA,
  - a maximum and minimum speed at the TMA entrance V_f^{max} and V_f^{min} respectively,

  - initial landing runway,
  - taxi-in duration,
  - terminal associated.
- for each departing or arriving/departing flight the following characteristics are known:
  - terminal associated,
  - initial off-block time P_f^0,
  - taxi-out duration,
  - initial departure runway.
- a set of terminals $\mathcal{K}$ composing the airport.
- a set of runways $\mathcal{R} = \mathcal{R}^D \cup \mathcal{R}^A$. Each runway is characterised by a maximum throughput per hour f_r^{max} and is dedicated either to departing flights or arriving flights.
- the list of connected flights which have at least one connecting passenger.
- for each couple of connected flights, the number of connecting passengers.
  - for each outbound passenger, the expected arrival time at the gate is known. This expected arrival time is updated depending if a disruption occurred on the ground transport mode used to access the airport.
  - a maximum pushback delay ΔT_{max}^p.
  - a minimum and maximum RTA delay $\Delta T_{min}^{\mathrm{RTA}}$ and $\Delta T_{max}^{\mathrm{RTA}}$ respectively.
  - for each terminal *k* a maximum capacity O_k is set.
  - a maximum capacity O_{Taxi} is set for the taxi network.

3.3.4.1.2 Decision variables

A vector of decisions of length equals to the number of operated flights at the airport is generated. Each decision is associated to a flight *f* and is composed of several characteristics.

- Regarding arriving or arriving/departing flights:
  - the entering time in the TMA T_f^{RTA} . A negative or positive delay can be applied to the initial RTA.
  - the entering speed in the TMA V_f .
  - the landing runway r_f^l.
- Regarding departing or arriving/departing flights:
  - the off-block time P_f.
  - the departure runway r_f^d.

3.3.4.1.3 Constraints

- For each terminal $k \in \mathcal{K}$ and for each time $t_i \in \mathcal{T}$, the number of flights $N_{t_i}^k$ occupying k at t_i is limited:

$$\forall\, k \in \mathcal{K},\ \forall\, t_i \in \mathcal{T},\ N_{t_i}^T \leq O_k\ (1),$$

- For each time $t_i \in \mathcal{T}$, the number of flights $N_{t_i}^{\text{Taxi}}$ occupying the taxi network at t_i is limited:

$$\forall\, t_i \in \mathcal{T},\ N_{t_i}^{\text{Taxi}} \leq O_{\text{Taxi}}\ (2),$$

- For each runway *r*, the number of flight movements per hour N_h^r is constrained:

$$\forall\, r \in \mathcal{R}, \qquad \forall\, h \in \mathcal{T}_h\ , \qquad N_h^r \leq\ f_r^{max}\ (3),$$

  where T_h the set of $\{t_i, .. t_{i+k}\} \subset \mathcal{T}$ of 1-hour length.

- Each departing aircraft needs to be assigned to one departure runway. Let denote $x_{rf} = 1$ if the departing flight *f* is assigned to the runway *r*, 0 otherwise. This can be written:

$$\forall\, f \in \mathcal{D} \cup \mathcal{AD}\,, \sum_{r \in \mathcal{R}^D} x_{rf} =\ 1\ (4)$$

- Similarly, each arriving aircraft needs to be assigned to one arrival runway. Let denote $x_{fr} = 1$ if the arriving flight *f* is assigned to the runway *r*, 0 otherwise. The following constraint can be written:

$$\forall\, f \in \mathcal{AD} \cup \mathcal{A}, \sum_{r \in \mathcal{R}^A} x_{fr}\ = 1\ (5)$$

- The delay assigned to each arriving flight is limited by a minimum and maximum value:

$$\forall\, f \in \mathcal{A} \cup \mathcal{AD}, T_f^0\ - \Delta T_{min}^{\text{RTA}}\ \leq T_f^{\text{RTA}} \leq\ T_f^0\ + \Delta T_{max}^{\text{RTA}}\quad (6)$$

- The delay assigned to each departing flight is limited by a maximum value:

$$\forall\, f \in \mathcal{D} \cup \mathcal{AD},\ P_f^0 \leq P_f \leq\ P_f^0\ + \Delta T_{max}^p\ (7)$$

- The speed assigned to each arriving flight at TMA entrance is limited by a minimum and maximum value:

$$\forall\, f \in \mathcal{A} \cup \mathcal{AD},\ V_f^{min} \leq\ \ V_f\ \leq V_f^{max}\ (8)$$

3.3.4.1.4 Objective function

The objective function is designed to find a balance between three objectives:

- The first objective aims at minimising the number of stranded passengers due to ground transportation disruption:

$$G = \sum_{f \in \mathcal{D} \cup \mathcal{AD}} N_{\text{ground}}^f\ (9)$$

  where N_{ground}^f represents the number of stranded passengers of departing flight f due to ground disruption

- The second objective is to minimise the number of air-connecting passengers missing their connections:

$$A\ = \sum_{f \in \mathcal{D} \cup \mathcal{AD}} N_{\text{air}}^f\ (10)$$

  where N_{air}^f represents the number of stranded passengers of departing flight f due to missed air-connections

- The last objective is set to keep the total delay assigned to flights relatively low:

$$D = \sum_{f \in \mathcal{AD} \cup \mathcal{A}} |T_f^0 - T_f^{\text{RTA}}| + \sum_{f \in \mathcal{D} \cup \mathcal{AD}} |P_f^0 - P_f| \quad (11)$$

- The overall objective is a balanced function of these three objectives:

$$\min \quad \alpha \times G + \beta \times A + \gamma \times D \; (12)$$

The next section presents the algorithm retained to solve this optimisation problem.

3.3.4.2 Simulated annealing

The problem described in the previous section is NP-Hard and its associated search space is large. An order of magnitude of the search space size can be computed based on the following hypothesis:

- 100 flights are operated on average at CDG airport during a two-hour time window;
- the interval time $\mathcal{T}$ is discretised with a time step of one minute;
- departing aircraft can be delayed up to 15 minutes;
- arriving aircraft can be either advanced up to five minutes or delayed up to 15 minutes;
- ten possible speeds at TMA entrance can be assigned to each arriving aircraft;
- the airport has two departure runways which can be used by every departing aircraft;
- the airport is composed of two arrival runways which can be used by every arriving aircraft.

For each aircraft, up to five decisions can be taken (one for each of the item listed above). Considering a set of 100 arriving/departing flights, the number of possible assignments is around $(15 \times 20 \times 10 \times 2 \times 2)^{100} = 8{,}28 \times 10^{407}$. Thus, an exact algorithm is likely to fail in solving this problem in a reasonable time. Moreover, the tool is designed to be used at tactical level with a fast execution time (around the minute). That is why an approximate method has been implemented. Such method is likely to provide good solutions in a short time. A meta-heuristic called Simulated Annealing [41] has been implemented to solve the optimisation problem described in the previous section. This optimisation method is quite powerful in real-life applications with large search spaces [42] [43]. The following description of the Simulated Annealing is inspired from [41].

Simulated annealing (SA) is a metaheuristic useful to solve black box global optimisation problems, whose objective function is not explicitly given, yet can be calculated via simulation. SA is characterised by two parameters: the temperature parameter referred as c_k, and L_k, the number of transitions generated at iteration *k*. The pseudo-code of the SA algorithm is described below:

1) **Initialisation** $i := i_{start}, k := 0, c_k = c_0, \; L_k := L_0$);
2) **Repeat**
3) **For** $l = 0$ **to** L_k **do**
   - **Generate a solution** j **from the neighborhood** S_i **of the current solution** i;
   - If $f(j) < f(i)$ then $i := j$ (j **becomes the current solution**);
   - **Else**, *j* **becomes the current solution with probability** $e^{\left(\frac{f(i)-f(j)}{c_k}\right)}$;
4) $k := k + 1$;
5) **Compute** (L_k, c_k);
6) **Until** $c_k \simeq 0$

One of the primary features of SA is its ability to accept transitions that degrade the objective function. This feature is useful to escape from local optimum. The acceptance of transitions depends on the value of the new solution and c_k. The value of the temperature c_k is initialised with a large value to favour the acceptance of transitions and the exploration of the state space. The more c_k decreases, the less transitions which degrade the objective are accepted. Finally, when c_k is relatively low at the end of the process, no deterioration of the objective is accepted, and the SA algorithm behaves like a Monte Carlo algorithm.

An upstream calibration process can be implemented to choose the initial temperature c_0. This parameter is fixed according to a targeted acceptance rate. The process starts with a low temperature and several transitions are tested. The average acceptance rate for this temperature is then computed. If this rate is higher than the targeted rate, the process is stopped. Else, the temperature is slightly increased, and the process is repeated until reaching the targeted acceptance rate. The usual targeted acceptance rate is $\chi = 0.8$. The number of transitions L_k at iteration k is generally constant and is fixed depending on the size of the problem or a computational time constraint.

In many optimisation applications, the objective function is evaluated thanks to a computer simulation process which requires a simulation environment. In such a case, the optimisation algorithm controls the vector of decision variables, ***X***, which is used by the simulation process to compute the performance (quality), ***y***, of such decisions, as shown in Figure 17.

