Master Thesis

Risk Analysis in Synchro-modal Logistics Networks

Yuanyuan Li

Supervised by

Roberto Tadei
Denise Holfeld
Axel Simroth

Final Project Report for the
Master in Software Engineering

Department of Control and Computer Engineering
Politecnico di Torino
Italy, Turin

December, 2017

Risk Analysis in Synchro-modal Logistics Networks

Yuanyuan Li

Supervised by:

Roberto Tadei

Roberto Tadei, Department of Control and Computer Engineering

Denise Holfeld

Denise Holfeld, Fraunhofer Institute for Transportation and Infrastructure Systems IVI

Axel Simroth

Axel Simroth, Fraunhofer Institute for Transportation and Infrastructure Systems IVI

Abstract

SYNCHRO-NET is a Horizon2020 European research project that aims to overcome the stress due to increasing transportation distances, higher complexity, and vulnerability of modern supply chains. In SYNCHRO-NET consortium, considering the high uncertainty of synchro-modal transportation scenarios, Risk Analysis module is as an essential part to contribute to a cost-effective solution. The paper aims at the development of the decision in synchro-modal freight considering the potential risks (e.g. time delay, non flexibility and safety issues). Since the fastest route may not be the best when there are other safer or cheaper routes, it is necessary to include risk analysis considering different risk aspects.

The paper begins with an overview of the components in the SYNCHRO-NET system including the trip planner. Then the design of historical storage for storing data on execution and stakeholder's experience is introduced. Next, minimal sample size are analyzed to guarantee the balance of time efficiency and precision, and there are detailed explanation of two working phases (stochastic distribution functions and feature selection) in Risk Profiler (see chapter 4). There are general discussions about the structure, and working principle of Risk Analysis module including the solution approach named Monte Carlo Rollout which contains 2 players "playing the game" to simulate the situation of inter-modal transportation afflicted with several possible disturbances (time delay, path deviation and mode deviation). After that, the implementation of the Risk Analysis is described in detail, especially the meaning of four Key Risk Indicators. And then in the client side development, how to show to end users the values of four different risk aspects are introduced. Finally the main service functions are introduced.

This is an initial implementation for the Risk Analysis module. Hopefully, there will be more considerations in future developments, like more risk aspects may be accounted by analyzing the accumulated historical data, some other techniques may be taken to deal with growing data size and the historical storage will be analyzed to "dig" the more relevant features leading to possible risks.

With more data collected into historical storage, the prediction accuracy is expected to be higher.

Dedication

To all my friends, who make my German life more colorful than expected!

And to my professor Roberto Tadei at Politecnico di Torino. He consistently helps students to solve the puzzles in study and career. The door to Prof. Tadei's office was always open whenever I met a trouble or had a question about my research or course.

Also to my German supervisors Denise Holfeld and Axel Simroth who guided me to do the research with open mind. I am grateful to their valuable comments on this thesis.

Contents

List of Figures

Chapter 1

Introduction, Motivations and Goals

After decades of market expansion, global supply chain is rising rapidly. A reliable delivery of materials to market and manufacturing center has become crucial in global supply chain. However, there are several types of risks that impact the delivery routes of supply chains. For example, as reported in Safety & Shipping Review 2016[1], there were shipping loss of over 100 gross tons in 2015. With the risk of damage, loss, theft and delays, incidents are not just about the damage to goods, but also the impact on the price of goods sold, customer satisfaction, brand affections and so on. Taking into account the issues inter-modal operators usually meet, it is necessary to provide operators with the potential risks and their happening rates for all possible freight routes in advance, thus the operators could make a better choice.

The risk analysis in synchro-modal logistics networks is one part of the SYNCHRO-NET ecosystem. SYNCHRO-NET is a Horizon2020 European research project that aims to overcome the stress due to increasing transportation distances, high complexity, and vulnerability of modern supply chains. It emphasizes high synchro-modality and slow steaming practices (i.e., operating cargo ships at a significantly low speed compared to the designed speed, reducing fuel costs and greenhouse gas emissions [2]). In SYNCHRO-NET, the module named Real Time Monitoring enables the adaption during execution of planned and booked routes to current circumstances which guarantees reliability.

The SYNCHRO-NET ecosystem concerns de-stressing synchro-model freight by providing the cost-effective scheduling solutions in multi-modal transportation networks. To de-stress means to reduce the production emission and cost while increasing reliability and service levels. It considers several aspects: cost, environment impacts, risk and so on. Therefore, besides this decision support on an operative level, Booking module and Real Time module, a so-called Strategic Optimization Toolset provides logistics operators a decision support about routing and scheduling freight movements on truck, rail and ship on a strategic level. While planning a trip from A to B, the Simulator module is used to consider all relevant Key Performance Indicators (KPIs). Additional to these, Risk Analysis module provides Key Risk Indicator(KRIs) according to different risk perspectives of a route.

Given the high uncertainty of multimodal transportation scenarios, the aim of risk analysis is to get the common attributes that lead to risks. Through the analysis

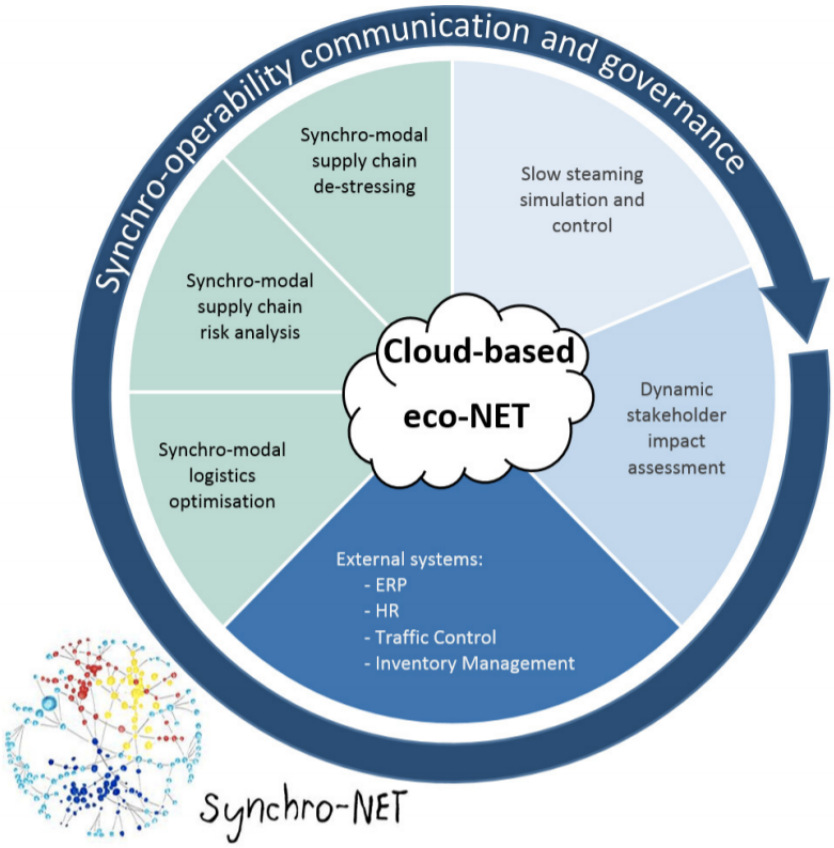

Figure 1.1: The modules of SYNCHRO-NET ecosystem.[3]

of historical data, Key Risk Indicator (KRIs) according to different risk perspectives are given for each alternative route. After that, we provide to users a clear view of the potential risks to alternatives. Then operators could make a wiser decision to choose the shipping route.

In this work, we focus on the simulation based risk analysis techniques developed to the purpose mentioned above.

Chapter 2

Risk Analysis as Part of the Synchro-NET Ecosystem

In the following we provide an introduction to the Risk Analysis approach applied to synchro-modal planning. The text is based on the work [4] conducted in the SYNCHRO-NET project.

2.1 Optimization modules and their interactions

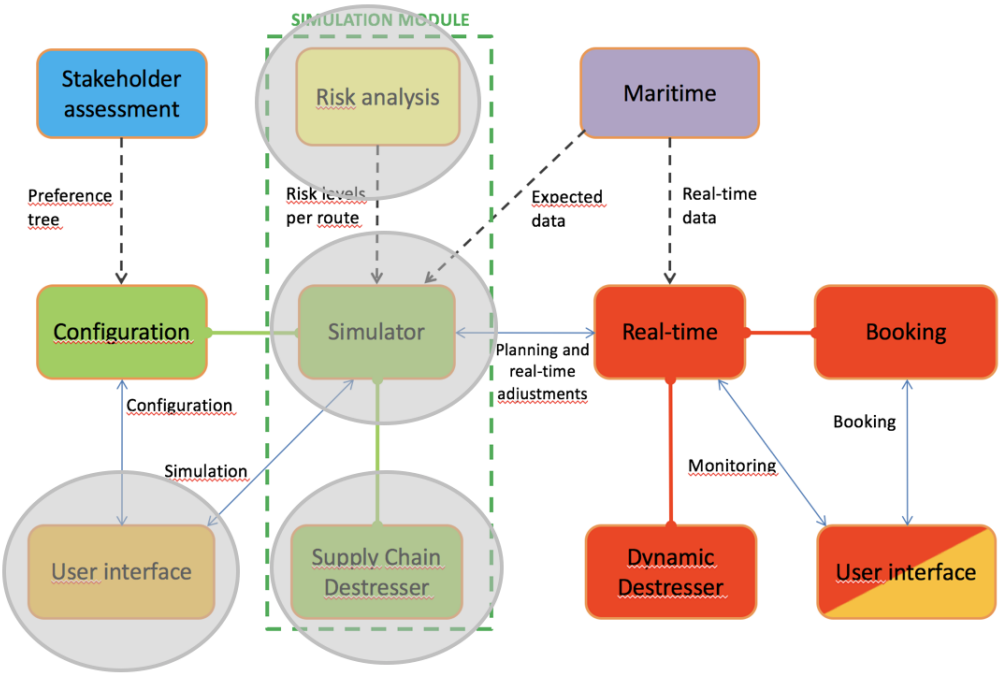

Figure 2.1: Optimization modules of SYNCHRO-NET and their interactions. The enclosed rectangles are where the paper focuses on.

To achieve the aim of SYNCHRO-NET mentioned in the introduction, the SYNCHRO-NET consortium is developing an integrated optimization and simulation system incorporating strategic and real-time logistics optimization, smart steaming ship simulation and monitoring, risk analysis, and stakeholder impact assessment.

The optimization process focused here, involves mainly three parts:

1. *Simulator* and *Supply Chain De-stresser (SCD)* encourage end-users to evaluate, assess, and select lower risk and lower emission options, more in general, options allowing the reduction of the stress in ports, hubs, and corridors congested by the traffic. They consider end-users requirements "Configuration" (given by Stakeholder Assessment) and two trip end-points A and B with dates, to return a list of potential routes associated with a set of Key Performance Indicators (KPIs. e.g. duration, length, cost, CO2, etc.) and a set of Key Risk Indicators (KRIs, e.g. time deviation, cost deviation, flexibility and safety).

2. *Real Time* (RT) module, which traces the transport on the selected route so if a trip is disturbed somewhere, the route can be easily adapted by the user. First, this will be detected promptly and the already existing route alternatives are checked. If all routes fail and none is feasible, a REST service on the Simulator is called to generate a new list of potential routes from the last feasible node to the destination node, by providing new release time and due date.

