REDUCTION
2011-2014

Deliverable 3.1

Report Covering Requirements, Specification, Data-Flow Analysis, and the System Architecture

11 December 2012

Public Document

D3.1 Report Covering
Requirements, Specification, Data-Flow Analysis, and the System Architecture

Project acronym: REDUCTION

Project full title: Reducing Environmental Footprint based on Multi-Modal Fleet management Systems for Eco-Routing and Driver Behavior Adaptation

Work Package:	**3**
Document title:	Report Covering Requirements, Specification, Data-Flow Analysis, and the System Architecture
Version:	1.3
Official delivery date:	2012.03.01
Actual publication date:	2012.03.01
Type of document:	Report
Nature:	Public
Authors:	Ove Andersen(AAU), Benjamin B. Krogh(AAU), Kristian Torp (AAU) Christian S. Jensen (AU), Bin Yang (AU) Nick Christiansen (FD) Niels T. Larsen (FD)
Approved by:	______________________

Version	Date	Sections Affected
1.0	2012-02-14	First full version
1.1	2012-02-28	Review comments from AU and Hildesheim taken into consideration. Updates throughout the entire document.
1.2	2012-12-11	Incorporated the comments from 1st year review. In particular the document is updated to focus solely on the transportation of passengers, the description of the use cases has been extended considerably, and a conclusion. Minor layout changes + typos fixed Updated to new template
1.3	2014-01-31	Second year review comments. • Section 1 on work package overview and summary table. • Listed objectivities and how met. • Listed contribution to state-of-the-art. • Internal review comments from FlexDanmark and UHI

Executive Summary

This document describes the requirements to a software prototype that can convert vehicle-related data to metrics that can capture the environmental impact. The vehicle related data is mostly GNSS data and CANBus data. The document uses a number of use cases to establish the requirements to the system. A detailed study lists the requirements to the data foundation, divided into what is needed and what could be useful. A study of related work divided into EU-funded projects and academic papers compares the system designed to what is currently possible. In conclusion, it is found that all requirements are covered by use cases. In addition, all use cases are used.

Table of Contents

List of Figures

List of Tables

1. Introduction

This section identifies the high-level objectives and targets of the project in Section 1.1, followed by the objectives and contributions of work package WP3 in Section 1.2. Section 1.3 shows the objectives and contributions of this deliverable.

1.1 Project Overview

Reduction of CO2 emissions is a great challenge for the transport sector nowadays. Despite recent progress in vehicle manufacturing and fuel technology, still a significant fraction of CO2 emissions in EU cities is resulting from vehicular transportation. Therefore, additional innovative technologies are needed to address the challenge of reducing emissions. The REDUCTION project focuses on advanced ICT solutions for managing multi-modal fleets and reducing their environmental footprint. REDUCTION collects historic and real-time data about driving behavior, routing information, and emissions measurements, that are processed by advanced predictive analytics to enable fleets enhancing their current services as follows:

1) Optimizing driving behavior: supporting effective decision making for the enhancement of drivers' education and the formation of effective policies about optimal traffic operations (speeding, braking, etc.), based on the analytical results over the data that associate driving-behavior patterns with CO2 emissions;

2) Eco-routing: suggesting environmental-friendly routes and allowing multi-modal lets to reduce their overall mileage automatically; and

3) Support for multi-modality: offering a transparent way to support multiple transportation modes and enabling co-modality.

REDUCTION follows an interdisciplinary approach and brings together expertise from several communities. Its innovative, decentralized architecture allows scalability to large fleets by combining both V2V and V2I approaches. Its planned commercial exploitation, based on its proposed cutting edge technology, aims at providing a major breakthrough in the fast growing market of services for "green" fleets in EU and worldwide, and present substantial impact to the challenging environmental goals of EU.

1.2 Work Package Objectives and Tasks

The objective of WP3 is to design and develop a software prototype that can convert vehicle-related data, primarily GPS data, to metrics that capture environmental impact. The prototype must handle very large volumes of data from different types of vehicles and must efficiently compute the multi-modal eco-routes in both real-time and offline modes. In addition, the prototype must be able to report on the temporal evolution of eco-routes, e.g., due to a variety of changes in the transportation infrastructure and its use. The work package will (a) define the interfaces for how vehicles communicate with the server side and with each other; (b) invent and prototype techniques for computing eco-routes; (c) invent and prototype techniques for the validation of eco-routes; and (d) design and prototype high-performance data structures and algorithms for the handling of very large volumes of streaming data from the vehicles.

1.3 Objective of this Deliverable

A first and fundamental step is to specify the requirements of eco-routes. Particular focus must be put on multi-modality transportation requirements. The next step is an analysis of the data that can be transferred from the vehicles to the envisioned software architecture (and vice versa). This includes an analysis of the fundamental and optional data elements that must be transferred in order to fulfill the field trials carried out in WP5. The final step is to specify the software architecture, including non-functional requirements such as performance and costs.

1.3.1 Content

This document contains the requirements specification to the software prototype that computes and evaluates eco-routes. These prototypes are implemented and evaluated in the field trials described in WP5. In particular Task 5.2 Phase 1 Field Trial at BeKTra/FlexDanmark and Task 5.5 Phase 2 Field Trial at BeKTra/FlexDanmark.

First a system definition is provided that includes a description of the system boundaries. There is a listing of the known limitations of the prototype, e.g., a focus on passenger transport and using mainly GNSS data. The business goals of the prototype are listed. Here concrete goals for the use cases are listed.

Next, the actors of the system are found followed by the use cases. This is followed by listing the main risk and a risk assessment.

The minimal requirements to the spatio-temporal data used by the prototype are listed and existing formats are discussed. Next, the data flow from incoming GNSS data to spatio-temporal data analysis.

The overall packages and the classes are then described in details. This forms the foundation of actually implementing the prototypes.

Related work is described next. This is split into academic work (papers and reports) and listing of the EU funded projects that have an overlap with the REDUCTION project. The requirements to the system are then listed and compared to the use cases.

Finally, there is a conclusion and a list of references.

1.3.2 Summary Table

Table 1 provides an overview of the objectives of this deliverable and how these are meet.

Objective	How Meet
Requirement to eco-route	• Make overall system definition • Define business goals • Describe use cases • Study related work ○ Including other EU projects • Risk analysis

Data foundation	• Study data available.
	  ○ What is required?
	  ○ What is optional?
	• List existing data formats

Software architecture	• Describe the overall data flow • Define UML class diagrams

Table 1 Objectives of Deliverable and how meet

2. Glossary

This section lists the central terms and abbreviations used in the document

Term	Description
GHG	Greenhouse gas, such as CO_2 and NO_x
GNSS	Global Navigation Satellite System, e.g., GPS, GLONAS, Compass, or Galileo.
Flex-Traffic	On-demand collective transportation using smaller vehicles. Routes are according to passenger requests and there is no fixed timetable.
Map matching	The process of mapping latitude and longitude coordinates received from a GNSS device to locations in a digital road network. Necessary due to the inherent imprecision of GNSS equipment.
GNSS measurement	The recording of vehicle ID, longitude, latitude, speed, and other relevant data for a single vehicle using, e.g., a GPS device.
System	The complete software described in this document, i.e., the software prototype developed in the work package.
Trip	The path taken by a single vehicle in a road network. The trip consists of a start point, a destination, and zero or more planned stops along the path.
Vehicle type	The type of vehicle such as car, mini-bus, bus, truck, or train

Table 2 Terms used in Document

3. Definitions

The definitions used in the document

- **Low-frequency GNSS data**. GNSS measurements received from vehicles with substantial time gaps between two consecutive GNSS measurements from the same vehicle. The gap is so large that the exact route driven by the vehicle cannot be determined. As an example, if there are 10 minutes between two GNSS measurements from a car driving in a city center, it is impossible to determine which route the vehicle has driven between the two GNSS measurements because there are many alternative routes to pick among.
- **High-frequency GNSS data**. GNSS measurements received from a vehicle with so small time gaps that the accurate route driven by the vehicle can be determined accurately by the map-matching component of the system. Currently high-frequency means receiving a GNSS measurement from a vehicle every 8-9 seconds. Note that time gap between two consecutive GNSS measurements where a route can be established depends on the vehicle

type, where the vehicle has driven (motorway versus city center) and the quality of the map-matching algorithm.

4. System Definition

The system efficiently computes and evaluates eco-routes for vehicles. Focus is *solely* on the transportation of people (passengers), and all vehicles are assumed to be operated by a dedicated driver such as a taxi or bus driver.

The system stores and processes huge volumes of traffic-related data in a scalable and efficient manner. The data is received from a diverse set of vehicles. In addition, data from various operative systems are used to manage both single vehicles and fleets.

The system outputs evaluations of the environmental impact of a single trip and of a set of trips. This output is used as feedback to the operative systems such that the environmental impact of trips can be used as a new optimization and quality parameter in the scheduling of trips.

4.1 System Boundaries

The system specified in this document interacts with a number of other systems.

- The trip scheduling system. This system is responsible for the best utilization of a fleet of vehicles that is controlled by a single traffic coordinator. Data from multiple trip scheduling systems may be received.
- The timetable system. This system can be queried for travel-time information between a start point and a destination. Alternative timetable system can be used, e.g., one for bus transport and another for train transport.

4.2 Limitations

The known limitations of the system include the following.

- The system only supports road transport, i.e., no support for air or sea transport. The reason is that air and sea transport is significantly different from road transport.
- The system only considers transportation of passengers. As an example, the transportation of freight is not considered. The main reasons for focusing on the transportation of passengers is use cases in WP5 will solely look at passenger transport and that the requirements are very different from transportation of freight, e.g., passengers cannot be stored in a warehouse, requires as seat, and cannot be made to change vehicle a large number of times.
- The system only supports road vehicles with combustion engines, i.e., electrical or hybrid cars and trucks are not considered (electrical busses, trams and trains are supported). The main reason is that the number electrical or hybrid cars is still very low.
- The system only supports the GPS because this is the only GNSS in widespread usage. Extending the system to support the Galileo or the GLOSNASS GNSS, should be fairly

straight forward. However, no GNSS measurements from these Galileo or GLOSNASS GNSSs have been available when the system has been designed.

- The system will not support traffic information from sources such as number plate recognition systems [1] [2], Bluetooth detection systems [3] [4], or induction loops [5] [6]. These sources do not allowed the accuracy of determining the actual trips as GNSS measurements do. In addition, the integration of these additional sources will require a major effort, which will not add additional functionality to the system.
- The focus in the prototype is to support car and taxi data. However, there cannot be made design of implementation decisions that prevents other vehicles types, e.g., trucks and busses from supplying data to the platform. The reason for this limitation is that at the time of design only GNSS measurements from cars and taxis are available.
- The system has no information about the drivers of the vehicles. The reason for this limitation is that such functionality can be considered monitoring and tracking individual persons.