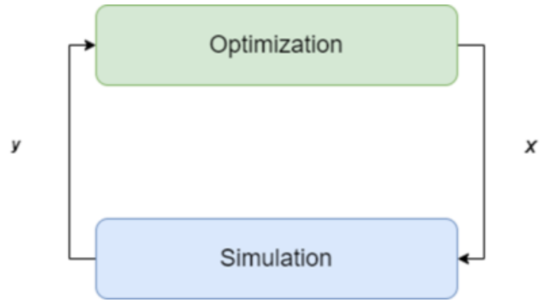

Figure 17: Objective function evaluation based on a simulation process

In this situation, population-based algorithms may not be suitable to address such problems, mainly when the simulation environment requires a huge amount of memory space as it is often the case in today real-life complex systems. As a matter of fact, in the case of a population-based approach, the simulation environment must be duplicated for each one of the population of solutions, which may require an excessive amount of memory. To tackle this issue, one may think about having only one simulation environment which could be used each time a point in the population has to be evaluated as follows: one first considers the first individual for which the simulation environment is initiated, and the simulation associated with this first individual is run. The associated performance is then transferred to the optimisation algorithm. After that, the second individual is evaluated, but the simulation environment must first be cleared from the events of the first simulation. The simulation is then run for the second individual, and so on until the last individual of the population is evaluated. In this case the memory space is not an issue anymore, but the evaluation time may be excessive and the overall process too slow since the simulation environment is reset at each evaluation.

On the contrary, the SA is a single-solution-based algorithm. The simulation is only run one time at each iteration, allowing faster computational time of the algorithm. Thus, SA is better fitted than population-based algorithm with such evaluation method.

3.3.4.3 Implementation

The relaxation of constraints by penalising the objective function is often used for black box global optimisation problems. Indeed, since the evaluation of several constraints' satisfaction is done through the simulation, it is hard to generate a new feasible solution at each occurrence. Thus, terminal capacities, taxi network capacity and runway throughput constraints are relaxed by adding a penalty term for their violations to the objective function.

Each decision is initialised with the initial flight schedule: initial entering time and speed at TMA entrance, initial arrival and departure runway and initial off block time. Decisions (i.e., flights) are inserted into the environment and the objective function is evaluated through the simulation environment. A performance is associated to each decision. As a reminder, the goal is to minimise the objective function and an increase in performance decision should be interpreted as an increase in the objective function (which translates into a decrease in the decision quality). For instance, if the capacity constraint of terminal k is violated at time t_i, performances of decisions which occupy k at t_i are increased. In the same manner, the more stranded outbound passengers of decision *i*, the higher the performance of *i*. The performance computation of each decision is used for the neighbour generation process. Indeed, the performance guides the algorithm in the selection of poor decisions which need to be changed. Only one decision of the decision vector ***X*** is modified to generate a neighbour of the current solution as illustrated in Figure 18.

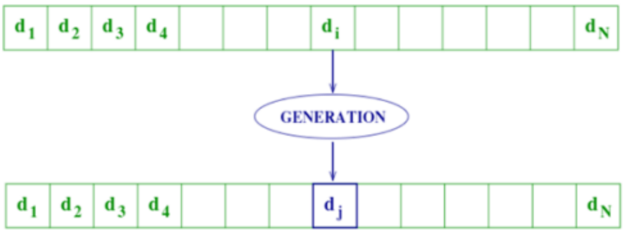

Figure 18: Illustration of the neighbour generation process

The selection of the decision is stochastic. The higher the performance of a decision, the higher the probability of generating a new neighbour by changing this decision. The performance of each decision can be split into five terms:

- ground movement performance
- arrival runway performance
- departure runway performance
- arrival passenger performance
- departure passenger performance

The first term is associated to the aircraft performance related to the violation of terminal and taxi network constraints. The second and third ones are related to the violation of arrival and departure runway constraints respectively. The fourth and the fifth ones are linked to the number of stranded passengers for arriving flights (stranded air connecting passengers) and departing flights (stranded outbound passengers and air connecting passengers). In the same way as the selection of which decision to modify, the selection of the decision's characteristic to modify depends on the five performance terms. For instance, if the arrival runway performance of decision *k* is higher than the other performances, the arrival runway of *k* is likely to be modified.

A sliding time window is also implemented to only modify decisions which are operated between one hour and three hours after the run of the algorithm. The implementation of the time window is well explained by [44].

Each flight, depending on its relative position with the time window considered, has one status among the four following ones:

- **Completed flight**: the flight has already finished its operation before the start of the time window considered
- **On-going flight**: the flight is operated within the time window, but decision variables related to this flight have already been fixed during the previous time windows. This status is for flights which have not been fully operated but are too close in the future to be modified. These flights are considered as a constraint for the active ones.
- **Active flight**: the flight is operated within the time window and its decision variables can be modified. This status represents the actual decisions associated to the considered time window.
- **Planned flight:** flights which will be operated after the considered time window.

The status of each flight is updated before each time the algorithm is run. The mechanism is implemented in Java.

A simplified class diagram of the code is presented in Figure 19.

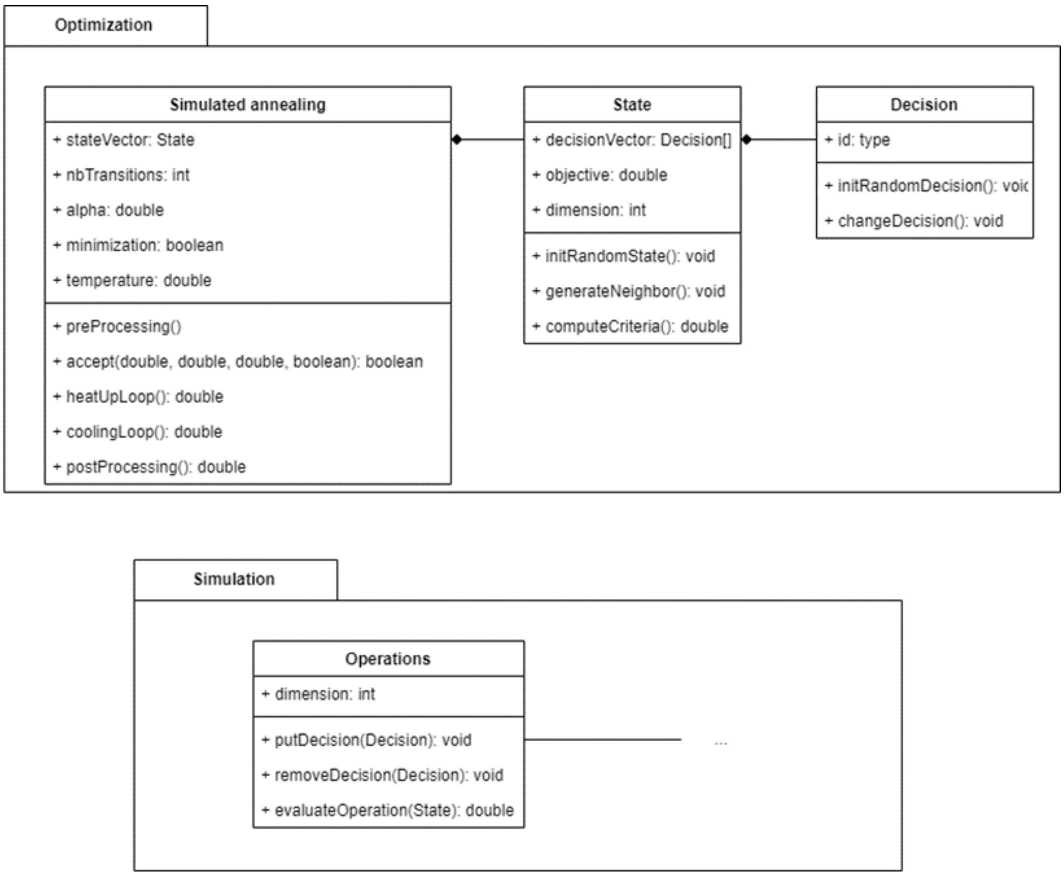

Figure 19: Simplified class diagram of the optimisation tool

The optimisation module is composed of three classes:

- *Simulated annealing*: the metaheuristic employed to solve the optimisation problem. It is at the core of the optimisation process. The hyperparameters defined above (temperature, number of transitions) are attributes of the class. It is composed of two main methods: the *heatUpLoop()* procedure and the *coolingLoop()* procedure. The heat-up phase will fix the temperature parameter which will serve as an input of the *coolingLoop()*.
- *State*: this class represents a solution. An instance of *State* gathers the set of decisions that are taken at each iteration. The state is evaluated in the simulation module.
- *Decision*: as explained above, a decision is made for each flight. Each decision has several characteristics such as the pushback assigned for a departing flight, a speed regulation or delay at TMA entrance for arriving flight or even a change in runway assignment. Every decision is initialised at the beginning of the optimisation process (*initRandomState()*) and later, is changed to improve the objective function (*changeDecision()*).

The simulation module is problem dependent. The *Operations* class acts as an interface between the optimisation module, which is generic, and the problem. It is composed of three main methods:

- *putDecision()*: this method adds the new decision to the simulation
- *removeDecision()*: this procedure removes the old decision from the simulation
- *evaluateOperations()*: this procedure runs the simulation and return the objective value.

The optimisation process is described by the following scheme: at each iteration, the current solution is stored and a new solution is generated through the method *generateNeighbor()*. This method changes a non-efficient decision of the copy of the current solution through the use of *changeDecision()*. This decision is removed from the simulation environment (*removeDecision()*) and replaced by the new one (*putDecision()*). The new state is evaluated through the *evaluateOperations()* procedure. The objective function related to this new state is compared with the previous solution. If the transition is accepted, the current solution is replaced by the new one. Else the current solution is kept and replaces the new solution into the simulation environment. This procedure is repeated *nbTransitions* times at each temperature step. The optimisation process ends when the temperature reaches a lower threshold.