3. *Risk Analysis* (RA) module, which considers uncertainties along routes. With help of Monte Carlo Simulation, basic KRIs are provided to the Simulator. So, collected data is not only used to adjust distributed routes, but also to include this knowledge for planning trip

2.2 The Synchro-NET Trip Planner

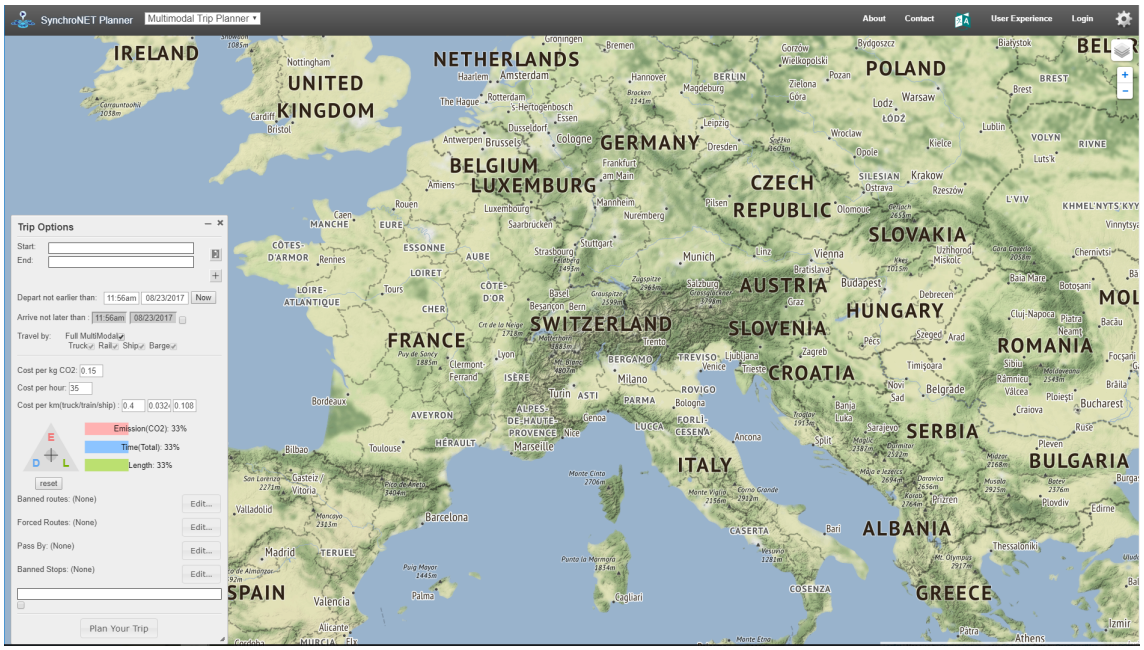

Figure 2.2: Synchro-NET Planner

OpenTripPlanner(OTP)[5] is an open source platform for multi-modal and multi-agency journey planning. It follows a client-server modal, providing several map-based web interfaces as well as a REST API for use by third-party applications. In the Synchro-NET project, OTP is used as the basis of the trip planner.

OTP consists of three basis components:

1. *Graph Builder*: building the graph.
2. *Routing Engine*: processing requests from GUI(the Graphical User Interface), extracting the requested data from the graph created by the Graph Builder, creating response and sending the response back to GUI.
3. *User Interface*: interacts with user to create requests and display response.

OTP was used as a basis and expanded within the project, so that users can select routes under the consideration of different risk aspect. But how are risks analyzed? Let's first have a look at the working flow.

2.3 Working Flow

The following picture shows the general working flow in the view of risk analysis:

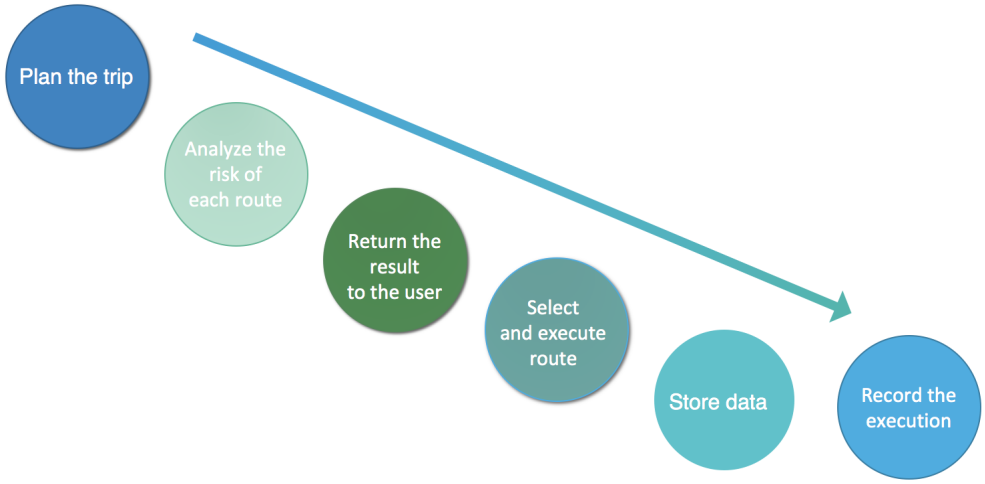

Figure 2.3: Working Flow of Risk Analysis

As shown in the picture above, user plans the trip by entering the starting and ending location, Several optimized planned routes are calculated and the Risk Analysis module is called for risk analysis. The risk of each route is analyzed and returned, together with the classical KPIs, to the user. Then user chooses one according to his own preferences. The selected route will be stored into a database. Finally, the execution will be recorded into the database for future analysis, too. This database is called Historical Storage for storing data on execution and end-users experience, which will be explained more.

To understand better the working principle of risk analysis, the internal and external structures are introduced in the next section.

2.4 Structure of Risk Analysis and Interaction with Other Modules

As shown in the figure 2.4, the Risk Analysis (RA) module is composed of three main components, Historical Storage (HS), Risk Analysis Tool (RAT) and Risk Profiler (RP).

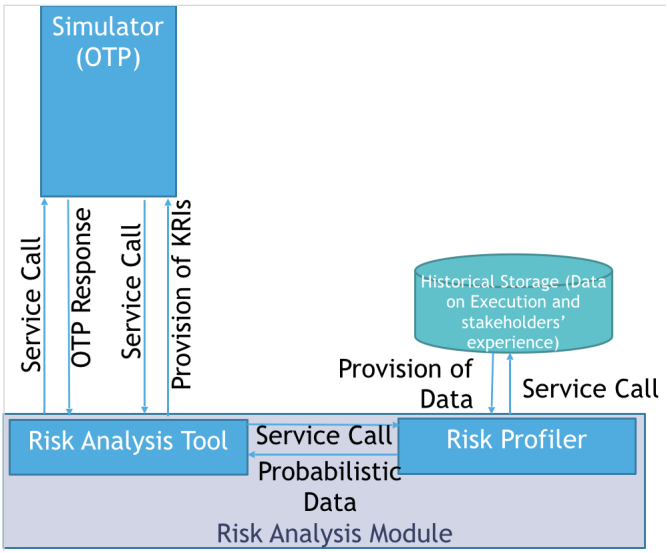

Figure 2.4: Structure of Risk Analysis and Interaction with other modules

The functions of the main components:

1. *Historical Storage (HS)*: collects data on execution and stakeholders' experience.
2. *Risk Profiler (RP)*: provides the stochastic data on the basis of historical data.
3. *Risk Analysis Tool (RAT)*: provides KRIs for each route using stochastic information from the Risk Profiler.
4. *Simulator*: calls RAT to get KRIs for routes, and returns the adapted route when needed.

   The components above are all integrated inside OTP.

The RAT receives a list of optimized alternative plans from Simulator, according to user preferences and constraints. With help of RP, Monte Carlo simulation using the probabilistic data generated basing on the historical data from the HS. After the simulations, the list of KRIs calculated will be returned to the Simulator.

2.4.1 Historical Storage

One route is composed by several links. One link is composed by 2 nodes and the path between 2 nodes. The RP collects data into HS about the planned links and the corresponding executed links, and compares the 2 links to detect the deviations.

The risk analysis needs to estimate probability distributions of potential disturbances, so stochastic models are needed, which should be based on historical data. Therefore, data from past routes is necessary. While discussing the use cases, it has turned out, that recording data during execution is not realistic. So we should come up with a new solution. We collect data from two sources, to compare planned and executed routes:

1. *the integrated Real-time Monitoring* (will be developed in other parts)
2. *the incident report panel* (see the figure below): it is integrated in the graphical user interface of OTP to record data about theft, loss or reliability. The recorded information will be used to calculate safety value of routes. The detailed calculation will be introduced later.

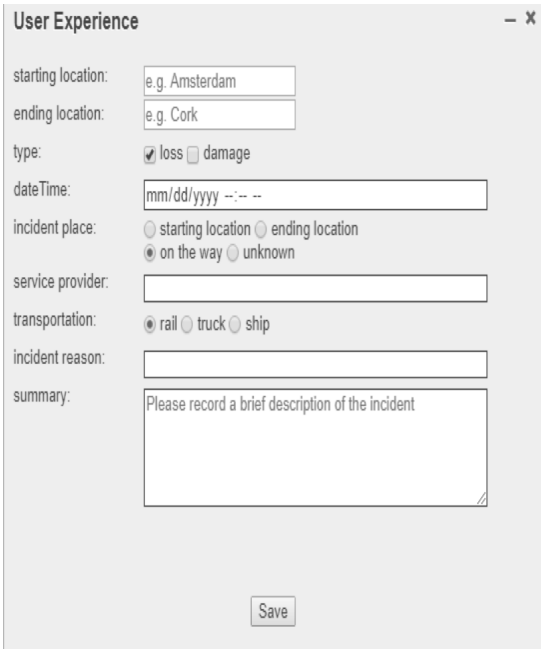

Figure 2.5: Accident Report Panel

2.4.2 Risk Profiler

The inputs of Risk Profiler are the booked and expected route and the execution of the scheduled route.

There are 3 use cases:

1. change of the transportation mode of a link.

   Indicating a problem with the transportation mode along the planned link. For example, some goods are planned to be transported using ship, but the cargo ship breaks down. Then the goods are transported by other transportation modes instead of using ship.

2. change in the departure time of a link.

   Indicating time delay in handling some problems which may happen to the first link or the transportation mode

3. change in the transportation path

   Indicating a problem happened to the link path. For example, some parts of the path are under construction, then there is be a need to change the path

We generate three different types of distribution functions to simulate these 3 scenarios (choosen according to experience), and the distribution functions will be introduced in next chapter in detail. The consequences of the three use cases result in re-scheduling and re-routing.

To simulate these use cases, there are 2 cases to be considered:

Case 1: when there is not enough historical data, different initial distribution functions (choosen according to experience) are used to generate random numbers for disturbances.

Case 2: when there is enough historical data to generate generate adequate distribution functions, we sample values for disturbances from the most relevant historical data. To identify the most relevant data for each link, we use some machine-learning feature selection techniques to select its most discriminative property. For instance, concerning the transportation of goods through Irish sea, *season* seems the most discriminative property (since in winter there are probably larger and more frequent time delays than in summer); if transport in London, the property *hour* maybe more important because of the frequently heavy traffic during rush hours.

2.4.3 Risk Analysis

The Risk Analysis Tool uses Monte Carlo simulations for each route received from the Simulator and then returns a set of risk indicators for each route. These risk indicators are displayed in the user interface. How is the risk simulation working?