4.3 Business Goals

From a business perspective, the overall goals are listed in the following.

- Reduction of the GHG emissions from vehicles used in flex-traffic [7] [8].
- Establishing environmental profiles of vehicle types.
- Estimation of GHG emissions based on GNSS measurements.
- Estimation of the GHG emissions of a single vehicle.
- Enabling the use of GHG emissions goals in public procurement (including bonuses for low GHG emission).
- Enabling of multi-modal planning of trips.

In the following, each business goal is described in further detail.

4.3.1 Reduction of the GHG Emission from Vehicles Used in Flex-Traffic [7] [8]

The public transport companies have a clear focus on improving the environment, such as air quality in the inner cities and GHG emissions in general. In addition, there is a focus on the reduction of the fuel consumption. Overall, the trips using flex-traffic are longer than in other types of public transportation; therefore, there is an increasing interest in reducing GHG emissions from such trips.

Goal: Enabling a 3 to 5% reduction in the fuel consumption.

4.3.2 Environmental Profiles of Vehicle Types

In planning on-demand public transportation, such a flex-traffic, a number of different vehicle types are used, e.g., sedans, station wagons, MPV, mini buses, and busses with special equipment for disabled persons. Currently, most planning is done according to a per-hour rate. This has the consequence that the planning does not take into consideration the different environmental profiles of a bus versus a sedan. If detailed and accurate environmental profiles for the different vehicle types are available, these profiles can be used in the planning. This can then lead to a reduction in GHG emissions simply by being aware of the different profiles for the different vehicle types.

Goal: An environmental profile for the main vehicle type.

4.3.3 Estimation of GHG Emissions Based on GNSS Measurements

Today, a large number of vehicles can report their positions using GNSS devices. A much smaller number of vehicles are equipped with devices that can report their actual fuel consumption. Therefore, to have an immediate impact, it is vital that existing data sources, such as GNSS measurements, can be used in the *estimation* of GHG emissions.

Goal: 90% of vehicles used by fleet operators can have their GHG emission estimated from GNSS measurements from a subset of vehicles in the fleet actually reporting their position.

4.3.4 Evaluation of the GHG Emission of a Single Vehicle

Today, the planning of a trip is based on the fastest route. It is a goal to be able to take the environmental impact of driving a specific route into consideration when planning trips. To ensure that the routes being used are in fact eco-friendly, it is important to be able to monitor each vehicle individually, such that the performance of individual vehicles can be compared and contrasted.

Goal: For 2-4 vehicles to make a detailed evaluation of the GHG emission.

4.3.5 Use of GHG Emissions Goals in Public Procurement

To facilitate a reduction in GHG emissions, it is essential to be able to operationalize the GHG emissions figures such that these figures can be used in public procurement. This can be done by requiring the use of vehicle types with known, low GHG emissions profiles or by being able to award subcontracts a bonus for GHG emission-friendly driving. Ideally, it must be possible to reward both the companies and the individual drivers where emissions-friendly driving can be demonstrated and recorded on a trip-to-trip basis.

Goal: To establish a model for the selection of emission-friendly vehicles types. The model must be operational in public procurement within the laws of the EU.

4.3.6 Multi-Modal Planning of Trips

A large reduction in GHG emissions can be achieved by using schedule-based public transportation instead of individual transport. As an example, a passenger may take the bus instead of a taxi. In particular, it is of interest to enable longer trips to use public transportation for the main part of the trip and only use taxis near the start and end points of the trip. Planning such multi-modal trips is challenging when dealing with passengers that are older or disabled. This requires, for example, detailed information about the walking distance at transit points.

Goal: To estimate the impact on GHG emissions for varying degrees of use of scheduled public transportation.

4.4 Additional Comments

- The GNSS GLONASS is now operational now and working as well as GPS. GLONASS may become increasing popular due to Russian tax on GNSS devices that do not do not support GLONASS but only GPS. [9]

5. Use Cases

This chapter first defines the actors and then lists the use cases.

5.1 Actors

The actors in the system are listed in Table 3.

Actor Name	Description
Driver	A person operating a vehicle. Examples are bus drivers, taxi drivers, and train operators.
Fleet owner	A person or government owning a number of vehicles. Examples are owners of taxis or owner of busses, and the (private or public) owner of trains.
Passenger	A person being transported. Examples are passengers on a bus or the person in a taxi.
Traffic planner	A person or company responsible for planning the trips. Examples are bus or taxi company employees. The traffic planner has domain knowledge about how traffic is planned and how trips are scheduled and executed.
Traffic evaluators	A person or the government that has the political power to determine if environmental impact of a trip should be used as a parameter in the scheduling of trips.
System operator	A person that is responsible for loading input data and assist in producing the output data. This actor is also responsible for the maintenance of the system.

Table 3 Actors in the System

5.1.1 Comments on Actors

- The system operator has been included because the system requires a person with sufficient IT skills to understand the possible complex input and output of the system.

5.2 Use Cases

In the following the use cases for the entire system are listed. The use cases are clustered into a number of packages. Each package is a subsystem, which provides a coherent set of functionality.

The use cases for the **data foundation package** (or subsystem) are shown in the figure next.

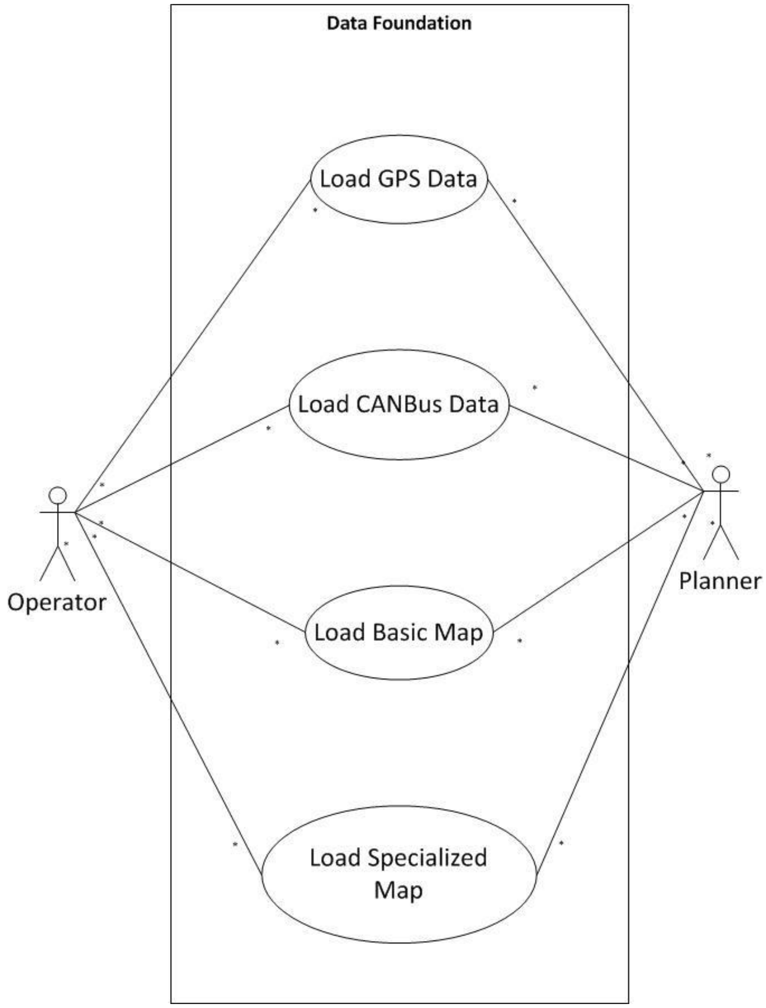

Figure 1 Data Foundation Use Cases

The purpose of the data foundation package is to establish the foundation on which all the computations of eco-route are based. The system operator is responsible for the loading of the data, and the traffic planner is responsible for the validation of the data loaded. The division of responsibility between the two actors is that the system operator must know the details of how the system works and be able to manage each component of the system, while the traffic planner can identify problems in the data sets loaded because this actor has domain knowledge about what is expected from the data loaded and what requirements is expected from the data to be valid for use with the system. The planner is the supervising actor that has power to accept or reject data, and is responsible for acceptable quality of data.

Comments on the Package.

- The specialized map can be a map annotated with multi-modality transport options (different parameters for, .e.g., bus, taxi, train, and ferry), eco-route options, topographic parameters, (e.g., altitude of map), or other information relevant for estimating GHG emission that is not available in typically open-source or commercial maps. This information can be collected or computed from additional third-party data sources.

- The operator actor could be partially replaced by using time as an actor, i.e., data or maps are automatically loaded as part of a setup where new GNSS or CANBus data is received with a regular interval. If the load use cases can be automated with a reasonable effort this will be considered, as this will help ensuring more regular flow of data and easy the job for the planner, as he can count on data being automatically refreshed at given periods. The operator will though still be needed in case of unexpected events, such as invalid data or system errors.
- The basic map and the specialized map are used in two separate use cases because they are based on different data sources and their update frequencies are different. In addition, the update of the specialized map may require additional checking by the planner compared to an update of the basic map. Such verification might be ensuring hubs for multi-modality works correctly or ensure that topographic information is correct for new roads.

The use cases for the **tour foundation package** are show in Figure 2.

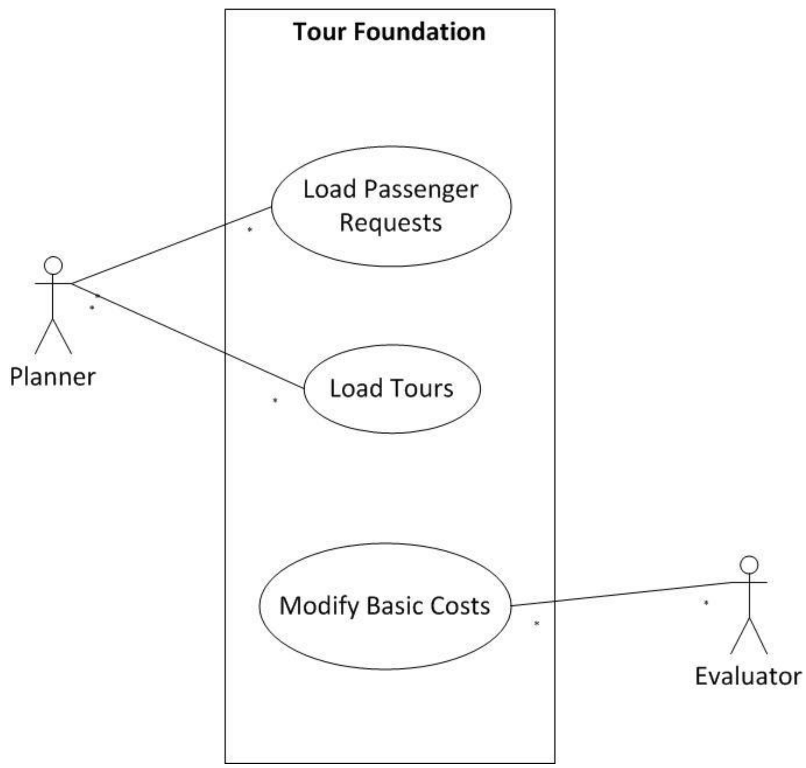

Figure 2 Tour Foundation Use Cases

The purpose of the package is to load the request for transportation and to load how these requests are grouped into trips. In addition, it is the responsibility of the package to facilitate the adjustment of the parameters used in the evaluation of the eco-friendliness of both a single trip and a set of trips. These adjustments are necessary to keep improving the quality of the map.