The *Operations* class acts as a generic interface between the *Optimisation* and *Simulation* package. Additional classes are defined in the Simulation package to model the features of the problem considered. For instance, in the ADGT implementation, classes *Flight, Airport* and *Passengers* are defined. The full class diagram of the ADGT's JAVA code is presented in Figure 20.

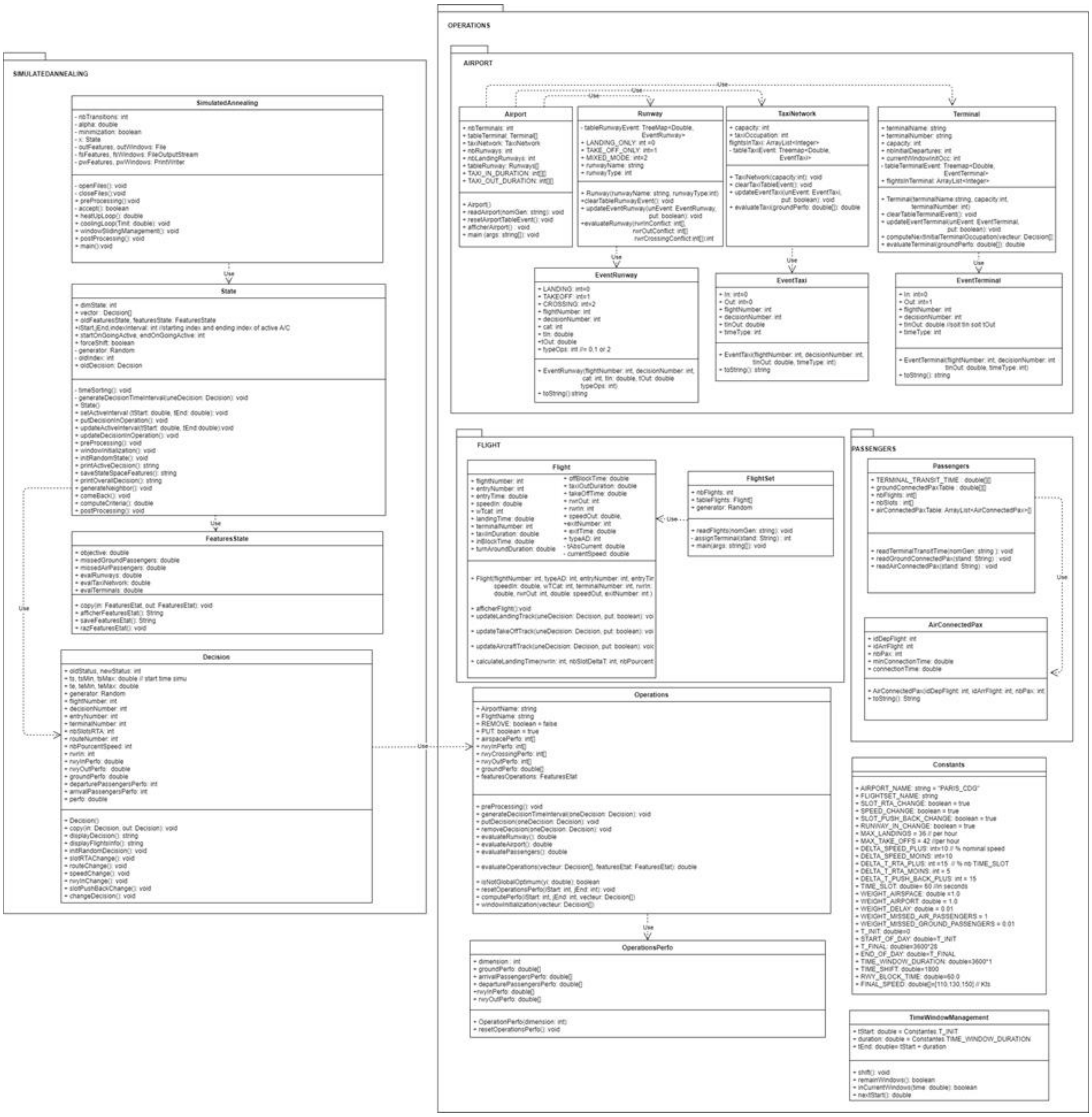

Figure 20: Class diagram of the ADGT

The code is composed of two main packages called '*SIMULATEDANNEALING*' and '*OPERATIONS*'. The first one stands for the generic Optimisation package previously introduced. *OPERATIONS* is problem specific and is composed of three packages:

- *AIRPORT:* includes all the classes related to the movement of aircraft. Terminal, taxi network and runway classes are included in this package. Each of these classes contains a Treemap where events are stored. For instance, an event on a terminal corresponds to an arriving or a departing flight which enters/exits the terminal. Classes *EventTerminal, EventTaxi* and EventRunway are linked to the classes cited below respectively. For instance, during the simulation, if a flight lands on runway *r* at t_i, an *EventRunway* is created and stored in the Treemap of the class linked to *r* with the id t_i.

- *FLIGHT:* includes the information of each flight and is composed of two classes: *Flight* and *Flightset*. The class *Flightset* is used to read the data while the class *Flight* is used to update the aircraft position during the simulation.
- *PASSENGERS:* includes the information related to outbound passengers and air-connecting passengers. A class *Passengers* is used to read the passengers' arrival time at the gate and information on connecting flights. A table *groundConnectedPaxTable* is precomputed to store the number of stranded passengers for each departing flight depending on the pushback assigned to this one. The class *AirConnectedPax* is implemented to gather information for each connecting flights (id arriving flight, id departing flight, number of passengers, minimum connection time, actual connection time). An ArrayList of *AirConnectedPax* is defined in the class *Passengers*.

Four other classes are defined in this package: *Operations, Constants, TimeWindowManagement* and *OperationsPerfo*. *Operations* acts as an interface between the *SIMULATEDANNEALING* and *OPERATIONS* packages as explained above. *TimeWindowManagement* is the class which handles the status of each decision (*Completed, On-going, Active, Planned*). *Constants* class contains all the constants of the problem and *OperationsPerfo* is used to store the performance of each decision during the simulation. This information is transmitted after the simulation to the class *Decision*.

3.3.5 Concluding comments

This subsection has presented the coordination mechanism tool (ADGT) which could be implemented in case of a disruption on the ground side. The ADGT uses a simulated annealing algorithm coupled with an evaluation of the objective function through simulation to provide a new flight schedule and flight runway assignment. This tool is supposed to be used at tactical level each time passengers' locations are updated. This information would be provided to the AOC by GTS. However, such level of information sharing between ground and air transportation stakeholders is not currently implemented. Hence, an agent-based simulation such as MATSim can provide reliable inputs to the tool and enable to assess the relevance of ADGT. Figure 21 summarises the integration of ADGT with MATSim. First, the initial flight schedule would be provided to MATSim and a disruption on the ground side would be simulated. Then, passenger arrival times at the gate would be forecasted thanks to MATSim. This information and the initial flight schedule would serve as inputs for the coordination tool. The ADGT would return a new flight schedule in response to this disruptive event. Finally, multimodal performance indicators defined in TRANSIT's deliverable D3.1 would be used to compare the quality of the transportation system with and without coordination mechanism.

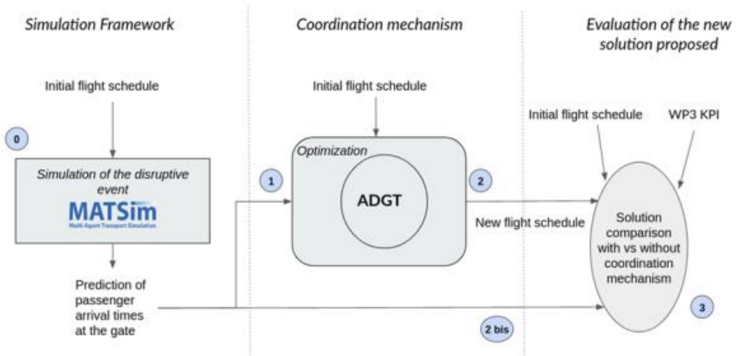

Figure 21: Integration of the ADGT with the simulation framework

3.4 Timetable synchronisation between air and rail

This subsection provides a detailed description of the timetable synchronisation mechanism proposed by TRANSTI. This mechanism aims at better integrating the air transportation with the ground transportation system by making the flight and train schedules dependent on each other. The purpose of the tool is to provide a new timetable, which is more attractive and robust for passengers who use the train to reach the airport.

3.4.1 Context

For every transportation supplier, the scheduling process follows a hierarchical process such as the one presented in Figure 22.

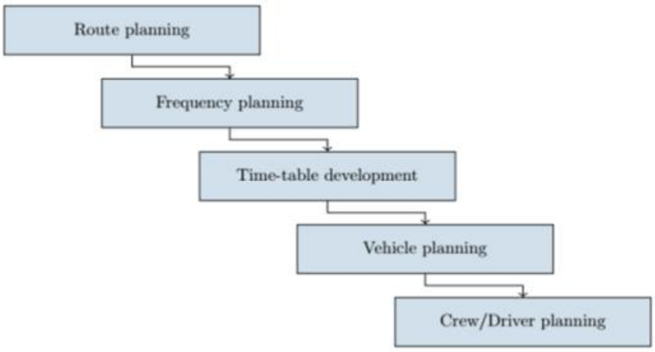

Figure 22: The hierarchical planning process

First, the route on which transportation suppliers will operate and with which frequency are defined. These routes and frequencies are chosen depending on the expected profitability and competition strategies of airlines. According to [45], airlines establish route profitability models to support their planning decision. These models contain accurate information on airline expenditures for each route. Based on these estimations, airlines then decide which route to integrate in their network. Subsequently, the frequency planning for airline is decided. Frequency is intrinsically related to competitiveness on a market. Indeed, as exposed by [46], a higher frequency will increase the market share of an airline. [46] proposes several models for airlines to construct their frequency planning to minimise their operational cost. [47] presents a heuristic to determine flight frequency and aircraft routing for small size airlines. [48] proposes a method to determine flight frequencies on route subject to competition.