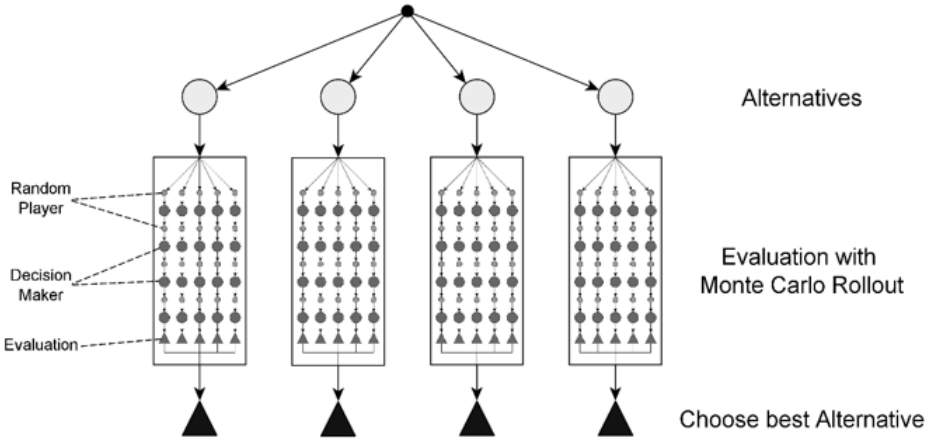

Figure 2.6: Schematic overview of the Monte-Carlo Rollout approach

For the simulation, a Monte-Carlo Rollout(MCRo) approach is used.

"It combines ideas from Rollout algorithms for combinatorial optimization and the Monte-Carlo Tree Search in game theory to handle sequential problems that are afflicted with uncertainties." In the two-player game, a so called *random player* generates different disturbances on each alternative route. As shown in the following figure, there are 3 disturbances generated in the simulation:

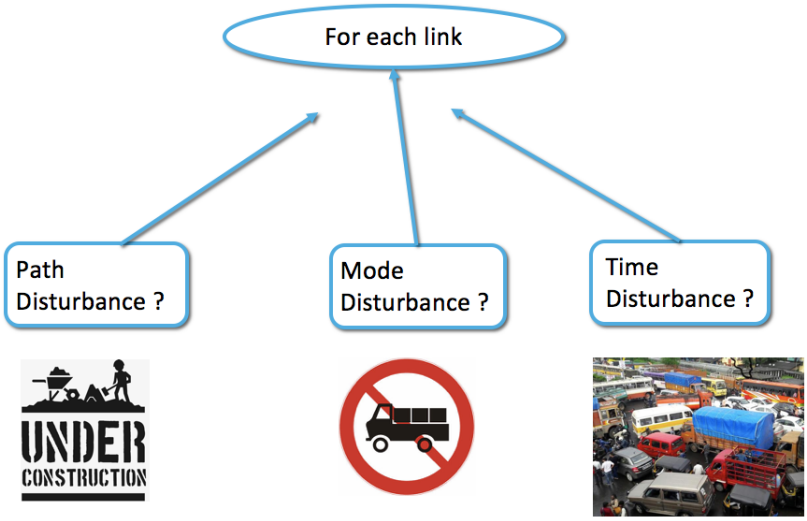

Figure 2.7: three different kinds of disturbances generated in the Monte Carlo simulation

A so-called *decision maker* checks the feasibility of the route after being afflicted with a certain disturbance. If the route is not feasible anymore (e.g., if the cargo is not ready to be boarded in time for the next transportation link due to time delay in the previous one), the RAT will send a request to the Simulator to ask for a new route for adaption (e.g. adapt with a new route with a different time schedule).

The following figure shows one scenario: after adding a time disturbance, the truck could not reach place B as planned, then the truck missed the scheduled transportation to place C. So a new route with a different time schedule will be asked for adaption.

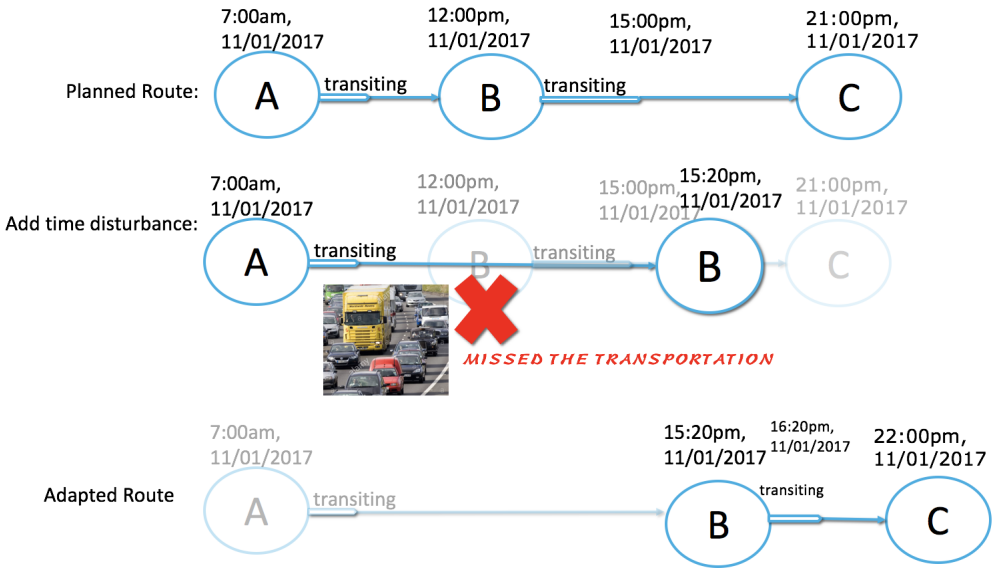

Figure 2.8: Schematic overview of unfeasible case and adaption

For each alternative route, the simulation procedure continues iteratively until the desired simulation time is reached. The long-term behaviour and robustness

against uncertainties of the alternative are analyzed and four key risk indicators (KRI) are returned according to the evaluations.

The KRIs returned are:

1. *Time deviation*: the average of all the generated time deviations
2. *Safety*: take the happening rate of incidents
3. *Cost deviation*: the average of all difference between planned cost and executed cost
4. *Flexibility*: the total waiting time in a route, which was caused by a disturbance.

In the following chapters the 3 parts of the RA module are described in detail. Here, the focus was on understanding the approach in general. The following chapters are in order of "data processing".

Chapter 3

Historical Storage

3.1 Design of Historical Storage

To trace the behavior of the freight routes for risk analysis, we need Historical Storage to store relevant data.

MySQL is chosen as the initial storage database for the following reasons:

- The data we analyze fit well into rows and columns, which means they are well structured, the relational database MySQL is suitable for storing structured data.
- In the initial phase, the data amount we will deal with is probably less than 1 TB, MySQL is good to process such amount of data.
- As a relational database, MySQL is easier to query for getting the needed data to analyze.
- It is free and easy to use. It also supports several development interfaces. Here we use JDBC which is an application programming interface for the programming language Java.

To well formalize the data and make the table structure consistent, we design database in the third normal form [6]. In database normalization, the third normal form reduces duplicated data and ensures the referential integrity, which improves database processing while minimizing storage costs. In the following picture, it shows the designed tables in the database for storing data to do risk analysis:

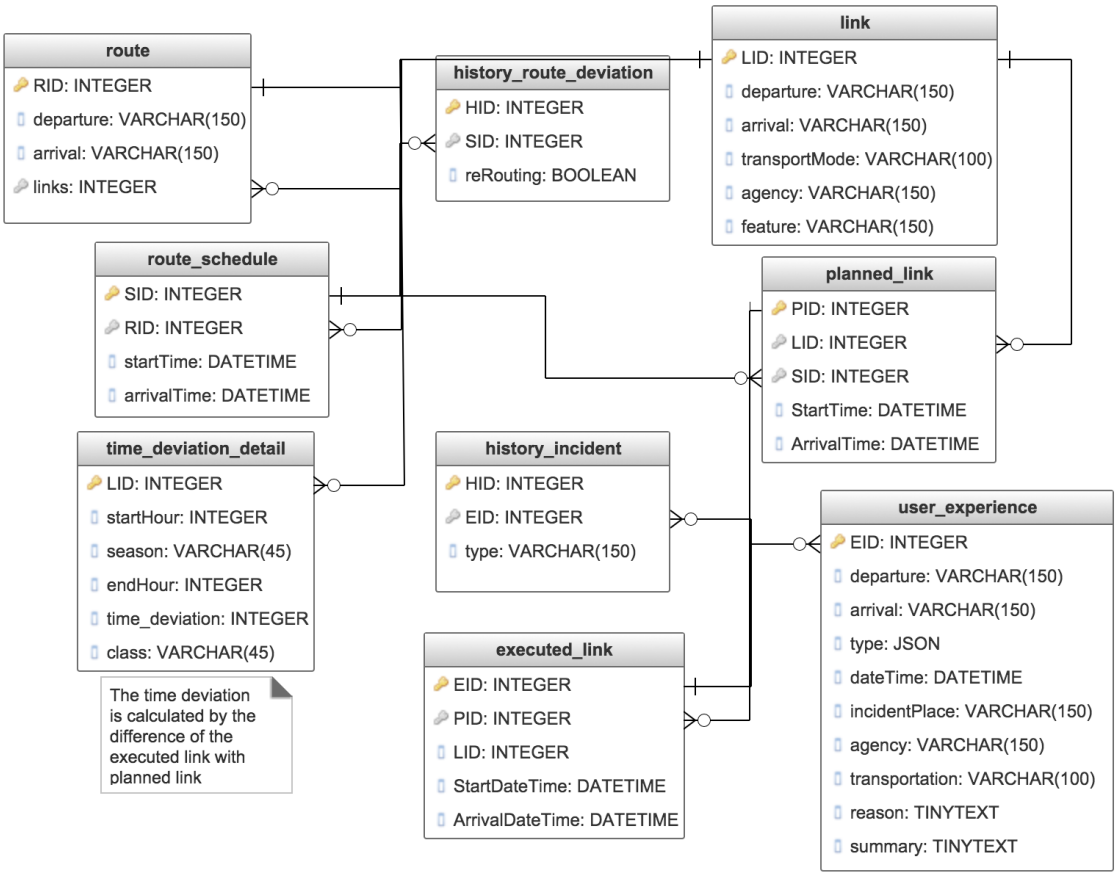

Figure 3.1: database structure

Since a route is composed by several links, the table named *route* is designed including a set of links stored as JSON type. The table named *link* stores the basic components to identify a single link. With the LID working as the foreign key, the tables named *planned_link* and *executed_link* store the data related to schedules and executions respectively. In addition, the table *planned_link* stores attribute *SID* in refer to the scheduled route, and the attribute named *PID* stored in table *executed_link* in refer to the link planned.

The time deviation stored in table *history_time_deviation* is the time difference between executed link and planned link. It is calculated by comparing the difference between the planned arrival time with the actual arrival time. The table *history_incident* records the data related to damage or loss happened in the freight. The table *history_route_deviation* stores information about rerouting. The table *time_deviation_detail* records the properties and time deviation of each link for feature selection which will be introduced in detail later.

If data amount of a link is not enough, an initial solution approach is used as described in the following section.