The use cases for the package **evaluate and annotate** are shown in Figure 3.

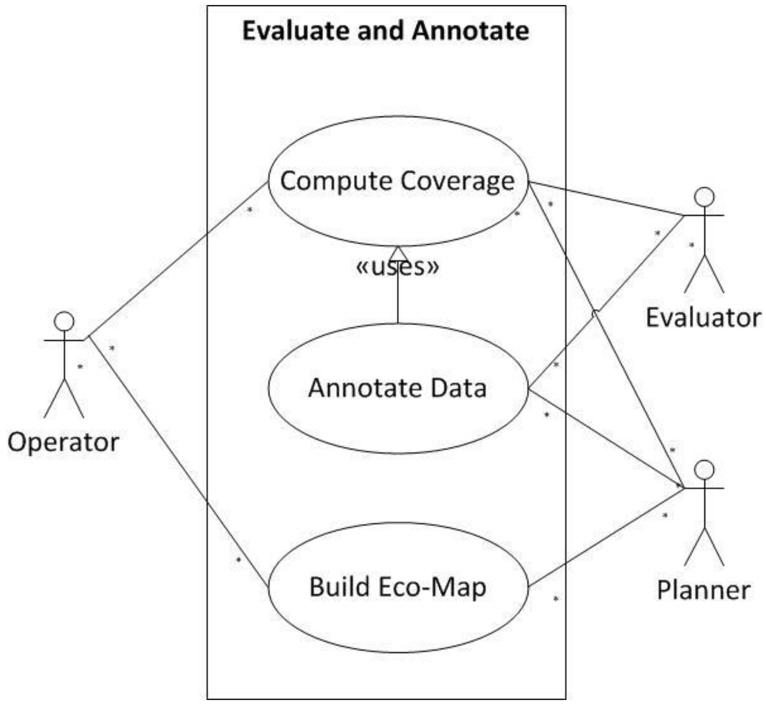

Figure 3 Evaluate and Annotate Use Cases

The purpose of the package is to allow for the computation associated with the map foundation used in the evaluation of trips. The package also offers facilities for making low-level annotations to the map, e.g., where an insufficient number of GNSS measurements is found. When low-level annotations are being performed, it is necessary that these are performed by the operator, since these are very fundamental changes.

The package for **querying** is shown in Figure 4.

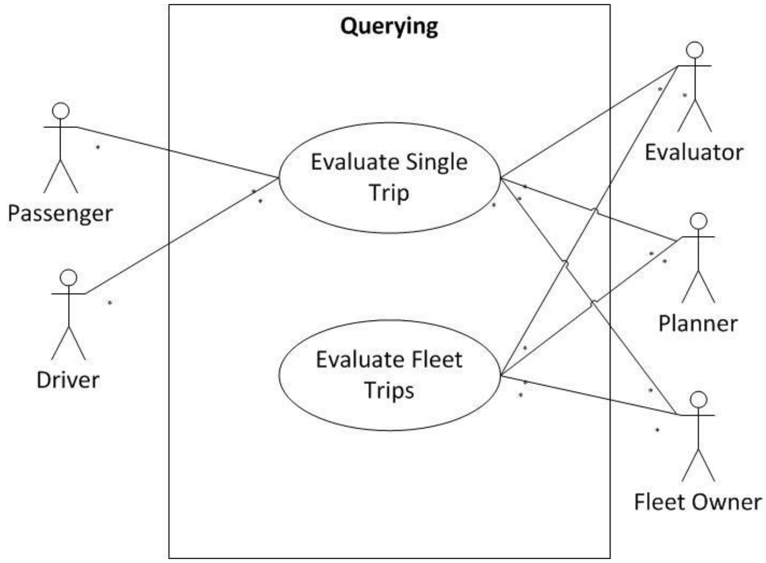

Figure 4 Querying Use Cases

The purpose of the package is to allow the different actors to evaluate both a single trip and the set of trips executed by a fleet. This is important to improve the quality of the data in the system. Using these evaluations, it becomes easier to point out, where improvements are required and where the system is performing well.

In the following, each of the use cases will be described in details.

5.2.1 Load GNSS Data

Preconditions

- A new set of GNSS measurements has been received from data source. This can be a daily load of data or more often.

Flow of Events

1. The use case begins when new GNSS measurements have been received.
    a. The operator enters the file name(s) with the GNSS measurements into the load screen.
    b. The system detects if the data has low or high frequency.
    c. The GNSS measurements are cleansed for known errors.
    d. The GNSS measurements are map-matched.
    e. Relationships to existing measurements are examined.
    f. The proposed updates to the map for computing eco-route are listed.
2. The operator notifies the planner that there are updates to the eco-route map.
3. The planner examines the proposed updates.
4. The planner confirms whether the updated information for the map is valid or seems faulty.
5. The system updates the eco-route map, if the map is accepted by the planner.

Alternative Paths

- If the planner cannot confirm that the update information is valid, the operator is notified.
- The planner and the operator will then together examine the problems and determine actions from the errors. This may lead to new cleansing rules being added to the system by the operator or the new GNSS measurements totally be rejected as not suitable for use. Causes for rejecting GNSS measurements can be errors in data due to faulty GNSS equipment (reporting wrong timestamp, position, speed, etc.) or unexpected conditions (e.g., snowstorm), that might lead to data being way different than "normal" conditions.

Postconditions

- The new set of GNSS measurements have been added to the system.
- The new GNSS measurements have been validated by the planner or reason for rejection recorded.
- The eco-route map is updated, if GNSS measurements are validated correctly.

Comments

- The load of additional GNSS measurements is assumed to happen regularly, e.g., once a day. Could happen every night, when activity is low and data will be available for planner next morning.
- Real-time update with GNSS measurements may be considered (e.g., to account for real-

time traffic conditions), but this will require additional tasks for handling load and validation of data.

5.2.2 Load CANBus Data

Preconditions

- A new set of CANBus measurements has been received.
- A set of GNSS measurements to match to the CANBus measurement comes along with CANBus data or is already available. This is necessary, for being able to match the CANBus data to a map.

Flow of Events

1. The use case begins when new CANBus measurements have been received.
    a. The operator enters the file name(s) with the CANBus measurements into the load screen.
    b. The system checks that GNSS measurements are available for each CANBus measurement.
    c. The GNSS measurements are sent to the Load GNSS Data use case for map matching and cleansing.
    d. The CANBus measurements are cleansed for known errors.
    e. The proposed updates to the map for computing eco-route are listed.
2. The operator notifies the planner that there are updates to the eco-route map.
3. The planner examines the proposed updates.
4. The planner confirms whether the updated information for the map is valid or seems faulty.
5. The system updates the eco-route map, if the map is accepted by the planner.

Alternative Paths

- If a CANBus measurement cannot be combined with a GNSS measurement, the CANBus measurement is removed from the input.
- If the planner cannot confirm that the update information is valid, the operator is notified. The planner and the operator will then together examine the problems and determine actions from the errors. This may lead to new cleansing rules being added to the system by the operator or the new CANBus measurements totally be rejected as not suitable for use. Causes for rejecting CANBus measurements can be errors in data due to faulty equipment (reporting wrong timestamp, position, speed, fuel data, etc.) or unexpected conditions (e.g., snowstorm), that might lead to data being very different than “normal” conditions.

Postconditions

- The pure GNSS measurements eco-route map is updated.
- The combined CANBus and GNSS measurements eco-route map is updated.

Comments

- The load of additional CANBus measurements is assumed to happen regularly, e.g., once a day. Could happen every night, when activity is low and data will be available for planner next morning.

5.2.3 Update Basic Map

Preconditions

- An updated basic map is received. This occurs when an up-to-date map is available and the planner or operator decides that this new map has improved features that are usable for the system. Thus the new map will have to be loaded and used.

Flow of Events

1. The use case begins when the updated map has been received.
    a. The operator enters the file name(s) with the update map into the map-update screen.
    b. The system checks that the update map is suitable as a map foundation.
    c. The differences between the existing map and the updates map are found.
2. The system operator notifies the planner that an updated map has been received. The differences between the existing and the updated map are sent to the planner.
3. The planner examines the map updates.
4. The planner confirms whether the updates the map is valid or seems faulty.
5. The planner confirms that the updates to the map are correct.
6. The system uses the Load GNSS Data use case to update all existing GNSS measurement to the update map, if they map is accepted by the planner.
7. The system uses the Load CANBus Data use case to update all existing CANBus data, if they map is accepted by the planner.

Alternative Paths

- If the planner rejects the updates to the map, neither the GNSS measurements nor the CANBus data is updated. This can happen, if the planner determines something is wrong with the GNSS or CANBus data, or if the new updated map seems to have errors.

Postconditions

- All GNSS measurements and CANBus data are modified to use the updated map.

Comments

- An updated map may contain new roads constructed, roads that are changed, or roads that are deleted. In addition, the map can have updates related to the quality, e.g., a route is more accurately modeled. The update to the map may lead to the situation that some of the existing GNSS measurements or CANBus data no longer can be used, e.g., if an existing road is removed.

- When the basic map changes, all existing data must be revisited because map vendors do not guarantee that updates to maps are backwards compatible. The means the updating the basic map is a very time consuming use case to execute.
- If both GNSS measurements and CANBus data are available for a part of the map, combining these data must be supported.
- A map source such as the Open-Street Map is updated continuously. Such continuous updates are not supported. Instead, it is assumed that an updated map is received every few months.
- It is interesting to record the transaction time of updates on the map, so that it is possible to use the map as it was at any time in past. As an example, if the time of updates of the weights is recorded, it is possible to get the temporal evolution of eco-routes. This would though require a lot of different versions of maps (in terms of evolution over time) to be loaded and populated with data (with a regular interval).

5.2.4 Update Specialized Map

Preconditions

- An update specialized map is received. This will probably happen, when a basic map is updated and hence the specialized map will also need to be updated, to make them usable together.