After routes and frequencies are decided, the final departure time and arrival time of each leg is set. Historically, due to computational limitations, these timetables were designed manually by transport operators or with the use of heuristics [49]. However, the development of computation capacity has permitted the use of optimisation methods to solve the schedule design problem. Regarding the air transportation system, the scheduling of flights is usually coupled with the fleeting problem. This problem corresponds to assigning for each flight a fleet type. Each fleet type has its own characteristics, such as a maximum capacity and a minimum turnaround time. Hence, the schedule directly affects the fleeting decision. For instance, a specific departure time can be more attractive for passengers and assigning a long-range aircraft is likely to be more profitable for airlines than a smaller one which could induce spill cost. Also, changing the schedule will modify the available turnaround time between two consecutive flights and question the initial fleeting decision.

To guarantee that the schedule is optimal, all the different tasks of the hierarchical process should be solved in an integrated manner. However, as explained by [45], airlines (and this is also a concern for other modes) face some difficulties:

- Accurate and recent data (such as passengers' expectation) required to establish the optimisation model are difficult to obtain
- Building an integrated schedule from scratch is computationally intractable
- An integrated model could result in large changes in the initial schedule. These large changes are not wished by airlines since regularity is valued by passengers

Hence, many of the optimisation models tend to create a new schedule close to the original one. [50] was one of the first to solve the fleet assignment problem allowing changes in the schedule if this could reduce the fleeting size. Also, [51] noticed that a little change in the initial schedule could reduce the cost of the fleet assignment. This resulted in a reduction of two in the number of aircraft needed and a cost saving of $50 million.

Regarding train operators, they must respect one additional constraint: the single-track allocation. Indeed, unlike aircraft, trains could not overtake by changing their trajectory since they evolve on tracks. Hence, the scheduling problem for railway operators consists in finding a feasible timetable that respects track constraints. The objective is to minimise the schedule deviation from the original schedule. [52] proposed a formulation of the train timetabling problem where the objective is to minimise the schedule deviation. [53] tried to solve the train timetabling problem considering passengers' travel time and fuel consumption.

Finally, the planning process ends by assigning crews and vehicles to each leg of the timetable. Each transportation stakeholder established its proper schedule, without considering other modes. However, some synchronisation between modes exists, especially for freight and urban mobility. [54] presented an overview of the multimodal planning process in the freight industry. Also, urban mobility, and especially coordination between public and private transit received a lot of attention over the past few years. For instance, [55] proposed a model that fixes buses frequencies, sizes and fares in order to find a balance between crowded buses and congestion at the bus station. [56] presents an analysis to define on which context the coordination between timetables of public transports is relevant. [57] developed a tool to synchronise the bus timetable with the last train scheduled.

More recently, coordination between air and the ground transportation modes receives a growing attention. Intermodal solutions are encouraged to alleviate airport congestion and mitigate the environmental footprint of passenger journey. For instance, the French government has enforced the removal of feeder flights when an alternative by train under 2h30 exists. Also, several studies which focus on multimodal coordination have been led. [58] developed a mechanism to increase the number of connections between trains and flights at an airport. [59] implemented a model to mitigate air passengers' disruption using buses during the Asiana crash crisis.

The timetable synchronisation tool aims at providing a robust planning from a passenger perspective to limit missed connection in case of delays. A more attractive multimodal network might encourage passengers to shift from feeder flights to trains. This will also benefit from the air transportation system by reducing airspace congestion, and free some space to additional long-haul flights. The mathematical formulation of the optimisation problem is presented in the following section.

3.4.2 Description of the tool

A coordination mechanism at the strategic level between flights and trains is presented. This mechanism refers to the establishment of a synchronised timetable between these two modes. The timetable corresponds to the scheduled departure time and arrival time of trains/flights set by transportation suppliers and provided to passengers. The objective is to generate a global timetable that will improve the passenger experience and encourage people to use an integrated service. It is especially suitable in case of airports equipped with a train station. Thus, this mechanism could be seen as a "soft measure" which will not require the building of new infrastructures. The tool developed aims at providing a timetable with a high level of service. This would be done by optimising the connection times between trains and flights regarding passengers' expectations.

The main idea of such mechanism lies in building a coordinated timetable from two independent schedules. Only small changes are authorised to keep the structure of the initial planning. Figure 23 illustrates this coordination mechanism.

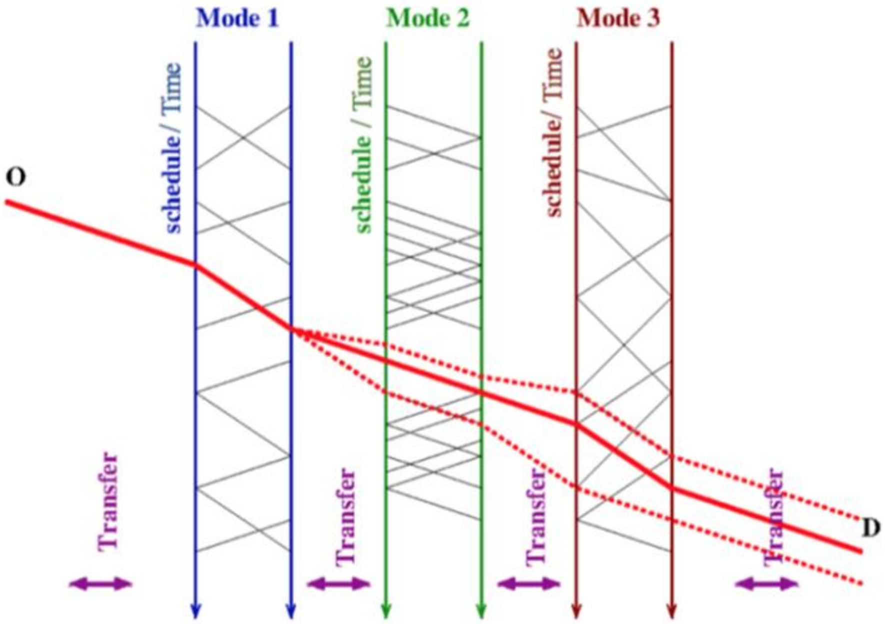

Figure 23: Illustration of the timetable synchronisation mechanism

Assume a passenger journey relying on three transport modes. Timetables of Mode 1 and 2 would be synchronised by optimising the transfer times between them. However, changing the timetable of Mode 2 is likely to modify the transfer times with Mode 3. Hence, an optimisation method can be implemented to handle the overall transfer times simultaneously.

Example: A passenger plans a trip between Marseille and New-York city, transiting through CDG airport. His trip relies on the use of several transportation means:

- *a bus from home to the train station Marseille-Saint-Charles;*
- *a train between Marseille-Saint Charles train station and CDG airport;*
- *a flight from CDG airport to John F. Kennedy (JFK) airport;*
- *a cab to leave JFK airport and reach the final destination.*

Independently, an airline operating transatlantic flights and the French railway operator SCNF provide initial flight and train schedules, respectively. Also, the public transportation company which operates in Marseille shares the daily schedule with buses frequencies that serves the train station. Then, the timetable synchronisation tool will assign slight modification to these rail, flight and bus schedules to smooth passengers' door-to-door journey. For each departure time or arrival time of a train/flight at CDG airport, a change within a 15-minutes time window is authorised. The bus frequency at Marseille train station can also be increased/decreased up to a factor of two. The three modes will be synchronised to target optimal connection times for bus/train connection and train/flight connection of 30 and 120 minutes, respectively. This process could be applied to the entire transportation network, considering the overall potential door-to-door demand.

Quantifying the quality of a connection

The mechanism developed aims at providing a comfortable timetable for passengers who transit between trains and flights at a hub airport. In passengers' perception, a good connection must be robust, and the waiting time should be limited. Indeed, according to [60], the waiting time can be perceived up to three times the in-vehicle time. The first step is then to define a score which quantifies the quality of a connection between two modes based on its duration. A Gamma distribution has been used to model the quality of a connection time as illustrated in Figure 24.