3.2 Working Flow about Database Storage

The storing process happens as described in the following:

User plans the trip by selecting the starting and ending location, and chooses one possible route from a list of choices. Firstly, the route and the links inside the

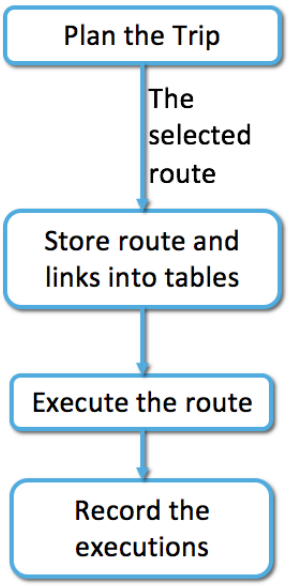

Figure 3.2: the working flow of database storage

selected route are stored into the corresponding tables. The scheduled route is stored inside the table named *route_schedule* (the startTime and arrivalTime are storing for simplicity rather than searching from table *planned_link*). After the execution of the route, the executed data is recorded into the other tables related to the executions.

In such a way, data is collected to serve as a basis of stochastic models for analyzing risks.

Chapter 4

Risk Profiler

4.1 Design of Risk Profiler

Figure 4.1: Logical Components of Risk Profiler

In the figure above, it shows the three logical components of Risk Profiler. It uses Java as main programming language to take and put data in the data storage directly. It also uses R language for statistical modeling (create stochastic distribution functions and sample random data) with the data provided from data storage. The reason that we use R to do statistical modeling are the following:

1. R has a large variety of statistical libraries, which provide more elaborate analysis and visualization tools;

2. Java has rich functionality to write business logic, but it is not efficient for statistical modeling. There is a R package called *rJava* allowing embedding basic R snippets in Java, and *Rserve* creating an R server which accepts request from Java code and returns response back to Java.

There are also some problems using R:

1. Since R does all its work in memory, the data amount is limited to the amount of RAM of the server. For solving this potential problem, there is a need to use other techniques like RHadoop which is designed for using R with Hadoop[7] for big data processing.

2. R was designed for data scientist in mind, not computers, then R is slower than Java or Python. There are some alternative R implementations improve the speed of R, like pqR[8]. However, the integration with R should be also checked.

Generally, a lowest bound needs to be defined for the amount of data necessary to generate well-defined stochastic information.

4.2 Analysis of Sample Size

To get a precise prediction, there should be enough data. But how much data is enough? So the key is how to determine the sample size considering the space and time efficiency.

Approach: To run R inside Java applications, we use JRI which is an interface loading R dynamic library into Java and provides a Java API to R functionality. We utilize JRI to draw plots for comparing the curve made by the input function with the one made by the output function. Here we take the example of time deviations of truck.

- *Input function*(represented with R function) is normal distribution function with mean *20*, standard deviation *40*:

  pnorm(x, mean=20, sd=40)

- *Output function*(represented by R function) is empirical distribution function, timeDevs is a vector filled with time deviation values of truck

  ecdf(timeDevs)

The plots with different sample sizes are the following:

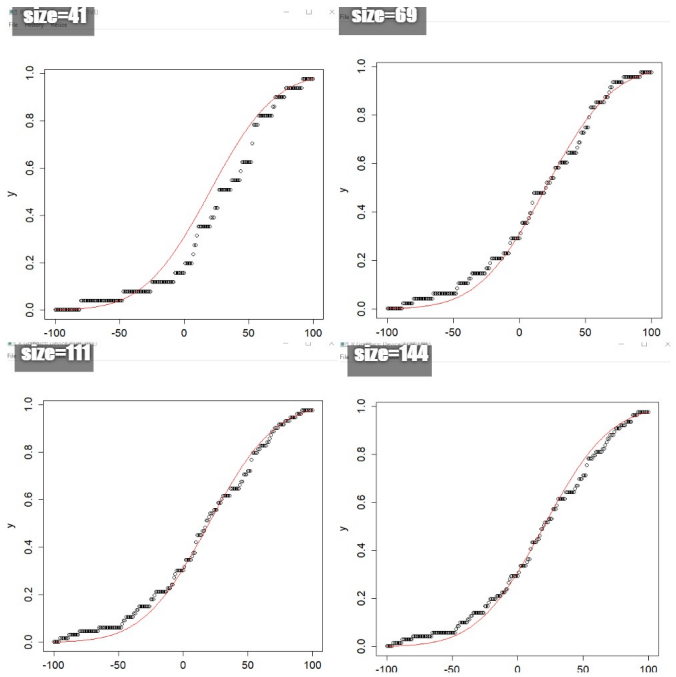

Figure 4.2: time deviations with size from 41 to 144

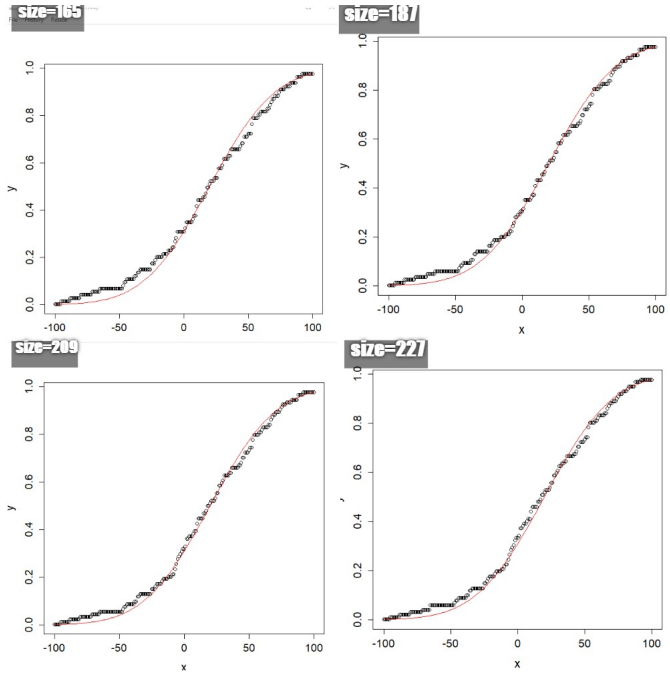

Figure 4.3: time deviations with size from 165 to 227

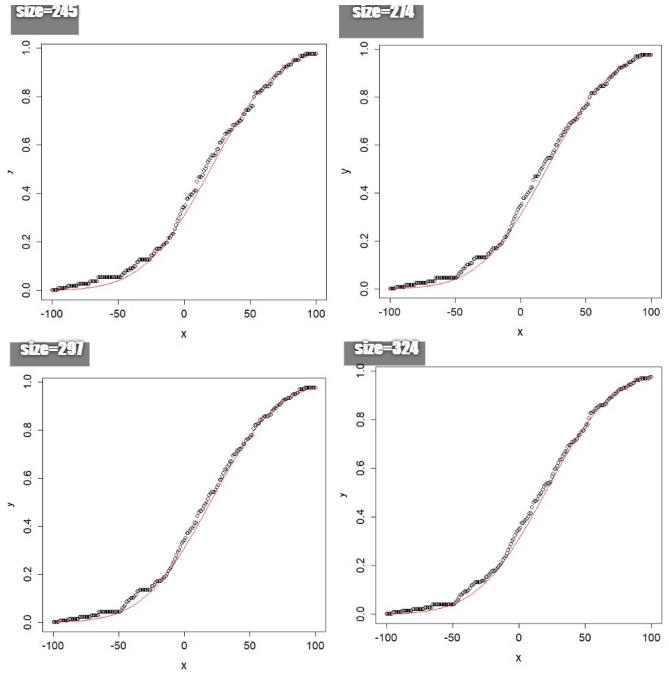

Figure 4.4: time deviations with size from 245 to 324

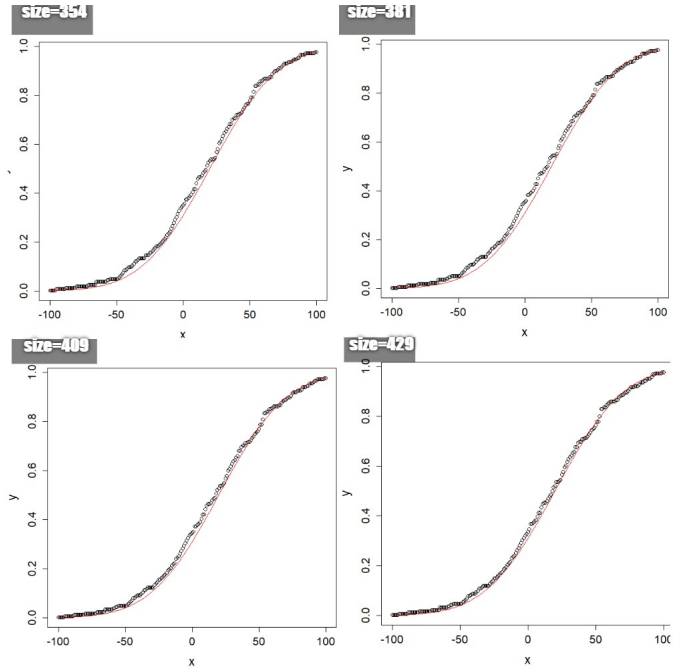

Figure 4.5: time deviations with size from 354 to 429

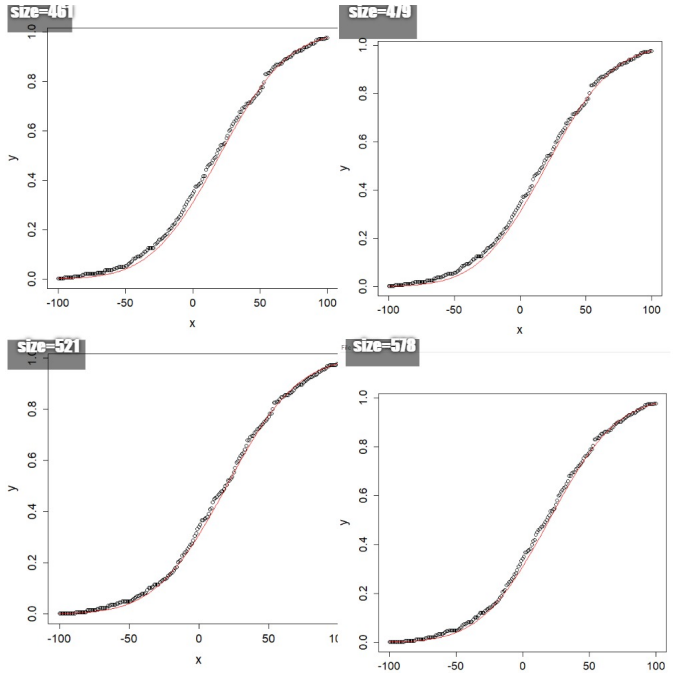

Figure 4.6: time deviations with size from 461 to 578

Through the plots, we can find from size 41 to 69, the 2 curves become more and more similar, then from size 69 to 578, the change is not obvious. So it is at least to take sample size at about 69.

4.3 Phase 1: Initial Stochastic Distribution Function

As mentioned before, there is no enough historical data in the initial phase, so we assume different stochastic distribution functions on experience. In the following subsections, the stochastic distribution for three disturbances will be introduced.