Flow of Events

1. The use case begins when the updated map has been received.
    a. The operator enters the file name(s) with the updated map into the map-update screen.
    b. The system checks that the updated map is suitable as a map foundation.
    c. The differences between the existing map and the updated map are found.
2. The system operator notifies the planner that an updated map has been received. The differences between the existing and updated map are sent to the planner.
3. The planner examines the map updates.
4. The planner confirms that the updates to map are correct.
5. The updated specialized map is loaded, if the updates to the map are validated.

Postconditions

- No changes to the GNSS measurement or CANBus data are made.

Comments

- Updating the specialized map is much faster than updating the basic map.

5.2.5 Load Passenger Requests

Preconditions

- New passenger request are available to the system, and needs to be loaded. This might happen often and therefore must be a simple and fast task to perform.

Flow of Events

1. The use case begins when additional passenger requests are available.
    a. The planner enters the file name(s) with the additional requests in the add requests screen.
    b. The system checks that all the necessary passenger information is available, e.g.,
        i. address information for the start point and the destination and
        ii. time information (when to pick up a passenger).
    c. The system checks that all address information is available on the map(s) stored.
    d. The system loads the passenger requests.
2. The system returns a status message to the planner on how many requests have been loaded and how many are rejected.

Postconditions

- No changes to the GNSS measurement, CANBus data, or maps are made.

Comments

- Loading passenger request data is a fairly simple task, and is handled by the planner only. The operator is not part of this task, as no low-level tasks are being performed.

5.2.6 Load Tours

Preconditions

- Passenger requests for the tours have been loaded into the system. This can be performed, when new passenger request are loaded into the system on beforehand.

Flow of Events

1. The use case begins when additional tours are available.
2. The planner enters the file name(s) with the additional tours in the add requests screen.
    a. The system checks that all the necessary passenger requests are available.
    b. The system combines the tours with the passenger requests, if necessary.
3. The system returns a status message to the planner on how many tours have been loaded and how many are rejected.

Postconditions

- No changes to the GNSS measurement, CANBus data, or maps are made.

Comments

- Loading tour data is a fairly simple task, and is handled by the planner only. The operator is not part of this task.

5.2.7 Modify Basic Costs

Preconditions

- Some basic costs are not up-to-date and needs to be altered. This happens, when changes occurs to the surrounding world and prices or other parameters changes in the real world.

Flow of Events

1. The use case begins when the traffic evaluator wants to change one or more of the basic costs of the system, e.g., fuel cost, price of man-month, hourly rate for vehicle type, estimated fuel consumption for areas not covered with actual data or estimated GHG emissions per kilometers for busses.
2. On the basic-costs screen, the evaluator identifies the basic cost to modify and adjust these parameters.
    a. The system checks that the changes are legal.
    b. The system asks to evaluator to provide a rational why each change is made.
    c. The system stores the updated basic costs and the rationale for these updates, e.g., a small text message saying update due to increase fuel prices.
3. The planner exits the system.

Postconditions

- All basic cost parameters must have a value.
- The updated values of the cost parameters are used in any further computations.

Comments

- The planner is responsible for the values being correct. If faulty values are entered, they should, if possible be caught by the system and a warning should be displayed. But some wrong values are hard to catch for the system, and the responsibility for the entered values is placed on the planner.

5.2.8 Compute Coverage

Preconditions

- An update of the base data, maps, or cost parameter has happened. Thus the coverage needs to be recomputed to reflect the current state of the system.

Flow of Events

1. The use case starts when a planner or an evaluator requests an update of the travel-time, the fuel consumption, or other values computed from the GNSS measurements and CANBus data and stored with the basic map.
2. The operator is notified that there is a request for computing the coverage.
3. The operator examines the requests.
    a. The program for computing the coverage is executed.
    b. The system notifies the operator when the computation has finished.
4. The operator notifies evaluator or planner that the coverage has been update.

Comments

- The planner or the evaluator cannot automatically recompute the coverage because it is a resource- and time-consuming task.
- It is an interesting research challenge to do automatic recomputing of the coverage. To support automatic recomputation will require computer hardware that may not be available to the REDUCTION project. However, this massive-data parallel processing challenge will be researched if the necessary man-power and computer hardware is available.

5.2.9 Annotate Data

Preconditions

- When a new coverage has been computed, data might need to be annotated using this new coverage value, when former values are insufficient.

Flow of Events

1. The use case starts when an evaluator determines that the current coverage is insufficient or considered inaccurate.
2. The evaluator notifies the planner about where on the map the coverage is considered insufficient.
3. The planner finds the detailed map information where the map is considered inaccurate.
    a. For each road segment, or group of road segments, considered insufficient, the planner documents the current setting and the base data associated with the segment.
    b. The planner updates the map with up-to-date settings, and describes why the new

settings are better than the former.

4. The planner notifies the evaluator about which road segments have been updated and which relevant segments have not been updated and gives the reasons why.
5. The evaluator approves or rejects each individual update made by the planner.
6. The Compute Average use case is executed.

Postconditions

- The coverage is updated to reflect the updated annotations of the basic data.

Comment

- The evaluator does not have the necessary background for updating the information related to the basis map.
- The adjustments made are at a much lower level of details than for example the adjustments made in Modify Basic Costs use case.

5.2.10 Build Eco-Map

Preconditions

- GNSS measurements, CANBus data, and annotations are loaded into the system. Then it is possible to build an improved eco-map, while more data is available now.

Flow of Events

1. The use case starts when any of the data used for building the eco-map has changed.
2. The planner is notified that the data foundation has changed.
    a. The planner evaluates whether an update of the eco-map is needed.
    b. The planner notifies the operator that the eco-map must be updated and explains why.
3. The operator starts the computation of the eco-map.
    a. The system notifies the operator when a new eco-map is available.
4. The operator notifies the planner that an update eco-map is available.

Comment

- The planner does not have the necessary background for building an eco-map. Thus the operator needs to perform this action, as low-level details are changed.
- Building an eco-map can be a time-consuming task hence this is not an ad-hoc task that can be performed immediately. This must be taken into account from the evaluator, when the planner decides an updated map is needed.

5.2.11 Evaluate Single Trip

Preconditions

- When a trip has been executed, it is possible to annotate this trip with evaluation information regarding how it was executed.

Flow of Events

1. The uses case starts when a user logs on to the system and picks evaluate single trip.
2. The user is presented with the list of trips that is associated with the user.
    a. The user picks a single trip.
    b. The system retrieves the trip details.
    c. The system displays the time, distance, and eco details computed by the system for the specific trip on a map.
    d. The user can add comments on the trip, e.g., the trip was longer than last time or that the trip was longer than usual due to bad weather conditions.

5.2.12 Evaluate Fleet Trips

Preconditions

- When trips have been executed, it is possible to annotate these trips with evaluation information regarding how they were executed.

Flow of Events

1. The use case starts when a user logs on to the system and picks evaluate fleet trips.
2. The user is presented with a list of dates and fleets
    a. The user picks a specific data and a specific fleet.
    b. The system retrieves the trips details for the specific date and fleet.
    c. The system displays a number of maps.
        i. A map with the time, distance, and eco details for all trips.
        ii. A map with the utilization of the vehicle, e.g., how many passengers in the vehicle.
        iii. A map with the trips for a single vehicle in the fleet with utilization.
    d. The user can add comments on several trips, e.g., the trips were longer than expected or that the trips were longer than usual due to bad weather conditions on the given day.

6. Risks

In the implementation of the system described in this document the following risks can be identified.

- Insufficient set of GNSS Data.
- Insufficient set of CANBus data
- Too large diversity in input data
- Too expensive to use map
- No access to bus-stop data
- Significant decrease in fuel prices

An evaluation of the each individual risk is shown in Table 4 below. The severity and the likelihood are defined as 1 low, 2 medium, and 3 high. The impact is calculated as the product of the severity and the likelihood (severity * likelihood). The impact it the classified into three categories 1-3 is low, 4-6 is medium, and 7-9 is high.

No	Risk	Severity	Likelihood	Impact
1	Insufficient GNSS data	3	1	3
2	Insufficient CANBus data	2	3	6
3	Too large diversity in input data	2	2	4
4	Too expensive to use map	3	1	3
5	No access to bus or train data	3	2	6
6	Significant decrease in fuel prices	3	1	3

Table 4 Overview and Evaluation of Risks

Risk 1: The partners in the project have themselves access to significant size of GNSS measurements from for an extended period of time. In addition, there are a number of organizations, private persons, and existing research projects where GNSS measurements can be accessed or downloaded.

Risk 2: The severity of not having access to CANBus data is smaller than the severity of not having access to GNSS measurements, because eco-routes can be estimated from GNSS measurements. However, CANBus data can have a large impact on the accuracy of the eco-routes and GHG emissions commuted. Therefore it is of a considerable interest for the project to have access to CANBus data.

GNSS measurements have been collected by many transport companies for a considerable period of time. However, collecting and storing CANBus data has begun more recently. Therefore less data source is available and most likely for smaller geographical areas than GNSS measurements. However, it a larger set of CANBus data is available for an area where a large set of GNSS measurements is also available it is a hypothesis that the estimation of GHG emission and reduction can be calculated for all areas where only GNSS measurements are available.

Never the less this is a risk! To minimize the potential negative impact on the project a number of measurements have already been taken. In particular, have a number of potential suppliers of CANBus data has already been contacted and negotiation about access to CANBus data is ongoing. Further, existing project that collects CANBus data has been contacted.

Risk 3: Most GNSS manufactures have their own format for delivering GNSS measurements. In additions, there are apparently no open standard for the exchange of CANBus data (will be discussed in a later section). This may cause that there are many different input formats to consider. To mitigate this risk the following measure will be taken. 1) Existing tools for converting between GNSS measurements format will be used, e.g., GPSBabel. 2) A few large suppliers of CANBus data will be preferred and the system will use isolated subsystems that can handle these few large suppliers.

Risk 4: The use and distribution of digital maps is central to the functionality supplied. In addition, commercial digital maps are very expensive to use in particular when the map must be distributed to a number of partners and users. Therefore to minimize this risk the open-source map Open-Street Map will be used. Preliminary investigations have shown that OSM map is suitable for the purpose envisioned. Therefore this risk is assumed to be low.

Risk 5: To plan and evaluate multi-modal transportation of passengers it is important to have access to where the bus-stops and train-stops are and the schedule of the transport modes. The bus/train stops are available via OSM maps; however, the quality of this information is unknown. In addition, the schedules may not be available in an appropriate format.

To mitigate this risk contacts to bus- and train companies must be done early in the project. It is assumed that gaining access to stop and schedule can be negotiated since the bus- and train companies have a commercial interest in providing access to the data because it may result in more passengers.

Risk 6: The current fuel prices are high and therefore transport companies have a keen interest in lowering the fuel consumption and therefore the GHG emissions. Of this reason transport companies have a strong commercial interest in collaborating with the REDUCTION project, e.g., by providing GNSS measurements of CANBus data. Should the fuel prices drop considerably, the interesting in allowing research access to data could be considerably lower. It is assumed in an assumption that the fuel prices will remain high for the duration of the REDUCTION project and that this risk is low.