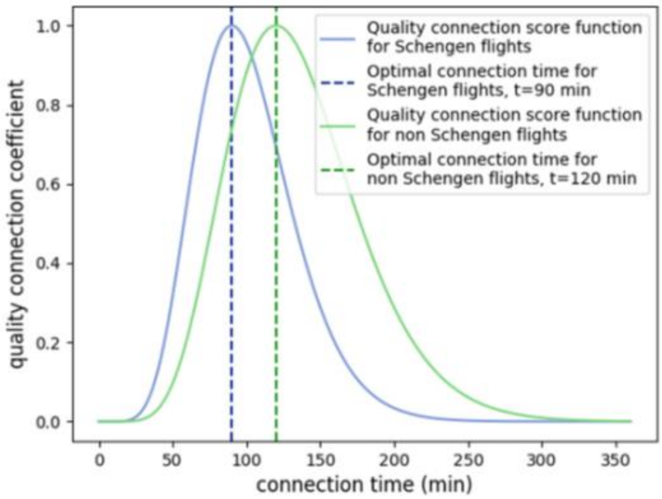

Figure 24: Connection time quality function

The asymmetrical nature of such function captures the trade-off between robustness and limited waiting time. A short connection is less valuable than a long connection because it is not robust in case of delay. Hence, one can assume that passengers prefer a 'too long' than a 'too short' connection time. A score between 0 and 1 is assigned for each connection duration. As the connection time gets closer to an optimal duration, the score increases toward 1. On the contrary, as the connection time draws away from the optimal duration, the score tends to 0. For the reason explained above, the score has a stronger decrease on the left side of optimal duration than on the right side. This will limit the generation of short connection times in the final timetable. The scoring function is defined as:

$$\sigma(t) = \frac{f(t, \alpha, \beta)}{f(t_{opt}, \alpha, \beta)},$$

where f is the gamma distribution with α the shape parameter and β the scale parameter, and t_{opt} the optimal connection time. Parameters α and β must be calibrated, depending on the optimal connection time set. Figure 24 displays for instance the scoring function calibrated for connections between arriving trains and departing flights at CDG airport. This airport is the largest French airport. Also, France belongs to the Schengen space. The processing time at the airport for the flights within the Schengen space is then reduced since no border control is involved. Hence, regarding the optimal connection time, a distinction is made between Schengen flights and non-Schengen flights. Optimal connection times of 90 minutes and 120 minutes were arbitrarily fixed for Schengen flights and non-Schengen flights respectively. These values can be tuned by the user depending on passengers' features (leisure/business travellers, age, conservative/stress-free, etc.) or airport features (hub/regional, passenger volume, etc.). Furthermore, a minimum connection time is set under which the connection time is not interesting for passengers. In the same manner, a maximum threshold connection time is fixed. Above that maximal threshold and under the minimal threshold, the connection score must be close to 0. A score lower to 0.01 has been retained.

Table 4 summarises the values of minimum, optimal and maximum connection times fixed for Schengen and non-Schengen connections.

Table 4: Minimum, optimal and maximum connection times retained for the study

	t_{min}	t_{opt}	t_{max}
Train/Schengen flight connection	20	90	240
Train/non-Schengen flight connection	30	120	300

The objective is then to calibrate the α and β of the scoring function, based on these assumptions.

The mode of the Gamma distribution is defined by:

$$Mode = \theta(\alpha - 1) \text{ , where } \theta = \frac{1}{\beta}.$$

The mode represents the most represented values, which can be interpreted as the value which maximise the Gamma function. Hence, the mode is equal to t_{opt}. Thus, the α and θ parameters are linked by the following relationship:

$$t_{opt} = \theta(\alpha - 1).$$

Several values of (α, θ) have been tested. Values which satisfy the 0.01 threshold for Schengen connections and non-Schengen connections are (9,15) and (9,11.25) respectively. These values are taken for the rest of the study. However, depending on the modes considered, these optimal connection times can be tuned. The quantification of the connection time was a mandatory step to design the coordination mechanism. Indeed, operators can build an optimal timetable regarding these targeted values.

The timetable synchronisation tool aims at providing a timetable that favours a qualitative connection from the passenger perspective. The objective is to improve each connection time by getting closer to an optimal one.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

3.4.3 Design

3.4.3.1 Mathematical modelling

Given an initial train schedule and an initial flight schedule for a specific day, the objective is to synchronise them to coordinate both modes.

3.4.3.1.1 Input data

- $\mathcal{F} = \mathcal{F}^A \cup \mathcal{F}^D$: the set of arriving and departing flights at the hub
- $\mathcal{T} = \mathcal{T}^A \cup \mathcal{T}^D$: the set of arriving and departing trains at the hub
- $\mathcal{C} \subset \mathcal{F}^A \times \mathcal{F}^D$: the set of flight couples which are successively operated by the same aircraft
- $\forall (t, f) \in (\mathcal{T}^A \times \mathcal{F}^D),\ \Phi_{tf}$: the flow of passengers between an arriving train t and a departing flight f
- $\forall (f, t) \in (\mathcal{F}^A \times \mathcal{T}^D),\ \Phi_{ft}$: the flow of passengers between an arriving flight f and a departing train t
- $\forall f \in \mathcal{F}^D, d_f^0$: the initial departure time of flight f
- $\forall f \in \mathcal{F}^A, a_f^0$: the initial arrival time of flight f
- $\forall t \in \mathcal{T}^D, d_t^0$: the initial departure time of train t
- $\forall t \in \mathcal{T}^A, a_t^0$: the initial arrival time of train t
- $\forall (f_1, f_2) \in \mathcal{C}, \lambda_{f_1 f_2}$: the minimum turnaround time for this flight couple
- Δ_{min}: minimum slack time
- Δ_{max}: maximum slack time
- α_{in} : the shape parameter calibrated for train/flight connections
- β_{in} : the scale parameter calibrated for train/flight connections
- α_{out} : the shape parameter calibrated for flight/train connections
- β_{out} : the scale parameter calibrated for flight/train connections
- $\delta_{\mathrm{in}}^{max}$: maximum train/flight connection time allowed
- $\delta_{\mathrm{in}}^{min}$: minimum train/flight connection time allowed
- $\delta_{\mathrm{in}}^{opt}$: optimal train/flight connection time from a passenger perspective
- $\delta_{\mathrm{out}}^{max}$: maximum flight/train connection time allowed
- $\delta_{\mathrm{out}}^{min}$: minimum flight/train connection time allowed
- $\delta_{\mathrm{out}}^{opt}$: optimal flight/train connection time from a passenger perspective
- $\sigma_{\mathrm{in}}(\mathrm{t}) = \frac{f(t, \alpha_{\mathrm{in}}, \beta_{\mathrm{in}})}{f(\delta_{in}^{opt}, \alpha_{\mathrm{in}}, \beta_{\mathrm{in}})}$: The scoring function for train/flight connection, where $f(t, \alpha, \beta)$ is the gamma distribution calibrated for the train/flight connection time
- $\sigma_{\mathrm{out}}(\mathrm{t}) = \frac{f(t, \alpha_{\mathrm{out}}, \beta_{\mathrm{out}})}{f(\delta_{\mathrm{out}}^{opt}, \alpha_{\mathrm{out}}, \beta_{\mathrm{out}})}$: The scoring function for flight/train connection, where $f(t, \alpha, \beta)$ is the gamma distribution calibrated for the flight/train connection time

3.4.3.1.2 Decision variables

- $\forall f \in \mathcal{F}^D, \Delta_f^{\mathrm{d}}$: slack time assigned at the departure time of flight f
- $\forall f \in \mathcal{F}^A, \Delta_f^{\mathrm{a}}$: slack time assigned at the arrival time of flight f
- $\forall t \in \mathcal{T}^D, \Delta_t^{\mathrm{d}}$: slack time assigned at the departure time of train t
- $\forall t \in \mathcal{T}^A, \Delta_t^{\mathrm{a}}$: slack time assigned at the arrival time of train t
- $\forall f \in \mathcal{F}^D,\ d_f = d_f^0 + \Delta_f^{\mathrm{d}}$: new scheduled departure time of flight f
- $\forall f \in \mathcal{F}^A,\ a_f = a_f^0 + \Delta_f^{\mathrm{a}}$: new scheduled arrival time of flight f
- $\forall t \in \mathcal{T}^D,\ d_t = d_t^0 + \Delta_t^{\mathrm{d}}$: new scheduled departure time of train t
- $\forall t \in \mathcal{T}^A,\ a_t = a_t^0 + \Delta_t^{\mathrm{a}}$: new scheduled arrival time of train t
- $\forall\, (t,f) \in (\mathcal{T}^{\mathrm{A}} \times \mathcal{F}^{\mathrm{D}}),\ \mathrm{x}_{tf} = \begin{cases} 1 \text{ if } \delta_{in}^{min} \leq d_f^0 + \Delta_{min} - (a_t^0 + \Delta_{max}) \leq \delta_{in}^{max} \\ 0 \text{ otherwise} \end{cases}$: The feasibility of a train/flight connection
- $\forall\, (f,t) \in (\mathcal{F}^{\mathrm{A}} \times \mathcal{T}^{\mathrm{D}}),\ \mathrm{x}_{ft} = \begin{cases} 1 \text{ if} \delta_{out}^{min} \leq d_t^0 + \Delta_{min} - (a_f^0 + \Delta_{max}) \leq \delta_{out}^{max} \\ 0 \text{ otherwise} \end{cases}$: The feasibility of a flight/train connection

3.4.3.1.3 Constraints

- The slack time assigned to each arriving flight is constrained:
$$\forall\, f \in \mathcal{F}^A, \Delta_{min} \leq \Delta_f^{\mathrm{a}} \leq \Delta_{max}$$
- The slack time assigned to each departing flight is constrained:
$$\forall\, f \in \mathcal{F}^D, \Delta_{min} \leq \Delta_f^{\mathrm{d}} \leq \Delta_{max}$$
- The slack time assigned to each arriving train is constrained:
$$\forall\, t \in \mathcal{T}^A, \Delta_{min} \leq \Delta_t^{\mathrm{a}} \leq \Delta_{max}$$
  - The slack time assigned to each departing train is constrained:
$$\forall\, t \in \mathcal{T}^D, \Delta_{min} \leq \Delta_t^{\mathrm{d}} \leq \Delta_{max}$$
  - The turnaround time for two consecutive flights operated by the same aircraft is constrained:
$$\forall\, (f_1, f_2) \in \mathcal{C}, d_{f_2} - a_{f_1} \geq \lambda_{f_1 f_2}$$

3.4.3.1.4 Objective function

The objective is to maximise the quality of the schedule:

$$Max \sum_{t\,\in \mathcal{T}^A} \sum_{f \in \mathcal{F}^D} \mathrm{x}_{tf} . \sigma_{\mathrm{in}}(d_f - a_t) . \Phi_{tf} + \sum_{f \in \mathcal{F}^A} \sum_{t \in \mathcal{T}^D} \mathrm{x}_{ft} . \sigma_{\mathrm{out}}(d_t - a_f) . \Phi_{ft}$$

Here, the objective is solved for one airport. However, several airports can be considered. A change in a schedule for a flight affects the origin and the destination airports; hence, all airports should be considered when building a global schedule. Since this consideration will significantly increase the complexity of the problem, a metaheuristic has been employed to solve it. The method chosen is simulated annealing. The implementation of the mechanism is detailed in the following section.