4.3.1 Time Deviation

In the initial step, to simulate the situation about some unavoidable time delays occurred in the inter modal transportation (e.g. traffic congestion), we create distribution functions separately for different transportation modes, since using different transportation mode for same route, the effect is probably different:

- **Truck**:

  It is assumed that either a small or a very large delay occurs, so we use the normal distribution to simulate the distribution of time deviations of truck

  In the following graph, it shows there is a highest frequency to have time deviation with number 20 (the mean of the distribution is set to 20)

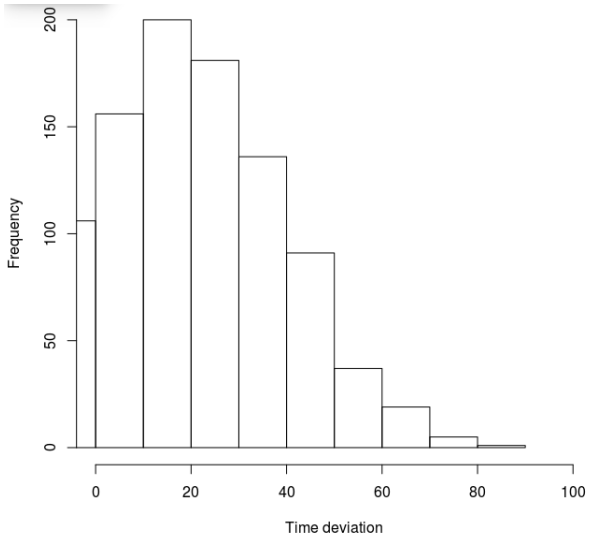

Figure 4.7: Distribution of Time Deviations for truck

  Here we take 20 as standard deviation, the mean value is determined by the duration of the link. $mean = 20 * duration/200$, it means for 1 unit(200 minutes) traveled, the average delay is 20 minutes.

  In the Monte Carlo simulation, a random time deviation(n=1) will be generated using rnorm(n=1, mean=20*duration/200, stdev=20).

- **Train**:

  For small delays of trains, it can be a few minutes. When a time slot is missed, the delay can be over one hour. So we choose a bimodal distribution which

can have more than one peak, meaning there are more than 1 frequent value. Here, the bimodal is combined by normal and gamma distributions.

normal distribution: mean=3, standard deviation = 3

gamma distribution: shape=50, scale=1.2

In the following graph(mean=20), it shows there are two peaks, which represents 2 most frequent values.

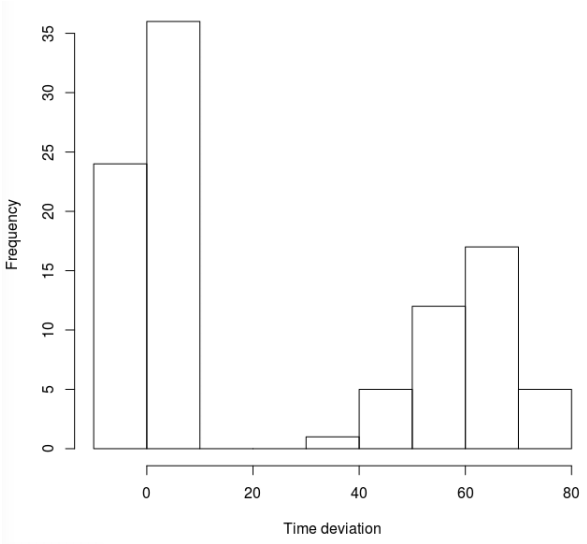

Figure 4.8: Distribution of Time Deviations for train

After generating a random number with the bimodal distribution, the time deviation is calculated by concerning 200 minutes as 1 unit.

- **Ship**:

  Ship is divided into 2 categories: Fast and Slow. In case of a "fast" planned ship route, the speed cannot be increased further, so a large delay is much more likely in comparison to the "slow" ship case.

  1. *slow ship*,

     Since a "slow" planned ship route can still be adjusted, in terms of speed increases, a small delay is much more likely than a large one. We also choose gamma distribution and to be more centered to small values.

     shape=7.5, scale=1

     In the following graph, it shows the values of time deviations are centered from 5 to 10.

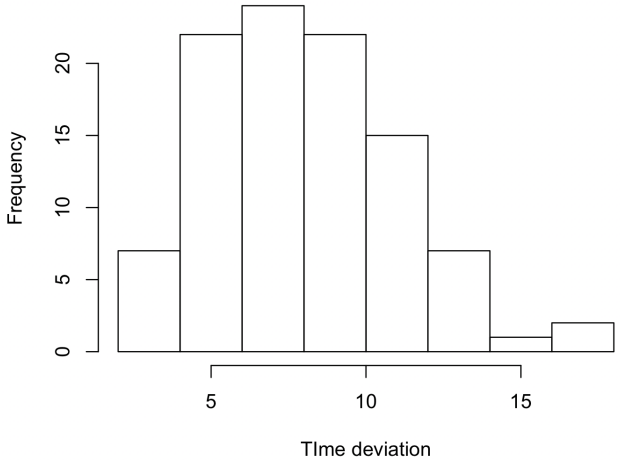

Figure 4.9: Distribution of Time Deviations for slow ship

After generating a random number with the gamma distribution, the time deviation is calculated by concerning 200 minutes as 1 unit.

2. *fast ship*,

   When the ship has already run with large speed, it can't speed up. So it is more likely to have a large delay. We choose gamma distribution and to be more centered to large values.

   shape=60, scale=1

   In the following graph, it shows the range of time deviations is centralized from 50 to 70.

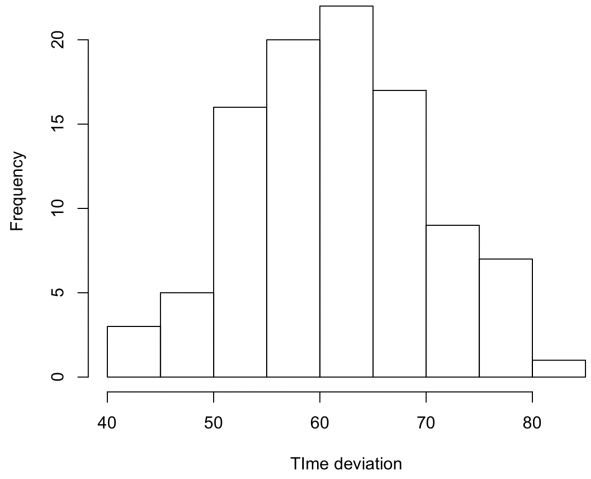

Figure 4.10: Distribution of Time Deviations for fast ship

After generating a random number with the gamma distribution, the time deviation is calculated by concerning 200 minutes as 1 unit.

4.3.2 Mode Deviation

As an initial assumption, the probability of a necessary transport mode change is assumed to be identical for all links. To simulate the situation whether it is necessary to change the transportation mode, because some transportation tools break down or other problems, or not, we use the inversion method[9]. The process is as following: we generate one random number from a uniform distribution on the interval [0,1].Then transfer to Bernoulli distribution ($X \sim \beta(\pi)$), which is determined by one parameter π. If the value is less than $\pi(\pi = 0.05)$, then the result will be projected to be 1, which means the mode would be changed (in the Monte Carlo simulation, a new link with a different transportation mode will replace the old link), vice verse.

4.3.3 Path Deviation

To simulate the situation whether it is necessary to change the transportation path, because some paths are under construction or other problems, or not, we use the inversion method[9]. It works like mode deviation, but we assume there is a lower possibility to change path than change the mode and then we take $\pi = 0.025$. If the value is less than π, then the path would be changed (in the Monte Carlo simulation, a new link with a different routing path will replace the old link), vice verse.

4.4 Phase 2: Feature Selection

When there is enough data, the random data will be generated from the historical data set instead of generating from stochastic distribution function described before, to estimate the deviations in time, path and mode.

For example, when a user plans a trip, to estimate the time deviation of the link using truck, we select data only related to the links which use truck. Then sample one random value from the selected data as time deviation.

However, not only modes, but also some other factors may influence the result, like climate, rush hour and so on. So if enough data is available, we may need to come up with a more precise prediction considering each link separately.

To achieve a more precise model to predict time deviation, we decide to use the Machine Learning Technique *feature selection*

Each object has many different features, however, not all features are relevant which means we select data by considering only essential features. Feature selection is a process to select a subset of relevant features. Here we use this technique to choose only one feature of a link to get suitable samples for estimating time deviation.

- Derive data about properties and time deviation of links from tables *planned_link*, *executed_link*, *link* and then store the derived data into table *time_deviation_detail*. The following figure shows data examples stored in the table.

LID	startHour	endHour	season	time_deviation	class
650	0	23	Summer	13	Insignificant
651	2	14	Summer	52	Minor
652	0	21	Summer	12	Insignificant
653	0	13	Summer	57	Minor
653	5	18	Summer	125	Maior
654	11	2	Summer	118	Moderate
632	10	23	Summer	85	Moderate
655	3	5	Summer	136	Maior

Figure 4.11: Detailed Time Deviation Table

1. In the table, LID stands for a link which is specified by departure, arrival and transportation mode. *startHour*, *endHour*, *season* are properties of the link, meaning the happened starting, ending hours and season. *time_deviation* is the time delay of the link *LID* executed from time *startHour* to time *endHour* in season *season*.
2. There are three features(startHour, endHour, season) listed, which are derived from the existing attributes *start time* and *arrival time* happended in the actual execution. Value of *time_deviation* is not treated as a feature, but it is selected for generating predicted time delay according to the most discriminative feature.
3. Here, we categorize time deviation(t) into 4 classes:
   Insignificant: $t < 30min$;
   Minor: $30min <= t <= 60min$;
   Moderate: $60min < t <= 120min$;
   Major: $t > 120min$

- Then we use a Java library called *weka*, which is a collection of machine learning algorithms for data mining tasks and the algorithms can be directly called in Java code. We use the package to solve the feature selection task. The algorithm of feature selection first ranks the input which are a set of link features, then according to the rank of the importance, we take one feature when the data is enough, otherwise we take more than one features to have more data.
- We select time deviations only relates to the selected feature. For example: transport through the link from Cork to Le Havre by ship in summer, if the selected feature is season, then we will select the time deviations of executed links from Cork to Le Havre by ship happened in the same season with the queried link, which is summer.
- We sample one time deviation from the selected time deviations by using R function.

This is just the initial approach. With more data collected, we could "dig" more features of a link. It may include some other features like weather, visibility, road conditions and so on. In addition, we can introduce more techniques like cross-validation[10], to select the most informative feature for generating the most predictive model, therefore, the collected data is divided into a training set and a test set.

We validate the model created on the training data set to predict the class of test set. Then we choose the best model according to the score we got in the validation. With the prediction of the selected model on test data, we get the selected feature and then store it into database for use in the next time.

With the most informative feature stored in the database, we can select the feature for each link directly from the database without recalculating each time. With the collection of data, the situation of links can be changed. So to keep the validity and precision of the prediction, we could recalculate the feature and update it periodically. Thus we could improve performance by saving the calculation time in real time situation.

Combined with the calculation method of time deviation introduced previously, some pre-processing steps will be done.

In the following picture, it shows the procedures of sampling data by considering the most informative feature.