7. Data Requirements

This section describes the mandatory and optional data elements that must be available to the system. The data is divided into fundamental and optional data of the following reasons.

- In practice the various data sources have many different.
- It is not know exactly what can be used in the field trials.

7.1 Spatio-Temporal Data

Spatio-temporal data describes the position of a vehicle at a given time. This data can be provided by the different types of GNSSs. In addition to the spatio-temporal data, a GNSS device may record a number of additional data elements.

7.1.1 Mandatory Data Elements

The following data elements are mandatory for each GNSS measurement.

- **Vehicle ID:** A unique identifier for a vehicle.
  - The ID of vehicle is not allow to change if it changes it will be considered another vehicle.
- **Longitude and latitude (location):** The longitude and latitude of the vehicle.
  - At the highest precision available from the GNSS device.
  - Indication if it the raw or map-matched longitude and latitude.
- **Timestamp:** The date and time of when the location data element was saved.
  - A seconds granularity
- **Velocity:** The velocity is needed for computing average velocities.
  - It must be indicate if it is an odometer velocity or a GNSS velocity
  - Unit must be provide, e.g., km/h or m/h
- **Direction:** The compass direction. (Degrees, rounded to nearest integer).

All GNSS measurements from a vehicle should be provided, i.e., there should be no filtering of GNSS measurements by the clients.

7.1.2 Optional Data Elements

The following optional spatio-temporal data elements

- **Altitude**: The altitude of the vehicle
- **Dilution of Precision**: Accuracy reporting from GNSS devices
  - **GDOP:** Geometrical
  - **HDOP:** Horizontal
  - **TDOP:** Time
  - **VDOP**: Vertical
- **Number of Satellites**: The number of satellites in the GNSS used to determine the spatio-temporal data.
- **Timestamps of dataflow**: When was data received by the backend.

- **Map-matched longitude and latitude coordinates**: GNSS device may support map-matching in the device.

7.1.3 Analysis

The spatio-temporal data is necessary to determine where a vehicle has been. The location is required, at least in the plane of longitude and latitude coordinates. The precision of the location should be down to a few meters of uncertainty. The location should be in latitude / longitude or any projection that can be transformed into latitude / longitude, such as UTM co-ordinate including description of UTM zone.

Except for the location, a timestamp is needed to determine when the vehicle was at the location, and the velocity is needed, to know how fast the vehicle was driving. These parameters are required for generating a speed map, which can tell how fast in average vehicles have been driving on different segments. The direction is required for being able to determine in which direction, on the segment, a vehicle was driving.

In addition to these, it would be nice to be able to know how precise the position data actually is, to know how precise the location recording actually was. Some parameters can indicate this, depending on positioning system. The altitude could also add extra value for the location data, while it would help compute fuel consumption. Different timestamps of the dataflow, e.g. a timestamp describing when the data is loaded into a backend system, can help determining if the timestamp from the location system seems plausible or faulty. At least, a high sampling frequency would be very nice, preferable a location update every 1 second. With such a high sampling frequency, it would be possible to very precisely follow a vehicle and it would also be possible to compute accelerations and more precise estimate fuel consumption.

7.1.4 Data formats

Several data formats exists for storing position tracking data, while some are proprietary others are open and documentation is freely available and usage is free too.

Here are listed the most used and interesting open formats for storing position data, along with a description of how well they conform to the requirements of data for this project.

- **GPX:** GPS eXchange Format [10], for describing positioning data, is an open format without any licenses needed. The older GPX 1.0 format supports velocity and direction in addition to a location and a timestamp. Then newer GPX 1.1 format does neither support velocity nor direction, while these should be able to be calculated from the difference between two positions. This is though only optimal if you have position updates every second while if more than one second exists between two positions the imprecision increases when computing velocity and direction.
- **GPXX:** Garmin GPX Extension [11] is an extension to the GPX format defined by Garmin. The GPXX format is an extension to the GPX 1.1 format, which means that it does not store velocity and direction. These have to be computed from positions. This would again require

1 second position logs to ensure optimal precision of velocity and direction.

- **TCX:** Training Center XML [12] is a format developed by Garmin and is similar to GPX, though does this contain additional information for training activities and body measurements. This TCX format does contain several kinds of velocities and direction measures, while the format is not meant for tracking moving objects.
- **AXM:** The Activity XML [13] format by Garmin is very similar to the TCX format, while it though can store personalization data too. Though it is improper of being used for recording vehicle trajectories.
- **KML:** The Keyhole Markup Language [14] is an XML format for expressing geographic annotations. It can contain elements for representing objects in space, while it cannot store velocity and direction. KML is a format made primarily for showing geographic elements rather than transferring GNSS data, while it though is capable of storing tracks.
- **NMEA:** NMEA, short for National Marine Electronics Association, has two standards for storing and transmitting positions, namely NMEA 0183 [15] and NMEA 2000 [16]. Both standards are describing interfaces and structures for how communication should be done through data busses. Some of this information, parsed in ASCII text strings called messages, contains description of locations, along with direction, velocity, and timestamps. Both standards are proprietary and not public available. The older NMEA 0183 standard has been reverse engineered and open-source software and descriptions exists for parsing the data messages. Though NMEA warns of using these tools as they should not be up-to-date with the latest and correct types of messages [15].
- **RINEX:** The Receiver Independent Exchange Format [17] is a data format which is independent of positioning system and sends data to the positioning unit, for the unit to compute its output data, such as location and velocity. This data format is quite raw and not very usable for tracking an objects trajectory.
- **BINEX:** The BINary EXchange format [18] is a binary format for exchanging all possible information from location satellite systems for research purpose. As with RINEX, this format seems too raw for this project.

7.1.5 Summary

An overview of the existing format examined for receiving spatio-temporal data in the form of GNSS data is listed in Figure 5 below. As can be seen from the figure only one of the formats fulfills the requirement of both being suitable for the purpose and an open standard.

Name	Format	Open	Match Purpose
GPX	XML	Yes	Yes*
GPXX	XML	Yes	No
TCX	XML	Yes	No
AXM	XML	Yes	No
KML	XML	Yes	No
NMEA	ASCII	No	Yes
RINEX	ASCII	Yes	No
BINEX	Binary	Yes	No

Figure 5 Overview of Spatio-Temporal Data Formats

The GPX format is marked with *, while the newest standard, 1.1, is not usable the older 1.0 standard is.

For converting between formats, various tools exists such as the open-source tool named GPSBabel [19]. This has the possibility of converting positioning tracks between many popular formats, such as GPX, KML and NMEA. It is possible of selecting versions of GPX schema, though it can only output as much information as read from the input file. That means, if a GPX 1.0 file with velocity and direction is read and a GPX 1.1 file is outputted, the velocity and direction is lost. If then the GPX 1.1 file is read and a GPX 1.0 file is outputted, the velocity and direction cannot be restored, while it was lost in the 1.0 to 1.1 translation.

7.2 Vehicle Information Data

Vehicle information data is a general term for any information of a vehicle, whether it is static information such as color or make, or more technical information such as fuel consumption, pedal positions or weight of vehicle. Social information is also vehicle information data, such as how many passengers are present, and for instance their weights.

Such information can be provided from a description of the vehicle for static information, or through different technical interfaces, such as OBD [20], ECU [21] and CAN [22] data. It might also be provided from external registered data, such as route planning databases or manually entered data.

7.2.1 Mandatory Data Elements

- **Vehicle ID**: The identification of the vehicle.
- **Fuel consumption:** Some kind of measure, preferable a precise measure, such as milliliters, that tells how much fuel a vehicle has consumed. Unit of measure must be known!
- **Location/timestamp data link:** Some kind of reference between vehicle information data and location data that can combine these two kinds of data.
- **Vehicle type:** Car, minibus, truck

7.2.2 Optional Data Elements

- **Vehicle information:**
  - **Make**: The make of the vehicle
  - **Model**: The model name
  - **Manufacturing year**: The year the vehicle was manufactured
- **Engine information:**
  - **Engine type**: The size of the engine, e.g., 1.2 liter and 3 cylinders
  - **Fuel type**: The fuel type, e.g., petrol, diesel, bio-diesel, or gas.
- **Air intake temperature:** Nice to know the temperature of the outside weather. The engine might run differently on extreme temperatures.
- **Vehicle kilometer stamp:** Total number of kilometers driven by car (unit of meters)
- **Brake pedal position:** Is the brake pedal being activated, and how much?
- **Clutch pedal position:** Is the clutch pedal being activated for changing gear?
- **Acceleration pedal:** Is the acceleration pedal begin pressed, and how much?
- **Cruise control position:** Is cruise control activated? This might have an impact on fuel consumption compared to driving without fuel consumption.
- **Odometer velocity:** If the odometer velocity is available this might be more precise than the calculated speed from the location system.
- **Engine speed:** How fast is the engine running, e.g. rpm.
- **Engine load:** How much strain is put on the engine?
- **Weight load:** How heavy load is put into the vehicle?
- **Transmission info:** Automatic or manual. What gear is the transmission in?
- **Gas emission information:** How much CO_2, O_2, and other gasses are present in the exhaust gas?
- **Passenger count:** The number of passengers could be very interesting, to compute the amount of CO_2 emitted pr. Passenger
- **Passenger weight:** To be able to compare how much one passenger affects the amount of CO_2 emitted.

7.2.3 Analysis

The most valuable, hence important, data is the fuel consumption reported by the engine. This parameter makes it possible to estimate how much fuel a vehicle has been consuming while driving a trip. Such information does though require to be linked with a location.

A load of additional information could be useful for better estimating and analyzing driving behavior and fuel consumption / emissions of a vehicle. It might be useful to know the number of passengers, their weigh, engine information, and general vehicle information.

7.2.4 Data formats

Many data formats and standards exist for information data of different kinds of vehicles, such as NMEA 2000 [16] for marine and ARINC 825 [23] for airborne use. Unfortunately the passenger vehicle industry has not agreed on one standard and every manufacturer has its own system for retrieving data from vehicles. The OBD standard exists, while it does not give a precise fuel measure. It can only give a lot of measures which can be used to estimating the fuel measure.

7.2.5 Summary

Retrieving vehicle information is a non-trivial task because no standard exists. Hence different kinds of values may be returned from different data sources and a general algorithm for loading vehicle information data cannot be developed.

7.3 Map

For a map to be usable for performing map-matching and shortest paths on, some features are required and others very useful.

7.3.1 Must have

The following data elements are required for the map data.

- **Segment ID:** An identifier of the segment. This does not have to be unique.
- **Segment category:** Segments are grouped into different categories, which later are used for filling missing data spots with similar grouped segments.
- **Start-point / End-point:** A value description what segments are joined in the ends. Needed for generating a graph to perform shortest path computations on.
- **Segment geometry:** A geometry describing the segments location on a map. From this, the length of the segment can be calculated.
- **Graph:** The map must be able to be loaded as a graph, for performing fast lookups and shortest-paths.