3.4.3.2 Implementation

The mechanism is implemented in Java. The process is summarised in Figure 25.

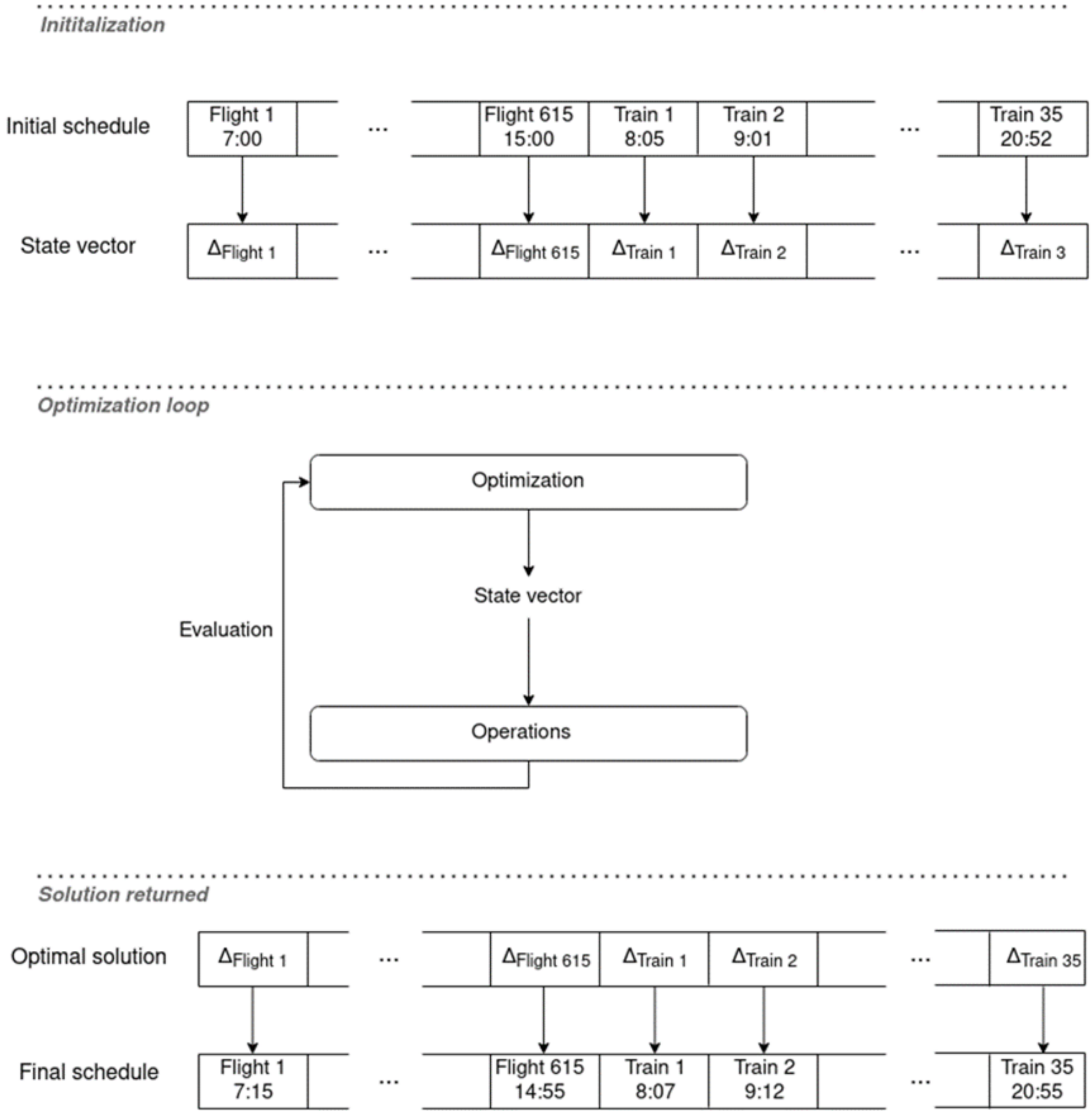

Figure 25: Overview of the optimisation process of the timetable synchronisation mechanism

For this tool, one decision corresponds to the Δ_f or Δ_t assigned to a flight or a train respectively. Each decision must be made between a Δ_{min} and $\Delta_{\max}$ which are fixed at -15 and 15 minutes respectively. Each decision is initialised by randomly choosing a value between -15 and 15. The objective of the optimisation loop is then to find the best decision to take for each leg. Each decision is also associated with a performance, which represents the quality of the decision in the global schedule.

At each step, a neighbour of the current solution is generated. Only one decision is modified to generate a neighbour of the current solution based on the decisions' performance. However, the timetable synchronisation optimisation aims at maximising the objective function. In this case, the lower the performance of a decision, the higher the probability to be chosen. Once a decision is selected, a new value between -15 and 15 is randomly chosen. As previous, the old decision is removed from the simulation and replaced by the new one.

For the synchronisation tool, the problem-specific class corresponds to the *Schedule* class. The most important attribute of the class is the *connectionScore* table. This matrix gathers the score of each feasible connection, as displayed in Table 5: Connection score matrix. The addition of the decision to the operation corresponds to changing the connection score between the leg affected and its connecting legs. Especially, every time a decision is changed, the *putDecision()* procedure is called. The latter calls the *putOneDelta()* method of the *Schedule* class which updates the *connectionScore* matrix within the *updateDecision()* procedure. The objective function is directly accessible in this matrix since it corresponds to the sum of all the connection scores with flights for trains (green column). It can also be computed by summing the connection score for each flight (orange row). Hence, every time the optimisation loop has to evaluate the new state, the *evaluateOperations()* function will call the *evaluateAllLegs()* function, that will read the objective value in the *connectionScore* table.

Also, the performance of each leg is directly accessible in the table. For a train *i*, the performance will correspond to the sum of scores with each flight, which corresponds to the element located at the i^{th} row and the last column of the matrix. In the same manner, the performance of one flight *j* is obtained by reading the value of the j^{th} column and the last row of the *connectionScore* table.

Table 5: Connection score matrix

Connection score	...	Flight j	...	Score by train
...				
Train i		α_{ij}		$\Sigma_j \alpha_{ij}$
...				
Score by flight		$\Sigma_i \alpha_{ij}$		$\Sigma_i\Sigma_j \alpha_{ij}$

Objective

A detailed class diagram specific to this problem is also displayed in Figure 26.