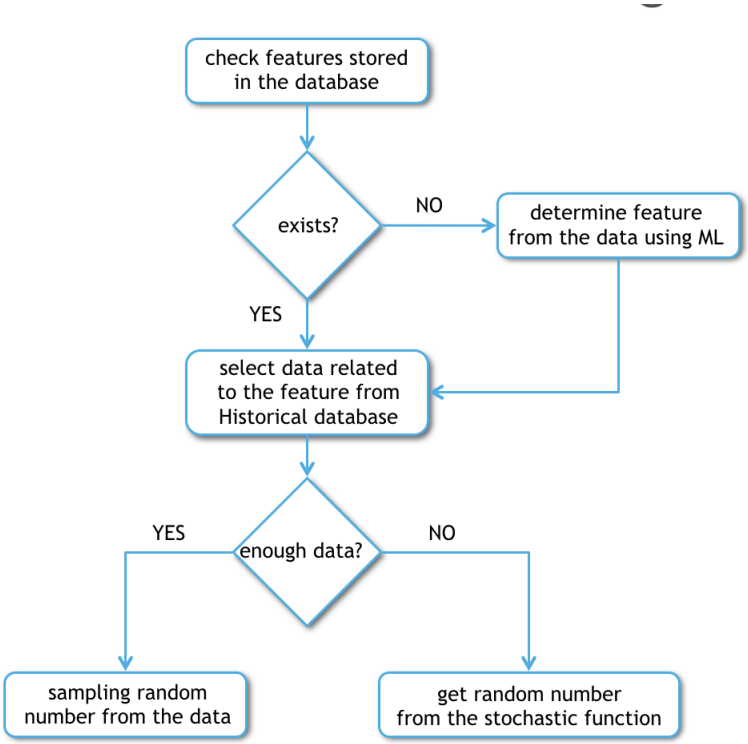

Figure 4.12: Procedures of Sampling Data by Considering the most Informative Feature

First check whether there is feature for the link already stored inside the database. If yes, get time deviation using the feature selected. If no, continue and check whether the data size related to the past executions of the link is enough or not. If yes, it will get time deviation using the feature selected; If no, it will get from the distribution function introduced for each transportation mode specifically.

The critical point is: not only the stored data related to the link must be enough, but also the selected data according to the feature must be enough. What if the selected data is not enough? We can have two approaches:

1. Using data resampling[11] technique. Data resampling is a method that consists of drawing repeated samples from the original data samples. There is no

size limitation in this method, however, the larger the size is, the more reliable the confidence interval generated by the method is.

2. Expand the range of data relating to more than selected features. If we select only one feature, the data amount is not enough, we can take second feature and take the data related to both the two features.

Chapter 5

Risk Analysis

5.1 Result Evaluation-KRI

The results of the Monte Carlo Simulation (MC simulations) are evaluated and summarized as key risk indicators.

For a route with n links, the key risk indicators are calculated as following:

- **Flexibility**: is the time spent for adaption along the route in ith simulation (by summing up the adaption time of n links in ith simulation). All the time derived from the waiting of new suitable movements, in the case of misses, is summed up and averaged over all simulations.

  t_i: average adaption time of the route in ith MC simulations;

  $MCwidth$: the number of risk simulations.

  $$flexibility = \frac{\sum_{i=0}^{MCwidth} t_i}{MCwidth}$$

  In addition, we also store $maxFlexibility$ calculated by summing up all maximum adaption time for each route, and minFlexibility calculated by summing up all minimum adaption time for each route.

- **Time deviation**: is the average of total time deviations for the whole route with respect to the expected one.

  T_i: time deviation of the route in ith simulation;

  $MCwidth$: the number of risk simulations.

  $$timeDeviation = \frac{\sum_{i=0}^{MCwidth} T_i}{MCwidth}$$

  In addition, we store minimum and maximum total time deviations occurred in the Monte Carlo simulation to record 2 extreme cases.

- **Safety**: the calculation is also based on the historical data. Through 2 approaches as introduced in the chapter 2.4.1.

  a. through real time monitoring,

b. through user input recorded in the user experience GUI. After user inputs data in the GUI, through AJAX (a web development technique for exchanging data with web server), the data is sent to the OTP server, then the incident is recorded into database.

Considering a similarity degree, in the sense of the more similar the planned conditions are compared to the incident conditions happened previously, the more likely the incident will happen again. In the initial stage, we only consider the agency and month as additional conditions to compare with. In the future stages, we can easily expand to more features like weather(e.g some damages are caused in a heavily snowing weather, it could happen again when it is predicted to snow heavily on the same route). With more and more collected data, the detailed information will be retrieved to make more precise decisions for calculation. The exact calculation method is as following:

- The initial safety value of every link is 100, which is maximum value.
- Count the frequency of the link(start, end, mode) that has been executed(F_i)
- If the frequency is not zero ($F_i \neq 0$), count the frequency of the link that has incidents. We have 3 criterions for each appearance compared to the query link:
  1. only the same transportation mode from same starting place to same destination, but the occurred month and agency are not same. In this case, it is weighted with 0.8 ($S_i = 0.8$)
  2. not only the same transportation mode from same starting place to same destination, but also either agency or month is equal and not both. Then it is weighted with 1.0 ($S_i = 1.0$).
  3. not only the same transportation mode from same starting place to same destination, but also same agency and month. Then it is weighted with 1.2 ($S_i = 1.2$).
- We use formulas to calculate the lost score ratio for each link($incident_j$ indicates lost score ratio of link j), which means happening rate:

  F: the execution times of the link j;
  m: the number of incidents happened to the link j;
  S_i: how many points are lost due to the ith incident for link j

  $$incident_j = 100 * \frac{\sum_{i=0}^{m} S_i}{F}$$

- We calculate the safety for the route(containing n links)

  $$safety = 100 - \frac{\sum_{j=0}^{n} incident_j}{n}$$

- **Cost deviation**: is the difference between planned and executed cost in a MC simulation. The executed cost is the total cost of all the executed links and the cost due to necessary adaptions caused by disturbances.

First for each MC simulation, we calculate cost due to disturbances of each link i of the route with

c_i: initial cost of ith link

cm_i: cost of the mode deviation for ith link

ct_i: cost of the time deviation for ith link

n: number of the running links for the route

cost of route in simulation j:

$$C_j = \sum_{i=0}^{n} (c_i + cm_i + ct_i)$$

then we calculate the cost deviation for the route compared to the cost of the initial planned route with:

c_i: initial cost of ith link

n: number of links in initial route

$$\Delta C = \frac{\sum_{j=0}^{MCWidth} C_j}{MCWidth} - \sum_{i=0}^{n} c_i$$

5.2 Risk Indicators Shown in the Client Side

After risks being analyzed, the trip plan is returned to the end users.

To show the distribution of risk values happened in MC simulations, we chose to represent the risk analysis result using bar charts with confidence intervals.

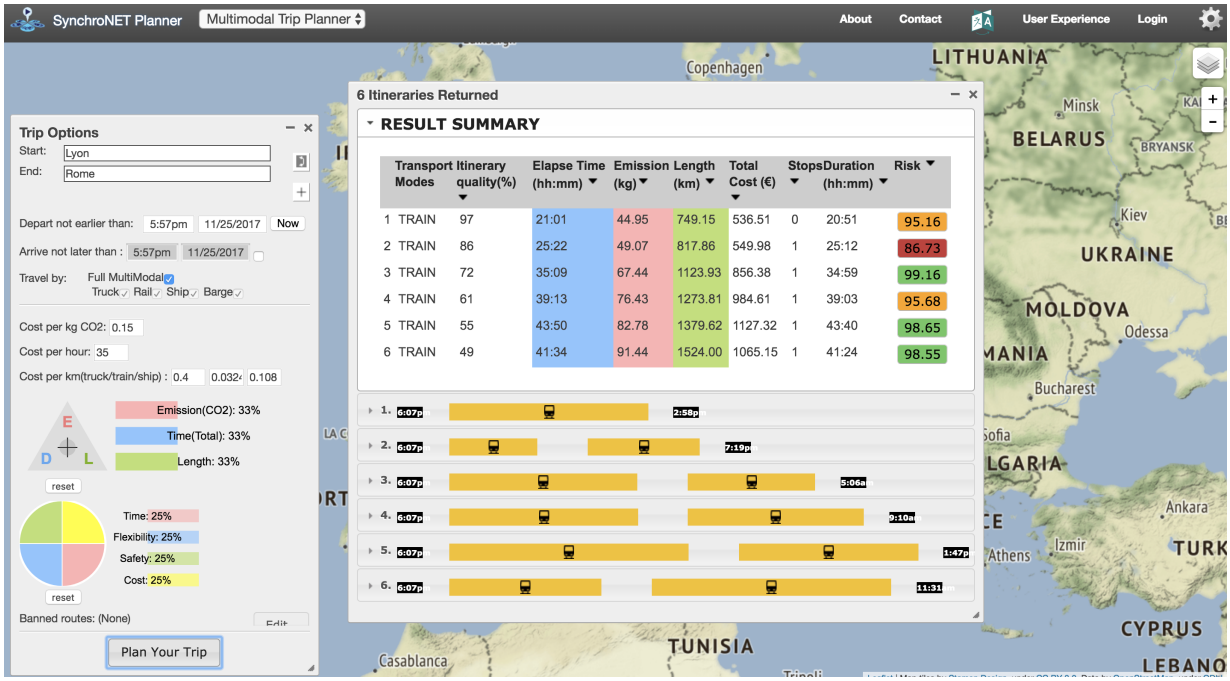

Figure 5.1: Risk Indicators shown to the client

In the figure above, we can see in the rightmost column, there are 3 colors:

- **Green** means values of all risk indicators are greater or equal to 90%;

- **Yellow** means not all values are greater or equal to 90%, but all values are greater than or equal to 70%;
- **Red** means not all values of different risk indicators are greater than or equal to 70%.

To scale all key risk indicators to the same level, the key risk indicators are represented in percent and normalized as stated in the following. Note that all the max values are 100, representing 100% and the bigger the value is, the better quality it stands for.

1. **Time Reliability**:

Taken into account that the durations are different for different routes, we normalize it with the duration of the route. It means how precise the real time execution is in comparison with planning time,

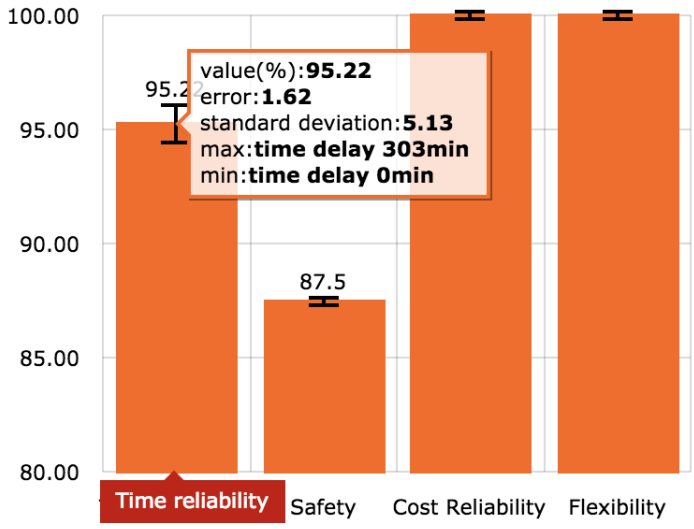

Figure 5.2: Risk Indicators shown to the client

time_deviation: as explained in chapter 5.1;
T: the route duration.

$$timeReliability = 100 - \frac{time\_deviation * 100}{T}$$

As shown in the figure 5.2, there are other four parameters:

1. *standard deviation*: quantify the dispersion of time delays occurred in the simulation. The higher the value, the more spread out the data is;
2. *error*: standard error, the standard deviation of its sampling distribution. The higher the value, the more spread out the data is. When the standard error is small, the data is said to be more representative of the true mean;
3. *max time delay*: for the same route, the maximum occurred time delay during the simulation;

4. *min time delay*: for the same route, the minimum occurred time delay during the simulation.