7.3.2 Nice to have

The following data is optional for the map data.

- **Name:** A name of the segment, to ease output representation and understanding.
- **Speed limits in both directions:** Useful for filtering unreliable speeds from location data.
- **Segment direction:** A description of in which directions it is allowed to drive on a segment.

- **Modality tag**: A tag in the map that states which modality can be used, , e.g., a trail way, a bus route, or a normal road segment for vehicles.
- **Transition point**: Points on the map that indicates that a change in modality is possible, e.g., at a train station it possible to transfer from train to bus or train to taxi.

7.3.3 Analysis

The map is required for having something a data foundation to map the GNSS data against. For being able to perform fast map-matching the map has to be loaded into a graph, which means all segments must be joined in the ends only. It is not allowed for segments to be joined other segments between the two endpoints of a segment. Each segment has two endpoints, named Start-point and End-point. These correspond to vertices of a graph and are represented as integers. All Start/Endpoints with equal integer values are joined together.

The map has to have IDs of the segments, in order to identify the segment. A category is needed for being able to group segments with similar other segments. At last, a spatial geometry is required for storing the segment itself. When storing the segment as geometry, measures can easily be computed and queried from the segment and it is also simple to transform the geometry into other projections.

Some features would be nice to have too, such as a name for easier identifying and recognizing a segment. Speed limits can be very useful both when recognizing wrong data but could also be used for generating analysis on segments with low amount of data. At last direction is needed, if it should be possible to tell difference between whether a segment is a one-way or two-way traffic segment. This is especially important if the map has two segments for each direction on wide roads, such as motorways. If two parallel segments are present, without directions, it is very hard to guarantee map-matching won't match the wrong directional segment.

7.3.4 Data Formats

Many data formats exist for storing maps in files. Some have a strictly defined structure while others are more general and can contain all kinds of data. Some are proprietary while others have open specifications. Here is a list of the most popular map formats, all used widely for many purposes.

- **Shapefile:** The Shapefile format [24] is a vector based data format for storing maps and geographic information. The format is being widely used and has existed since the beginning of the 1990s. Due to its age, it is also though a quite limited format, in the sense it is not possible to store topographical information, data files cannot exceed 2 GB in size, poor Unicode support, only 10 character field names, highest integer values is 255, and text values may only be 254 character long. There is no standard interface for naming of fields in a shapefile, hence there cannot be made a standard loader for loading shapefiles. Every shapefile might have different structure.
- **OpenStreetMap:** The OpenStreetMap (OSM) uses two kinds of data formats: osm and pbf. The osm format is xml based while the pbf is a binary format. The binary format is much

smaller in size and faster to use. An OSM map has the disadvantage, when it comes to making computations, that it cannot directly be transformed into a graph. The segments are not only joined in the ends but also at the middle of segments. Tools do though exist, such as osm2po [25], for splitting an OSM map into smaller segments, which then can be loaded as a graph. A graph is necessary for many map-wise computations, e.g., when querying for a shortest path.

- **KML:** The Keyhole Markup Language [14] is an XML format for expressing geographic objects. It can contain elements for representing objects in space. KML has the disadvantage of not being able to tell if a road has one-way or two-way traffic.
- **MapInfo TAB:** MapInfo TAB [26] is a proprietary geospatial vector format for storing geographically objects. The format is very popular and used widely across the industry.

7.3.5 Summary

An overview of some of the most popular existing map formats are shown below in Table 5. All four formats are vector based, while they are stored in different kinds of file formats. Only Shapefile and OpenStreetMap files do though match the purpose for this project, while they are open formats and they

Name	File format	Format	Open	Match Purpose
Shapefile	Binary	Vector	Yes	Yes
OpenStreetMap	Binary/XML	Vector	Yes	Yes*
KML	XML	Vector	Yes	Yes**
MapInfo TAB	Binary	Vector	No	No

Table 5 Overview of Map Formats

The OpenStreetMap format, marked with *, needs to be converted using extra tools for matching the requirement of being loaded as a graph.

The KML format, marked with **, is usable, while it does not contain information of driving directions on segments.

7.4 Summary

The system needs three main types of information spatio-temporal information, vehicle information, and digital maps as shown in Table 6. It is mostly relevant to get the spatio-temporal data in real-time. The vehicle and map data can be received in a batch-job manner because this information changes slowly whereas the spatio-temporal information by nature changes very frequently, e.g., with the congestion level in the cities or on the motorways.

Main Type	Real-Time
Spatio-Temporal	Yes
Vehicle	No
Map	No

Table 6 Overview of Data Foundation and Real-Time Relevancy

8. Data Flow

This section describes the data flows through the system from input to output. This helps understanding how information is used through the system. First an overview is given and then a more detailed description shows what parts of the system actually uses the data.

8.1 Input / Output Data Flow

On overview of the input/output flow is shown in Figure 6. The main purpose is to illustrate the data input need and output from the system.

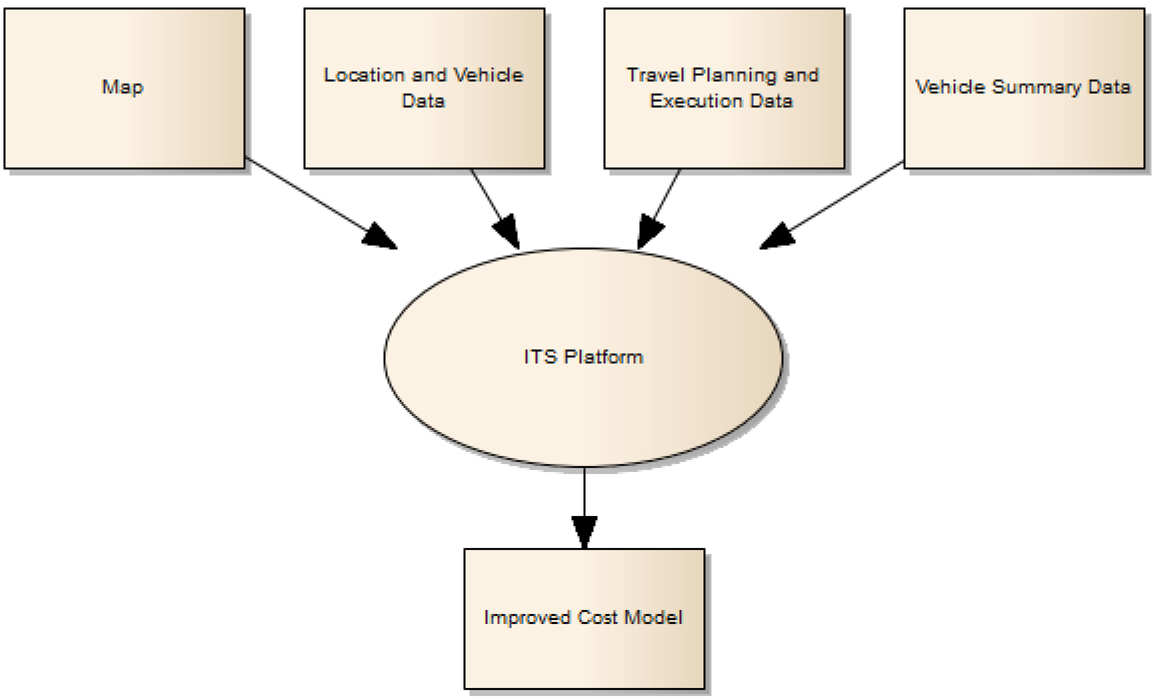

Figure 6 Main Data Flow

The system, called ITS Platform, takes four kinds of inputs from external sources:

- ***Map***: A road map of the region, which is going to be analyzed and used for the model. This should cover the region for the *Location + Vehicle Data*.
- ***Location and Vehicle Data***: Data describing a vehicles position at a given time, along with information regarding the vehicles status and fuel consumption, velocity, direction, etc.
- ***Travel Planning and Execution Data:*** Data describing existing travels organized and executed. This source keep information of every single transport, the customer, the passenger, the vehicle, economy, travel time, waiting time, requirements and so on.
- ***Vehicle Summary Data:*** This source contains information of every vehicles summarized data for an amount of transportation. Such an amount of transportation is called a **vehicle run**. A vehicle run can contain a whole day's number of transportations for, e.g., a taxi, while

an entire day's transportations for a taxi can be separated into several vehicle runs.

The ***ITS Platform*** can be seen as a black box component, for now, that takes these four inputs and returns an ***Improved Cost Model***, which can be used by planning companies.

8.2 Detailed Data Flow

When unpacking the *ITS Platform*, it becomes visible that it contains some processes and data storages for handling the input data and transforming them into a usable output. The flow and usage of input data can be seen from Figure 7 below.