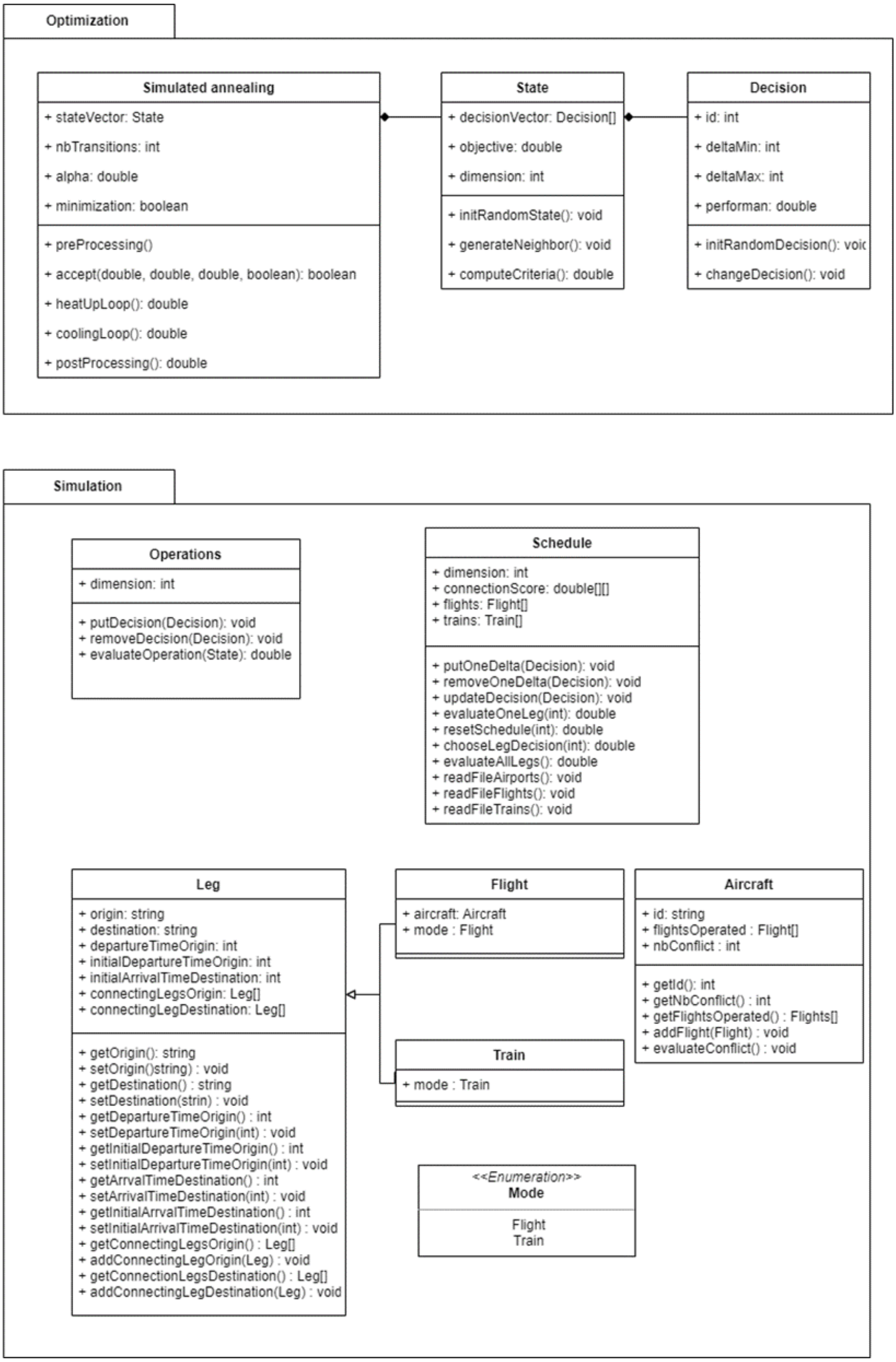

Figure 26: Class diagram of the timetable synchronisation tool

As explained above, the *Schedule* class is specific to the problem and gathers information on the connection time between each leg. Two other classes are implemented:

- *Leg*: this superclass gathers the characteristic of each leg such as the origin, destination, departure and arrival time, but also which potential legs can be connected. For instance, for a train arriving at 9:00 am at CDG airport, flights that depart between 9:00 am and 2:00 pm are considered as connecting legs. This information reduces the computation time by only changing the score between connecting flights with a score potentially higher than 0.
- *Aircraft*: this class is used to compute the number of turnaround conflicts. For each aircraft, the number of conflicts will be evaluated when a decision is made among the flights operated by this aircraft. The aircraft turnaround constraint is relaxed in the implementation. A low performance will be assigned to flights which violate this constraint. Furthermore, a penalty term is added to the objective function, which will reduce it if turnaround conflicts exist.

The classes *Flight* and *Train* extend the *Leg* class. Other classes such as *Bus* or *Subway* could be implemented for a problem extension.

3.4.4 Concluding comments

The model developed aims at synchronising train and flight timetables at a hub airport. From two independent schedules, a new one that favours the optimal connection time for passengers is generated. The optimisation problem is solved using a simulated annealing algorithm. This mechanism is designed to be used at strategic level, when airlines and rail operators are building their schedules, several months before the day of operations. The evaluation of this mechanism will be done using J-TAP. Indeed, the attractiveness of a transportation mean depends on the connection time with other modes. This new schedule, and the connection time scores associated could be used as a new input to calibrate J-TAP. The simulation framework will then evaluate how the new schedule has impacted the demand and the relevance of this mechanism.

4 Using the tools: code repositories and technical documentation

4.1 J-TAP

Since J-TAP is a software library, a developer-oriented work is involved. A version control system to support, coordinate, and orient the development of new features in the J-TAP community is crucial. A version control system enables multiple developers or teams to work in an isolated fashion without affecting the work of others. This isolation enables features to be built and tested in the development phase and eventually deployed to the stable version.

J-TAP relies on GitHub. J-TAP can be cloned from the remote repository and integrated in a Java-Maven project (Apache Maven) specifying the dependency in the Project Object Model.

J-TAP repository can be found here: J-TAP

The branching workflow used in the development of J-TAP is Git Flow.

Updated documentation is maintained in the J-TAP repository in the branch igg_doc.

A complete tutorial on how to use and extend J-TAP can be found here.

4.2 MATSim Within-day Re-planning with Mode Choice

As J-TAP, MATSim is a software library, so a version control system is also crucial to enable for independent and isolated model enhancement. This isolation enables features to be built and tested in the development phase and eventually deployed to the stable version.

MATSim relies on GitHub. MATSim can be cloned from the remote repository and integrated in a Java-Maven project (Apache Maven) specifying the dependency in the Project Object Model.

MATSim itself can be found at https://github.com/matsim-org. However, for a modeller looking to use the functionalities developed during the TRANSIT project, MATSim will simply be a dependency that is specified in the pom.xml file. For new users and interested readers, the traditional usage of MATSim can be found in the MATSim Book [22].

eqasim will also simply be a dependency (for an example, see the pom.xml file in the eutransitWDR repository). For reference, though, one can find an introduction to the eqasim framework here: https://eqasim.org/ . The code itself is available on GitHub here: https://github.com/eqasim-org . For the scenario generation pipeline, see https://github.com/eqasim-org/ile-de-france . For the code that is used in the MATSim loop, see https://github.com/eqasim-org/eqasim-java .

For new users, the repository maintained by the lecturers of the Agent-Based Modelling course at the ETH Zurich may be useful. This also includes some simple examples of how to use Guice in the context of MATSim: https://github.com/matsim-eth/abmt2021 .

The eutransit WDR repository, which contains the TRANSIT specific changes to the WDR code and blends WDR with eqasim, can be found at https://github.com/matsim-eth/eutransitWDR/tree/master

5 Conclusion

The outcome of this deliverable has been a set of tools that will enable transport modellers and planners to evaluate the potential effects of innovative intermodal concepts for long-distance passenger transport. These tools include:

- J-TAP, a framework for developing activity-based long-distance travel demand models that include intermodal routing, enabling the evaluation of strategic intermodal concepts and solutions, such as an integrated air-rail schedule, integrated ticketing, or infrastructure improvements such as adding an HSR station at an airport.
- The enhancement of MATSim's WithinDay Replanning functionality to allow mode choice, not just route choice, when agents are allowed to change their plans during the traffic simulation. This provides a flexible "sandbox" for testing out tactical decision-making tools, such as the AMAN/DMAN Ground Tool. This enhancement of MATSim can also be used to evaluate the impact of passenger information systems and information dissemination strategies on passenger delay and network congestion during transport system disruptions.
- The development of the AMAN/DMAN Ground Tool, which allows flights to wait for passengers delayed by ground-transportation disruptions, minimising the number of passengers stranded at the airport, with minimal disruption to the flight schedules.
- The development of the Timetable Synchronisation tool, which allows schedulers to optimise the schedules of multiple modes (for instance, air and rail, or air, rail, and long-distance bus services) to provide optimal transfer times between the different modes, making intermodal trips more attractive.

This document has described these tools and the motivation to create these tools on a conceptual level (See Section 2) and has given a technical overview of the design and implementation of these tools and their use (See Section 3). For J-TAP and MATSim, Section 4 documents how to access these tools via Github.

References

[1] M. Janzen, “Simulating Annual Long-Distance Travel Demand,” 2019.

[2] C. Dobler and K. Nagel, “Within-Day Replanning,” in *The Multi-Agent Transport Simulation MATSim*, 2016, pp. 187-191.

[3] J. Rich and S. L. Mabit, “A long-distance travel demand model for Europe,” *European Journal of Transport and Infrastructure Research,* vol. 12, no. 1, pp. 1-20, 2011.

[4] A. Schäfer, J. B. Heywood, H. D. Jacoby and I. A. Waitz, Transportation in a climate-constrained world, 2009.

[5] G. Mattioli and M. Adeel, “Long-Distance Travel,” in *International Encyclopedia of Transportation*, Elsevier, 2021, pp. 272-277.

[6] R. G. Schiffer, Long-distance and rural travel transferable parameters for statewide travel forecasting models, TRB’s National Cooperative Highway Research Program (NCHRP) Report 735, 2012.

[7] J. Bose, G. Lee, J. Sharp, J. Memmott, M. Khan and E. Roberto, “A picture of long-distance travel behavior of Americans through analysis of the 2001 national household travel survey,” Transportation Research Board, 2004.

[8] Y. Jeong, A. Zvonkovic, Y. Sano and A. Acock, “The occurrence and frequency of overnight job travel in the USA,” *Work, Employment & Society,* vol. 27, no. 1, pp. 138-152, 2013.

[9] L. Aultman-Hall, C. Harvey, J. Sullivan and J. LaMondia, “The impact of long-distance tour generation and travel attributes on data collection in the United States,” *Transportation,* vol. 45, no. 3, pp. 875-903, 2016.

[10] N. McGuckin, J. Casas and M. Wilaby, “MI Travel Counts III Travel Characteristics Technical Report,” Michigan Department of Transportation, 2016.

[11] L. Aultman-Hall and H. Ullman, “Chapter 11 - Long-distance and intercity travel: Who participates in global mobility?,” in *Mapping the Travel Behavior Genome*, Elsevier, 2020, pp. 187-207.

[12] J. LaMondia, C. Bhat and D. Hensher, “An annual time use model for domestic vacation travel,” *Journal of Choice Modelling,* vol. 1, no. 1, pp. 70-97, 2008.

[13] C. Van Nostrand, V. Sivaraman and A. Pinjari, “Analysis of long-distance vacation travel demand in the United States: a multiple discrete–continuous choice framework,” *Transportation,* vol. 40, no. 1, 2013.