2. **Safety**:

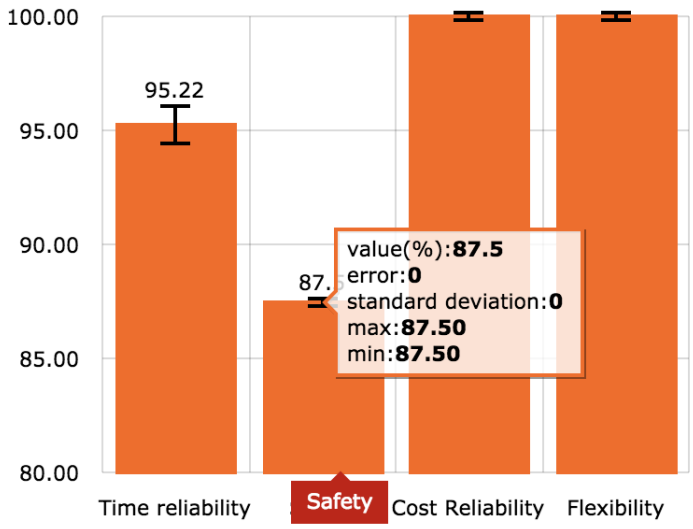

Figure 5.3: Safety Analysis Shown

Because the value of safety is calculated according to what happened in the past, which means it is a fixed value, so *value*, *max* and *min* are the same and standard error and standard deviation are 0.

3. **Flexibility**:

Figure 5.4: Flexibility Analysis Shown

The flexibility here is also normalized by the duration of the initial route.

1. T: the duration of the route.

2. *flexibility*: as explained in chapter 5.1.

$$flexibility = 100 - \frac{flexibility * 100}{T}$$

4. **Cost reliability**:

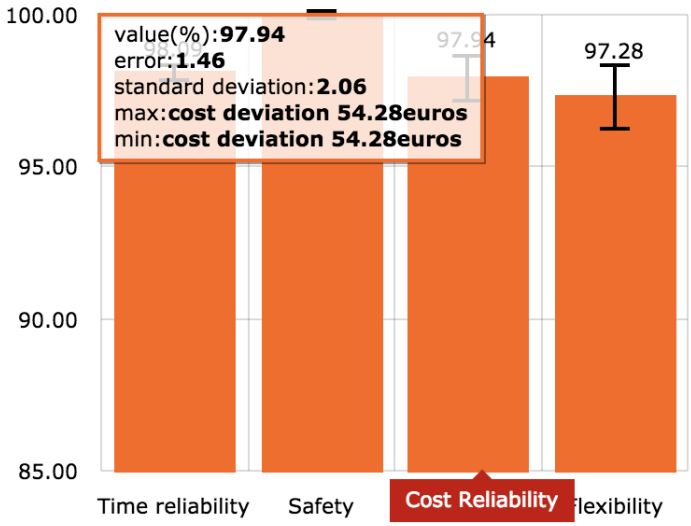

Figure 5.5: Cost Analysis Shown

Cost reliability is normalized by all the cost of links.

1. *cost_deviation*: as explained in chapter 5.1;
2. c_i: cost of ith planned link

$$costReliability = 100 - \frac{cost\_deviation * 100}{\sum_{i=0}^{n} c_i}$$

5. **Average risk rating**:

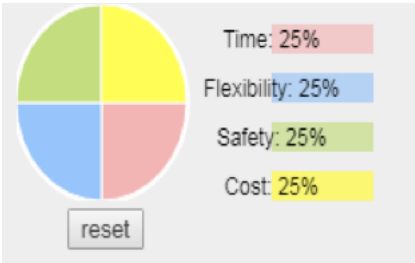

Figure 5.6: Weighting Pie

While some users believe "Time is money" and have a higher focus on time reliability, some other users think safety is more important. To enable users a differentiated view with different weighting preferences among the four risk indicators, the trip planner is designed to include a weighting panel. Users could rank the risks with respect to a common scale before clicking a button written *plan the trip*. By double clicking the pie with different colors representing different risk aspect, the proportions of the four risk indicators are changed under the constraint of the summation with a total amount of 1. The average risk rating is calculated by using the formula below:

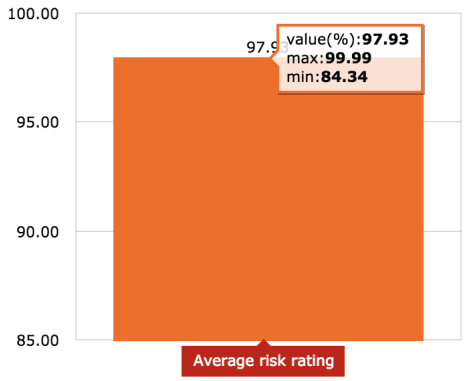

Figure 5.7: Average Risk Rating Shown

1. P_t: the proportion of time;
2. *timeReliability*: the value of time reliability;
3. P_f: the proportion of flexibility;
4. *flexibility*: the value of flexibility;
5. P_s: the proportion of safety;
6. *safety*: the value of safety;
7. P_s: the proportion of cost;
8. *costReliability*: the value of cost reliability

$$averageRiskRating = P_t * timeReliability + P_f * flexibility + P_s * safety + P_c * costReliability \quad (5.1)$$

As shown in the picture above, there are two additional parameters:

1. *max*: for the same route, the combination of max values of time reliability, flexibility, safety and cost
2. *min*: for the same route, the combination of min values of time reliability, flexibility, safety and cost reliability.

Chapter 6

Class Structure

6.1 Class structure for main service functions

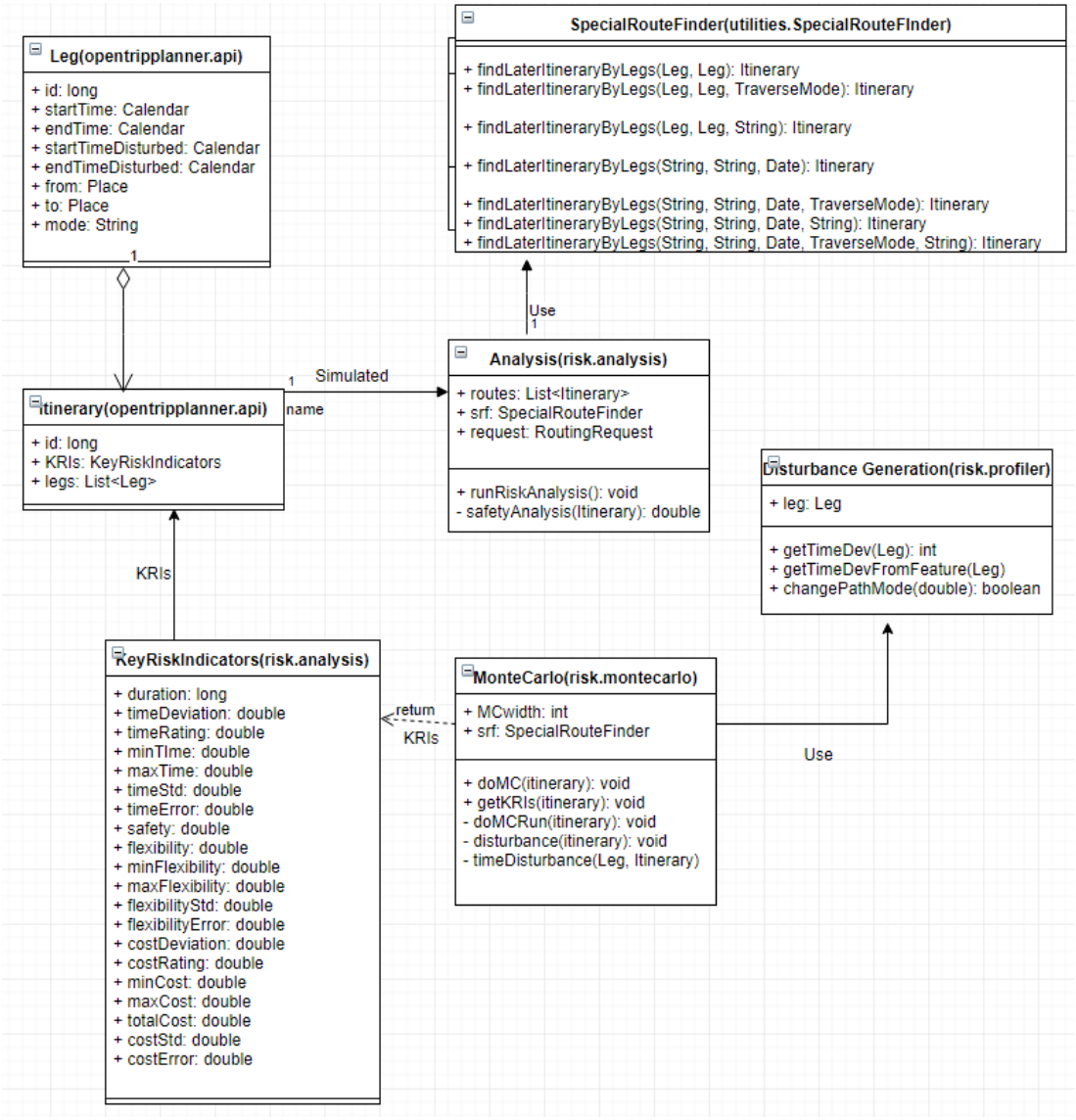

Figure 6.1: Class Diagrams related to risk analysis(1)

The data structures in the graph shown above:

- **Leg**:

It means a link. It is defined in the OTP package as a connection between 2 stops($from$ and *to*), using transportation mode *mode*. *StartTime* and *endTime* are planned time, *startTimeDisturbed* and *endTimeDisturbed* are time after being afflicted with possible disturbances during simulation.

- **Itinerary**:

It means a route. It is defined in the opentripplanner package. A sequence of legs defines a whole itinerary from the start to the end that user selects in the GUI.

- **KeyRiskIndicators**:

It is defined in the risk analysis package. Considering what operators care most, it contains mainly four key risk indicators representing different risk aspects: safety, flexibility, cost and time deviation. For showing clearly to users in the GUI, it also includes other parameters, like min and max values, to show the 2 edge cases occurred in the risk simulations to users.

- **Analysis**:

It is defined in the risk analysis package. It calls Monte Carlo simulation for getting time deviation, cost deviation, flexibility and do safety analysis here. The 3 parameters it includes:

1. *routes*: a list of itineraries to be analyzed
2. *request*: RoutingRquest which defines trip planning request with several parameters specified
3. *srf*: one instance of SpecialRouteFinder, it is for the route adaption in case of in-feasibility of the route during Monte Carlo simulation.

There are also two methods included:

1. **runRiskAnalysis**: call the MC simulation to analyze the risks and collect result.
2. *safetyAnalysis*: collect incident happened to the itinerary and calculate the safety scores.

- **SpecialRouteFinder**:

It is defined in utilities package. It includes three methods with same name but different parameters for getting new routes:

1. findLaterItinerary(String departure, String arrival, Date date): find a new route from *departure* to *arrival* not before date *date*, which is used when the previous link is disturbed, and it is not possible to catch next link on time, in that case, it is necessary to adapt the route, i.e. to find a new link with a different time schedule.

2. findLaterItinerary(String departure, String arrival, Date departureDate, String bannedStop): find a new route from *departure* to *arrival* not using the stop *bannedStop* and not before *departureDate*. It is used when the path is needed to be changed.

3. findLaterItinerary(String departure, String arrival, Date departureDate, TraverseMode mode): find a new route from *departure* to *arrival* not before date *departureDate*, without using transportation mode *mode*. It is used when there's a need to change the transportation mode.

- DisturbanceGeneration:

It is defined in risk profiler for generating disturbance to a link. It generates random time deviation from distribution functions(explained in previous section) and checks whether there is a need to change the transportation mode and routing path or not.