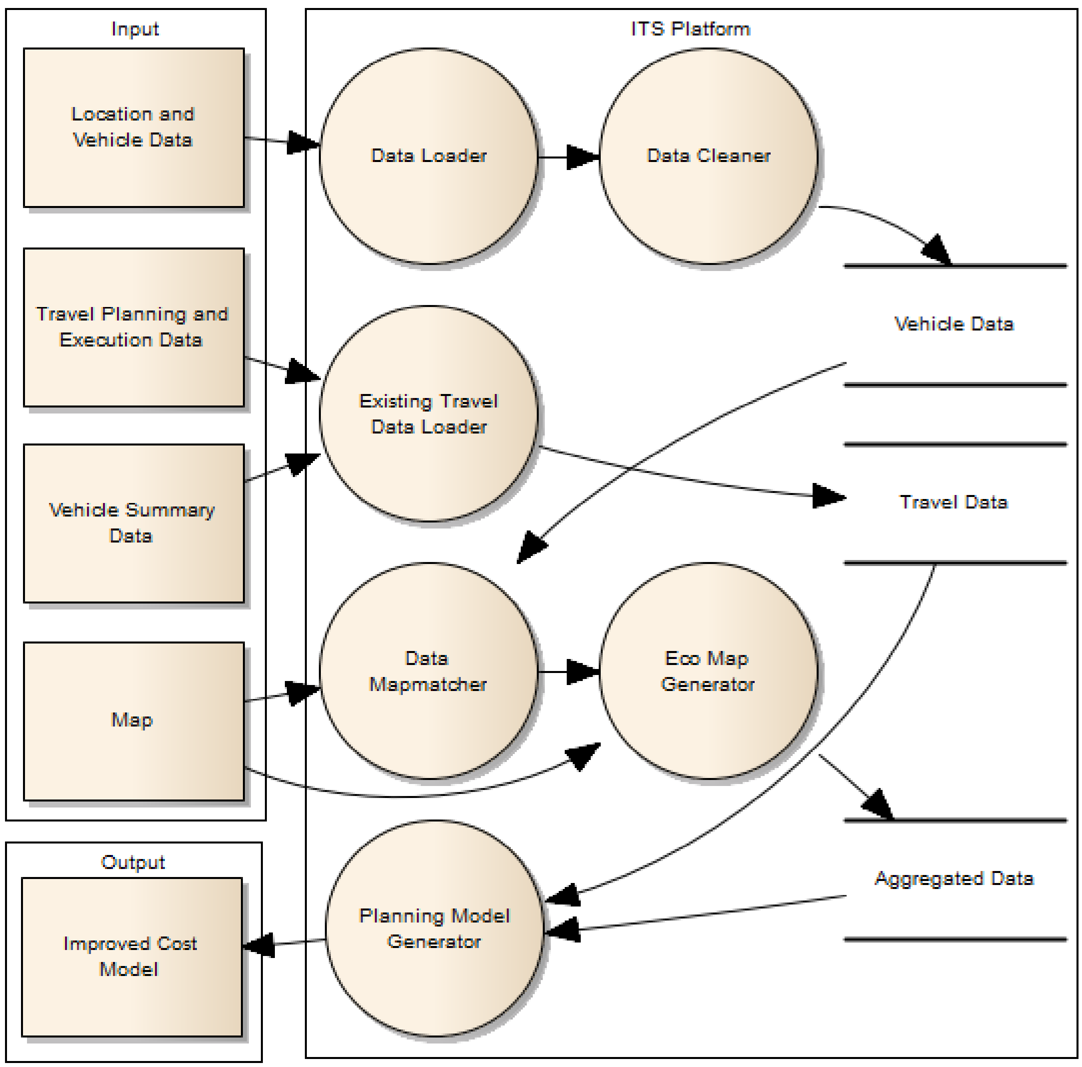

Figure 7 Detailed Data Flow

It can be seen, that data from *Location and Vehicle Data* goes through the *Data Loader* and a *Data Cleaner* processes in its way to the data storage *Vehicle Data*.

The *Travel Planning and Execution* Data and the *Vehicle Summary Data* both goes through the *Existing Travel Data Loader* before it is stored in the *Travel Data* storage.

From the *Vehicle Data* storage the data is loaded into the *Data Mapmatcher* process, which used an External *Map* for mapmatching the location data. The output from this process is processed by the *Eco Map Generator* which output is stored into the Aggregated *Data* storage.

The data from the *Aggregated Data* storage and from the *Travel Data* storage is processed by the *Planning Model Generator*, which outputs an *Improved Cost Model*.

9. Class Descriptions

In this chapter the classes will be described, which are contained by the model, along with how the classes interact with each other.

9.1 Introduction

A complete UML class diagram model of the system is shown below in Figure 8. It provides an overview of how the system is structured.

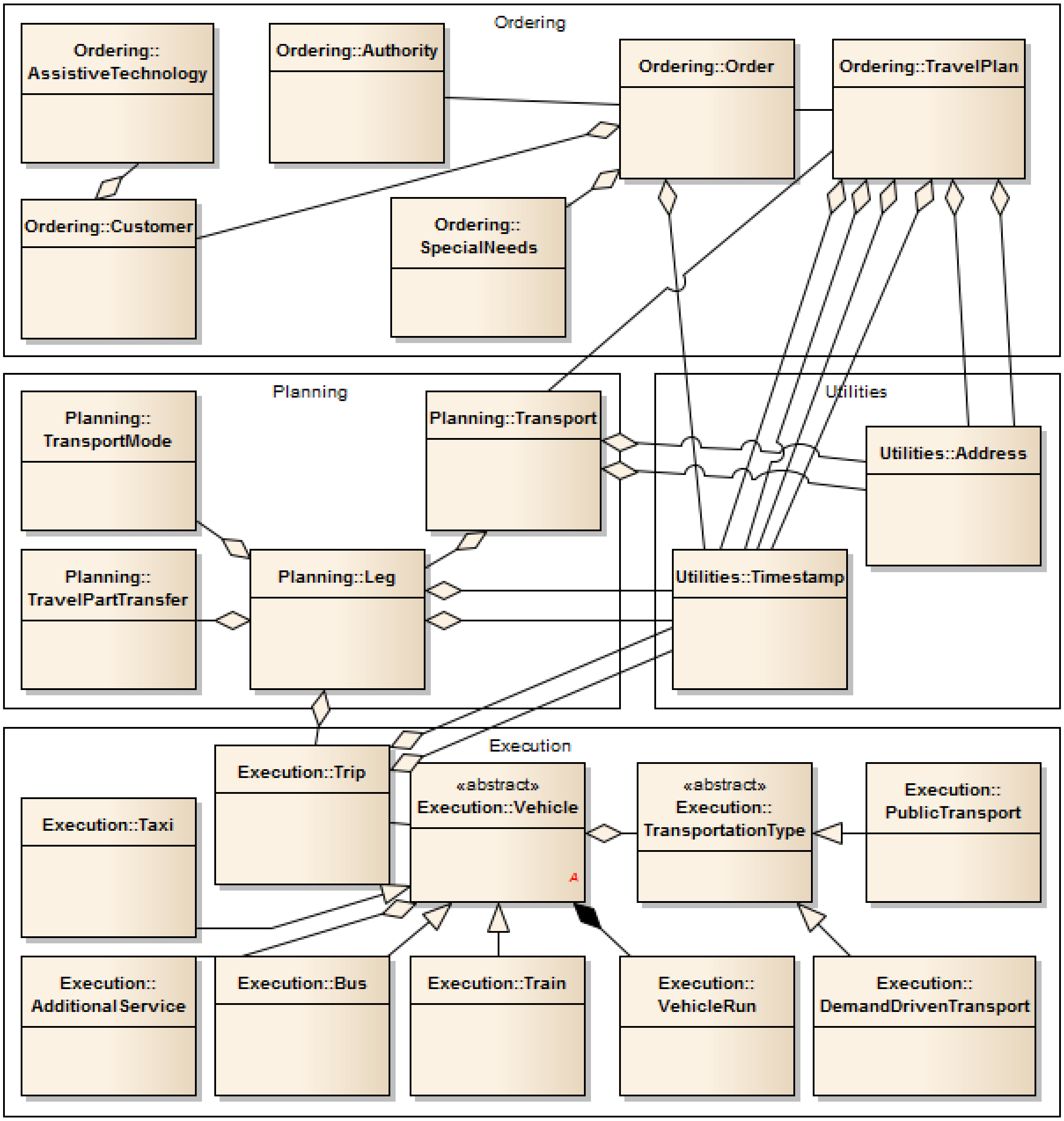

Figure 8 UML Class Diagram for System

It can be seen from the diagram that the system is organized into packages. These packages are logically created from when the classes are used, hence the naming of the packages: *Ordering*, *Planning*, *Execution,* and *Utilities*. More details on these packages later. These packages will be described in greater details later.

In the following sections, the packages are described in in a specific structure. First the package will be introduced in general terms, then it will be described more detailed what the purpose of the content of the package is and how it is used. Next the individual classes will be described, while at last the connections between the classes will be explained.

9.2 Conventions

Some conventions have been defined for keeping consistent notations throughout the document.

Then naming convention of classes is to use capitalized letters as beginning letters and not using spaces, hence a name if a class could look like *ClassForStoringWeather*.

Packages and classes will always be referred to in *italic*.

9.3 Packages

The system is divided into logically packages, defined by the order of the package execution and combinations. The packages will be described in the following sections.

9.3.1 Utilities

The *Utilities* package is a package for classes that are used across several other packages and cannot be restricted to existing in one package only. Thus the *Utilities* package can be seen as a group of shared classes.

Description

The classes within this package are independent of other classes within this package and from other packages, as can be seen from the diagram below in Figure 9.

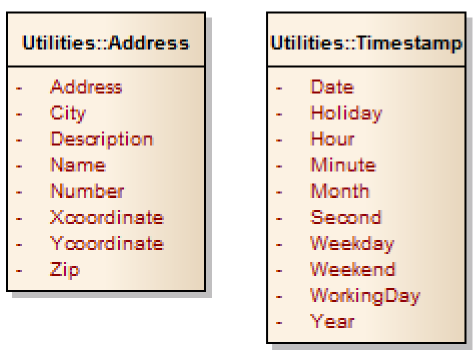

Figure 9 Utility Classes

The advantage of having such shared classes is consistence of class structures when accessing data across different packages, along with the possibility of reusing information, such as addresses.

Classes

Two classes exist within the *Utilities* package, namely *Address* and *Timestamp*, as can be seen from the diagram in Figure 9. These two classes are both referenced from other packages. The *Address* class contains all kinds of addresses that could exist throughout the system, while *Timestamp* stores all timestamps.

Address

The *Address* class contains general address information such as name, road name, number, zip, city along with a description of the location. This description could say "Entrance B of the shopping mall" or "Taxi stop at the mall". Also an X and Y coordinate of the location is present, to precisely identify the location.

Timestamp

The *Timestamp* class describes a specific timestamp. The class describes both date and time, along with special descriptions of every timestamp, such as whether the date is a working day, a weekend, a holiday, etc. The precision of the *Timestamp* is 1 second.

Connections

There are no connections between any classes in the *Utilities* package. All connections to the classes within this package are described from the referencing classes from other packages.

9.3.2 Ordering

The *Ordering* package contains all classes needed for when creating an order for a new transportation request. This covers everything from the order is requested to storing information on what/who is being transported to when the transport is going to be executed and if any requirements are necessary for the transportation.

Description

The structure of the *Ordering* packages can be seen below in Figure 10 where all classes are shown. Please note, that *Timestamp* and *Address* classes are imported from the *Utilities* package.