[14] C. Llorca, J. Ji, J. Molloy and R. Moeckel, “The usage of location based big data and trip planning services for the estimation of a long-distance travel demand model. Predicting the impacts of a new high speed rail corridor,” *Research in Transportation Economics,* vol. 72, pp. 27-36, 2018.

[15] G. D. Erhardt, J. Freedman, A. Stryker, H. Fujioka and R. Anderson, “Ohio Long-Distance Travel Model,” *Transport Research Record,* vol. 2003, no. 1, pp. 130-138, 2007.

[16] F. S. Koppelman and V. Sethi, "Incorporating variance and covariance heterogeneity in the Generalized Nested Logit model: an application to modeling long distance travel choice behavior," *Transportation Research Part B: Methodological,* vol. 39, pp. 825-853, 2005.

[17] R. Moeckel, R. Fussell and R. Donnelly, "Mode choice modeling for long-distance travel," *Transportation Letters,* vol. 7, pp. 35-46, 2015.

[18] K. G. Goulias, A. W. Davis and E. C. McBride, "Long Distance Travel and Destination Attractiveness," ITS Reports, 2018.

[19] F. S. Koppelman, "Multidimensional model system for intercity travel choice behaviour," *Transportation Research Record,* pp. 1-8, 1989.

[20] M. L. Outwater, M. Bradley, N. Ferdous, C. Bhat, R. Pendyala, S. Hess, A. Daly and J. LaMondia, "Tour-Based National Model System to Forecast Long-Distance Passenger Travel in the United States," 2015.

[21] S. Bekhor, C. Dobler and K. W. Axhausen, "Integration of activity-based with agent-based models. An example from the Tel Aviv model and MATSim," 2010.

[22] A. Horni, K. Nagel and K. W. Axhausen, MATSim Multi-Agent Transport Simulation, Ubiquity Press, 2016.

[23] C. Anda and S. A. Ordóñez Medina, "Extending the hidden Markov model for activity scheduling," 2017. [Online]. Available: https://www.research-collection.ethz.ch/handle/20.500.11850/183738.

[24] S. Oloruntoba, "SOLID: The First 5 Principles of Object Oriented Design," 2020. [Online]. Available: https://www.digitalocean.com/community/conceptual_articles/s-o-l-i-d-the-first-five-principles-of-object-oriented-design.

[25] L. Aultman-Hall, J. J. LaMondia, H. Ullman and M. Suender, "Surveying social network geography to model long-distance travel," *Transportation Research Procedia,* vol. 32, pp. 542-552, 2018.

[26] F. C. D. Märki and K. W. Axhausen, "Agent-based model for continuous activity palnning with an open planning horizon," vol. 41, no. 4, 2014.

[27] N. Leng and F. Corman, "The role of information availability to passengers in public transport disruptions: An agent-based simulation approach," vol. 133, pp. 214-236, 2020.

[28] M. Rahimi and F. Corman, "Quantifying the heterogeneity of direct and indi- rect effects of a public transport disruption with agent-based simulation," in *In Proceedings of the 20th Swiss Transport Research Conference (STRC)*, 2020.

[29] S. Hörl, M. Balac and K. W. Axhausen, "Pairing discrete mode choice models and agent-based transport simulation with MATSim," in *98th Annual Meeting of the Transportation Research Board*, Washington, D.C., 2019.

[30] S. Hörl, M. Balac and K. W. Axhausen, "A first look at bridging discrete choice modeling and agent-based microsimulation in MATSim," *Procedia Computer Science,* vol. 130, pp. 900-907, 2018.

[31] S. Hörl and M. Balac, "Introducing the eqasim pipeline: From raw data to agent-based transport simulation," *Procedia Computer Science,* vol. 184, pp. 712-719, 2021.

[32] S. Hörold, “Instrument for the Evaluation of the Quality of Mobility Information, original in German "Instrumentarium zur Qualitätsevaluation von Mobilitätsinformation",” Springer Fachmedien, Wiesbaden, 2016.

[33] M. Ben-Akiva, A. de Palma and I. Kaysi, “Dynamic Network Models and Driver Information Systems,” *Transportation Research Part A,* vol. 25A, no. 5, pp. 251-266, 1991.

[34] C. Dobler, “Travel Behavior Modelling for Scenarios with Exceptional Events - Methods and Implementations,” Zürich, Switzerland, 2013.

[35] X. Li, C. Jiang, K. Wang and J. Ma, “Determinants of partnership levels in air-rail cooperation,” *Journal of Air Transport Management,* vol. 71, pp. 88-96, 2018.

[36] AIRail, 2008. [Online]. Available: https://web.archive.org/web/20080731234940/http://www.bahn.de/international/view/en/prices/air_passangers/airail.shtml. [Accessed 25 November 2021].

[37] Air France, “Billet d’avion + train : prendre l’avion et le TGV pour son voyage - Air France,” [Online]. Available: https://www.airfrance.fr/FR/fr/common/resainfovol/avion_train/reservation_avion_train_tgvair_airfrance.html. [Accessed 25 November 2021].

[39] TAP Air Portugal, “Rail & Fly - Travel by train and plane,” [Online]. Available: https://www.flytap.com/en-gb/other-bookings/rail-fly. [Accessed 25 November 2021].

[40] Singapore Airlines, “Rail-Fly,” [Online]. Available: https://www.singaporeair.com/en_UK/us/plan-travel/intercity-transfers/rail-fly/. [Accessed 25 November 2021].

[41] Paris Aeroport, “Enregistrement à l’aéroport,” [Online]. Available: https://www.parisaeroport.fr/passagers/preparation-vol/enregistrement. [Accessed 26 November 2021].

[42] D. Delahaye, S. Chaimatanan and M. Mongeau, “Simulated annealing: From basics to applications,” in *Handbook of metaheuristics*, Springer, 2019, pp. 1-35.

[43] S. Chaimatanan, D. Delahaye and M. Mongeau, “A methodology for strategic planning of aircraft trajectories using simulated annealing,” in *ISIATM 2012, 1st International Conference on Interdisciplinary Science for Air traffic Management*, 2012.

[44] G. Scozzaro, D. Delahaye and A. Vela, *Noise Abatement Trajectories for a UAV Delivery Fleet,* 2019.

[45] J. Ma, “Optimisation du trafic aérien dans de grands aéroports,” Université Toulouse 3-Paul Sabatier, 2019.

[46] P. Belobaba, A. Odoni and C. Barnhart, The global airline industry, John Wiley & Sons, 2015.

[47] R. W. Simpson, “Scheduling and routing models for airline systems,” Cambridge, Mass., 1969.

[48] A. Ghobrial, N. Balakrishnan and A. Kanafani, “A heuristic model for frequency planning and aircraft routing in small size airlines,” *Transportation Planning and Technology,* vol. 16, no. 4, pp. 235-249, 1992.

[49] D. Teodorović and E. Krčmar-Nožić, “Multicriteria model to determine flight frequencies on an airline network under competitive conditions,” *Transportation Science,* vol. 23, no. 1, pp. 14-25, 1989.

[50] M. M. Etschmaier and D. F. Mathaisel, “Airline scheduling: An overview,” *Transportation Science,* vol. 19, no. 2, pp. 127-138, 1985.

[51] A. Levin, “Some fleet routing and scheduling problems for air transportation systems,” Cambridge, Mass., 1969.

[52] B. Rexing, C. Barnhart, T. Kniker, A. Jarrah and N. Krishnamurthy, “Airline fleet assignment with time windows,” *Transportation science,* vol. 34, no. 1, pp. 1-20, 2000.

[53] A. Caprara, M. Fischetti and P. Toth, “Modeling and solving the train timetabling problem,” *Operations research,* vol. 50, no. 5, p. 851-861, 2002.

[54] K. Ghoseiri, F. Szidarovszky and M. J. Asgharpour, “A multi-objective train scheduling model and solution,” *Transportation research part B: Methodological,* vol. 38, pp. 927-952, 2004.

[55] M. SteadieSeifi, N. P. Dellaert, W. Nuijten, T. Van Woensel and R. Raoufi, “Multimodal freight transportation planning: A literature review,” *European journal of operational research,* vol. 233, pp. 1-15, 2014.

[56] A. Tirachini, D. A. Hensher and J. M. Rose, “Multimodal pricing and optimal design of urban public transport: The interplay between traffic congestion and bus crowding,” *Transportation research part b: methodological,* vol. 61, pp. 33-54, 2014.

[57] P. Knoppers and T. Muller, “Optimized Transfer Opportunities in Public Transportation,” *Transportation Science,* vol. 29, p. 101, 1995.

[58] X. Dou, Q. Meng and X. Guo, “Bus schedule coordination for the last train service in an intermodal bus-and-train transport network,” *Transportation Research Part C: Emerging Technologies,* vol. 60, pp. 360-376, 2015.

[59] Y. Ke, L. Nie, C. Liebchen, W. Yuan and X. Wu, “« Improving Synchronization in an Air and High-Speed Rail Integration Service via Adjusting a Rail Timetable: A Real-World Case Study in China,” *Journal of Advanced Transportation,* vol. 2020, 2020.

[60] A. Marzuoli, “ Improving disruption management with multimodal collaborative decision-making: A case study of the Asiana crash and lessons learned,” *IEEE Transactions on Intelligent Transportation Systems,* vol. 17, pp. 2699-2717, 2016.

[61] M. Wardman, “Public transport values of time,” *Transport Policy,* vol. 11, no. 4, pp. 363-377, 2004.