1. getTimeDev(Leg): generate time deviation. Firstly, it checks whether there is possibility to derive time deviation according to the most discriminative feature, if not, the time deviation will be generated from the initial distribution functions defined for different transportation modes.

2. changePathMode(double): generate a random number to estimate the need to change mode or path.

- **MonteCarlo**:

It runs the Monte Carlo Rollout simulation for each route/itinerary, to estimate the KRIs except for safety value.

The graph in the following page is the working flow of Monte Carlo simulation. In the simulation, for each leg (which means each link), firstly, the **mode disturbance** is generated. When there is a need to change the transportation mode, a new route with different transportation mode will be asked, the legs are updated and the additional cost for the change of transportation mode will be added.

If there is no need to change the transportation mode, the **path disturbance** is generated. After the disturbance, if there is a need to change the routing path, a new route with different routing path will be asked and the legs are updated.

If there is no need, the **time disturbance** will be generated and the cost of time delay will be added. After adding time delays, if the route is not feasible, a new route will be asked and the legs will be updated to continue the procedures with next leg. If the route is feasible, continue the procedures with next leg directly.

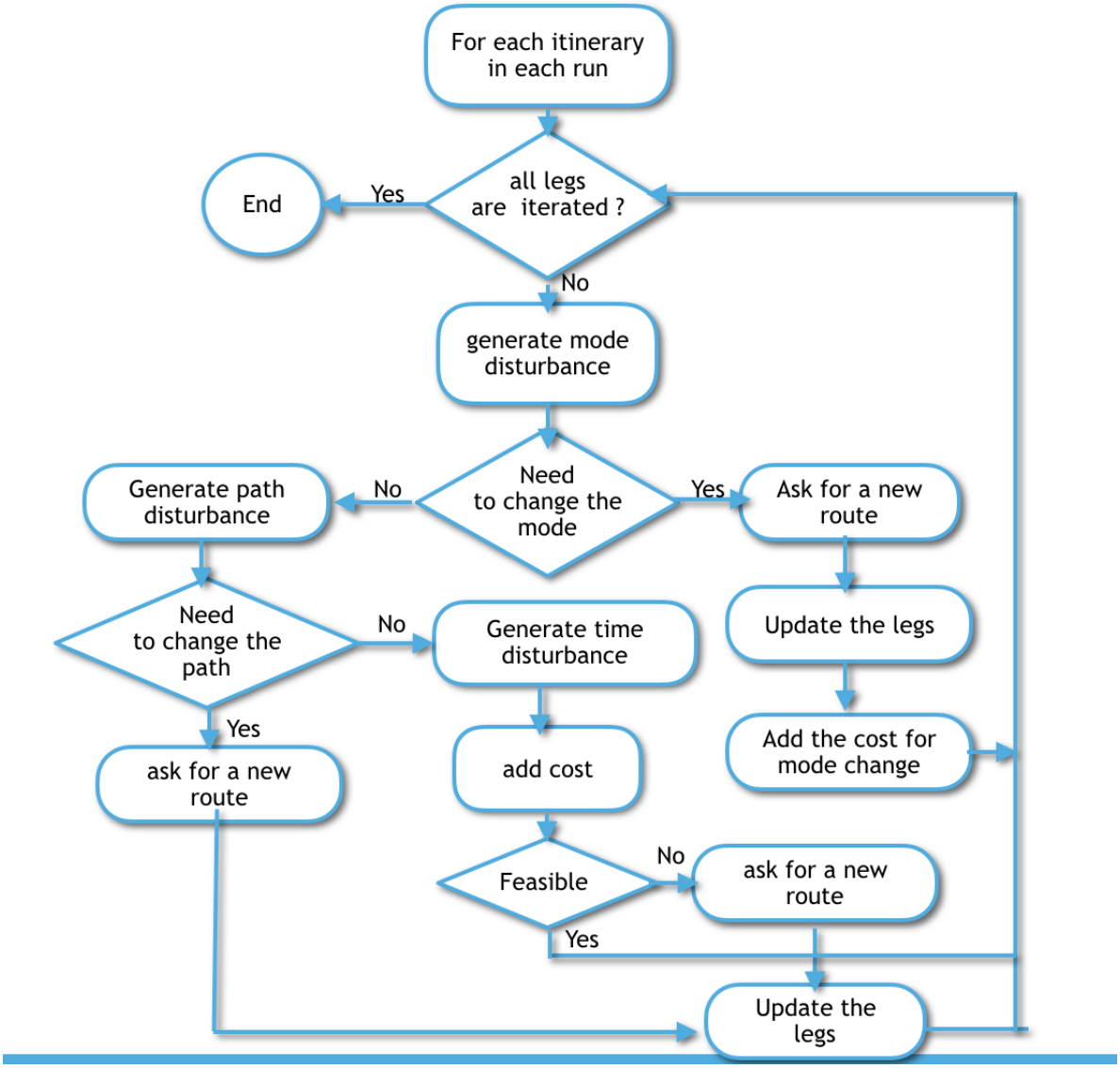

Figure 6.2: Flow chart of Monte Carlo simulation

- **RandomNumber**:

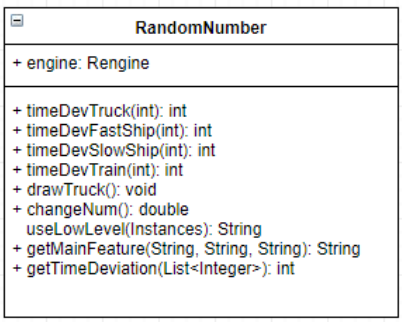

Figure 6.3: Class Diagrams related to disturbance generation

RandomNumber class is for generating random numbers, which represents time deviations and the necessity to change transportation mode.

The following functions get random numbers as time deviations from the stochastic distribution functions:

- Phase 1:
  1. timeDevTruck(int): generate time deviation for truck;
  2. timeDevFastShip(int): generate time deviation for fast ship;
  3. timeDevSlowShip(int): generate time deviation for slow ship;
  4. timeDevTrain(int): generate time deviation for train
- Phase 2:
  1. getTimeDeviation(List<Integer>): get the time deviation from the data selected which relate to the most discriminative property.

- **RiskCalculation**:

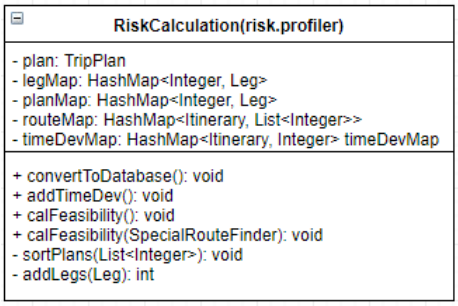

Figure 6.4: Class Diagrams related to risk analysis(2)

It is for storing data into MySQL database and adding time disturbance to see the route is still feasible or not after the disturbance. Those are just simulating the process of traveling through the links and testing the other functions.

- **RiskDataBaseManager**:

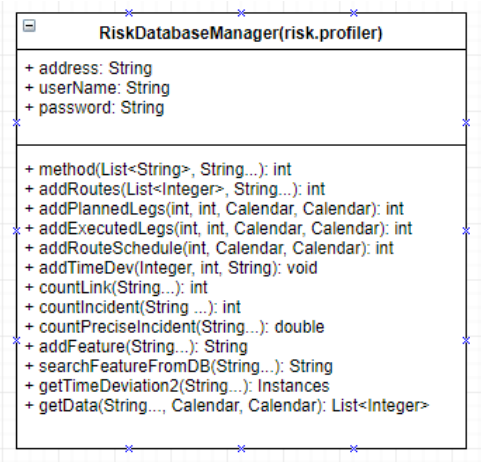

Figure 6.5: Class Diagrams related to risk database

It is defined in risk profiler. It uses JDBC interface to store data related to routes and user experience into MySQL database. It also counts the number of incidents happened and the number of all executions for each link.

The data structures above all together contribute to the service function for achieving the roles of Risk Profiler and Risk Analysis modules by means of Java, other open-source libraries for statistical analysis and MySQL database.

Chapter 7

Conclusion and Outlook

7.1 Conclusion

With the rising of global supply chain, to ensure the reliable delivery of materials to manufacturing centers or markets is very crucial. However, there are still no many mature trip planners in the market allowing inter modal ship planning and showing much information towards risks in multi-modal network.

In the paper, we have proposed a risk analysis approach, using Monte Carlo Rollout simulation algorithm, to support strategic robust decisions about freight transportation. The disturbances simulated in the Monte Carlo games are generated using different methods according to the amount of historical data. With more data collected, we can get a more precise prediction model. In the context of SYNCHRO-NET project, the Risk Analysis module has been implemented directly into the Strategic Optimization Toolset by means of Java and specific open-source libraries for statistical analysis.

7.2 Outlook

In the future developments, there are some points needed to be addressed or improved:

- Collect the properties of a link from shipping operators and increase the attributes of a link, then it could be more precise to predict the time deviation basing on the most relevant data set.
- Currently, KRIs are calculated only a posteriori with respect to the route optimization. A possible future research could be integrating risk attributes directly into optimization procedures.
- Besides storing most discriminative feature into database and update periodically to avoid recalculation each time, what other steps could improve the time efficiency in real time?

  It could be interesting to divide processes into two categories:

  1. Calculate in real time, which means the calculation is done once user requests a new trip plan.

2. Calculate in spare time, which means calculate some data during the time period that few users are online, like at midnight, then store the calculated data for future use.

- When there are large amount of historical data, it is not efficient to only use mySQL database and R technique for statistic modeling. Some other techniques like big data analysis and process might be needed.

Acknowledgements

Funding for this work was provided by the SYNCHRO-NET project, H2020-EU.3.4. – Societal Challenges – Smart, Green and Integrated Transport, ref. 636354.

Appendix A

Appendix

A.1 Abbreviations/Acronyms

HS	Historical Storage
RT	Real Time
RA	Risk Analysis
RA	Risk Analysis Tool
RP	Risk Profiler
OTP	Open Trip Planner
KRIs/KRIS	Key Risk Indicators
KPIs	Key Performance Indicators
SCD	Supply Chain Destresser

Bibliography

[1] Allianz Global Corporate and Specialty (AGCS). *Safety and Shipping Review 2016*. 2016. URL: http://www.agcs.allianz.com/insights/white-papers-and-case-studies/safety-and-shipping-review-2016/.

[2] M. Daniele et al. "Risk Analysis for synchro-modal freight transportation:the SYNCHRO-NET approach". In: *Odysseus* (2017).

[3] NMCI. *SYNCHRO-NET Solution Vision Document*. 2016.

[4] A. Simroth, D. Holfeld, and R. Tadei. "Risk Analysis for synchro-modal supply chain". In: *28th European Conference on Operational Research (EURO2016). Poznan, Poland* (2016).

[5] *OpenTripPlanner*. URL: http://www.opentripplanner.org/.

[6] *Third normal form*. URL: https://en.wikipedia.org/wiki/Third_normal_form.

[7] *RHadoop*. URL: https://github.com/RevolutionAnalytics/RHadoop/wiki.

[8] *pqR*. URL: http://www.pqr-project.org/.

[9] Matthia Templ. *Simulation for Data Science with R*. 2016. URL: https://www.packtpub.com/big-data-and-business-intelligence/simulation-data-science-r.

[10] *Cross Validation*. URL: https://en.wikipedia.org/wiki/Cross-validation_(statistics).

[11] *Data Resampling*. URL: http://www.statisticssolutions.com/sample-size-calculation-and-sample-size-justification-resampling/.