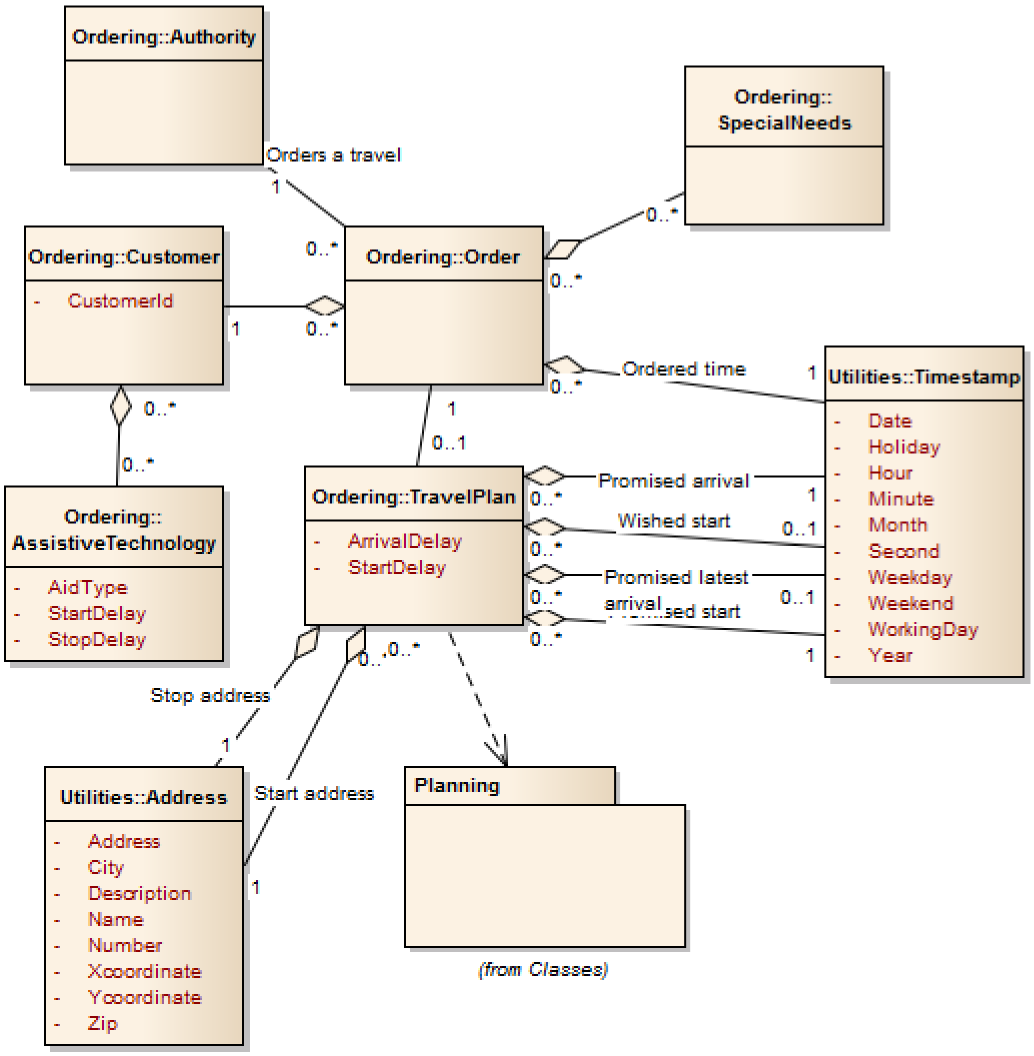

Figure 10 Ordering Classes

When a transport is ordered from an *Authority,* an *Order* is created. This *Order* will concern a *Customer* which is going to be transported. A *Customer* could be a patient, a school child or any other person. Such a *Customer* might carry some *Assistive Technology* such as a wheelchair or walker, which requires special attention. There might also be other *SpecialNeeds* required for transporting the *Customer*, such as the *Customer* might take up several seats or require assistive personal. At last it is saves at what *Timestamp* the order was created.

When the *Order* is created, a *TravelPlan* is being made, that is describing from and to what *Address* the *Customer* is going to be transported. Also it is necessary to know what time requirements are promised for the transportation.

Finally, when an order is created, it is ready for being processed by the *Planning* package.

Classes

The *Ordering* package contains classes defining how an order is put together. Also two classes from the *Utilities* package are utilized from this package.

AssistiveTechnology

AssistiveTechnology can be seen as an additional luggage a customer needs to bring with. This can be a wheel chair, a walker or perhaps medical facilities. Such additional aid might add startup and finish delay to the total transportation time, due to extra services required from the driver.

Authority

This class describes the authority who orders a trip. This can be a municipality or a hospital or other. This ordering *Authority* is important because this is different from the *Customer* and the *Authority* is deciding who is paying for the transportation.

Customer

A *Customer* is an individual person who is going to be transported from one location another. This person can be a patient going to the hospital or a child going to school.

Order

This class is unifying the *Customer* and the *Authority* with other *SpecialNeeds* and a creation *Timestamp*.

SpecialNeeds

SpecialNeeds describes if some requirements are necessary. This might be helping personal that is required for a patient to be transported or it might describe that the passenger has to use two seats or lay down.

TravelPlan

The *TravelPlan* represents a description of from what *Address* to what *Address* the travel is going to take place, along with descriptions of time requirements. These requirements are implemented with a *Timestamp* describing when the travel is wished to start or more strictly when the travel is promised to start. Also a promised arrival at destination is stored or more strictly a promised latest arrival is stored using *Timestamp*.

Connections

An *Authority* is associated with an *Order* in the way that an *Order* must be associated with one *Authority*, while an *Authority* does not necessarily have to be associated with any *Order* but may be associated with many *Order* classes.

An *Order* must have one *Customer,* while a *Customer* can be associated with none or more *Order* classes.

A *Customer* may have some *AssistiveTechnology* while *Assistive Technology* can be associated with many *Customer* classes but does not have to.

An *Order* can have some *Special Needs* though it does not have to and a *SpecialNeed* can be associated with one or many orders, but not necessarily.

An *Order* must have an ordered *Timestamp* while a *Timestamp* does not need to be associated with any *Order* classes.

When an *Order* is created none or more *TravelPlan* classes can be associated with this, while a *TravelPlan* must be associated with one *Order* only.

A *TravelPlan* must have one start *Address* and one stop *Address* class while the *Address* class does not need to be associated with any *TravelPlan* classes.

The *TravelPlan* class must have two *Timestamp* describing "promised start" and "promised arrival", though it can have two additional *Timestamp* describing "wished start" and "promised latest arrival". The *Timestamp* does not need to be associated with *TravelPlan.*

9.3.3 Planning

The *Planning* packages contain classes describing information of how a trip is planned to be executed, from how the trip is ordered. This is purely computational and is being done before the travel is being executed.

Description

A class diagram of the *Planning* packages can be seen below in Figure 11. Please note, that *Timestamp* and *Address* classes are imported from the *Utilities* package.

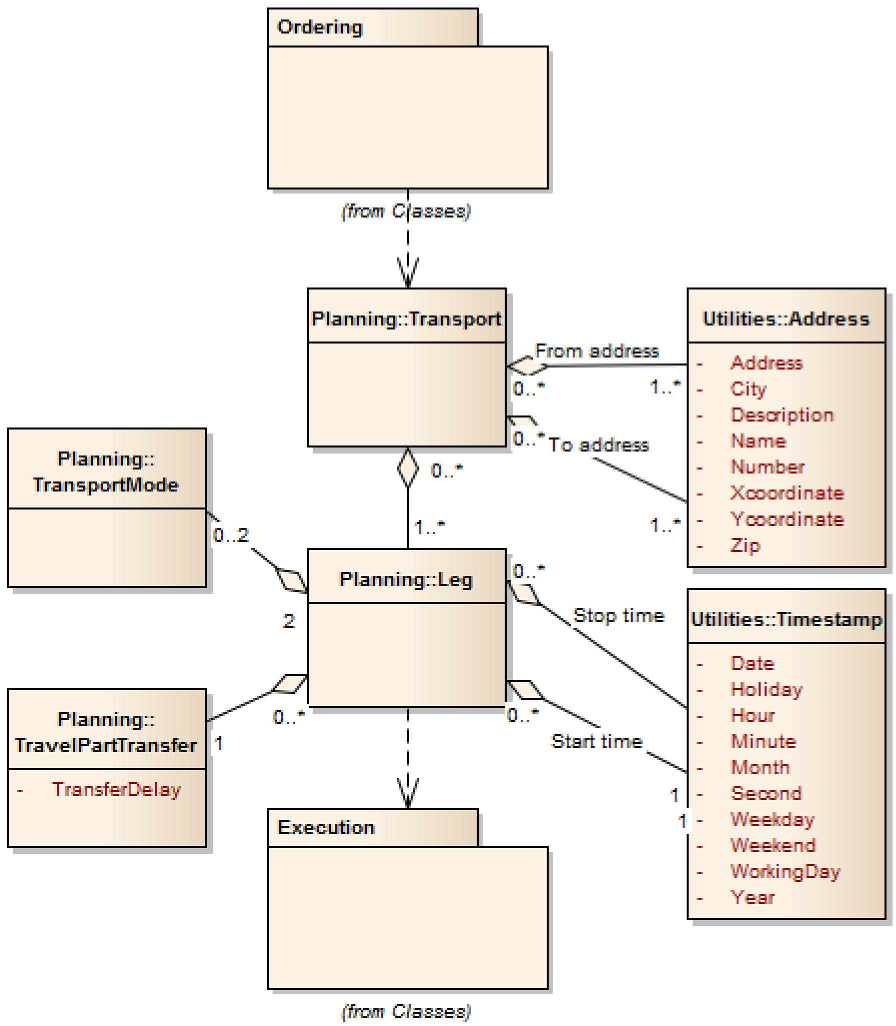

Figure 11 Planning Classes

When an *Order* has been places in the *Ordering* package, it is ready for being planned for later execution. The travel plan is being handled by *Transport*, which can have two connections, namely “from” and a “to” *Address* of the travel. A *Transport* can be divided into several parts, where one part is called a *Leg*. The reason for several *Leg* objects can exist for a *Transport* is that several types of *TransportMode* can be utilized for a *Transport*. Every *Leg* has one *TransportMode* attached describing whether, e.g. taxi or train is used. Also a *Leg* has a start and stop *Timestamp* describing when the leg is planned to begin and end being executed. At last a *TravelPartTransfer* can be attached to two *Leg* objects. This indicates a delay that might exist when switching between two *Leg* objects.

When the *Planning* package is completed, a transport is ready for being performed by the *Execution* package.

Classes

The classes of the *Planning* package describe how an order is planned for execution and contains information on entire transportation and every small step of a travel planned.

Leg

A *Leg* describes a part of a transport and a transportation of a patient can exists of many legs with different modes. Though a *Leg* can also describe a path where no passengers are being transported, so-called empty driving.

Transport

The *Transport* class keeps data describing how a transport is put together and is created from a *Travel Plan*. Whenever transportations are planned in the system, such a new *Transport* might be generated.

TransportMode

TransportMode describes what kind of transportation is being used. This can be different kinds of transportation types, such as bus, taxi or train.

TravelPartTransfer

Travel Part Transfer is a class that describes a delay which can occur between two different transfer legs. This delay is described by this class.

Connections

A *Transport* can have none or several *Leg* objects describing how a transport is put together, while a *Leg* can be attached to none or many *Transport* objects. None if for example a taxi is driving without passengers and many if a taxi is driving more than one passenger at a time.

A *Transport* can be attached to two *Address* objects, which describes from and to address of a *Transport*, while an *Address* does not need to be associated with any *Transport.*

A *Leg* can have up to two *TravelPartTransfer* objects, describing an initial and end transfer delay between two *Leg* objects. A *TravelPartTransfer* must be associated with two *Leg* objects, describing between which two *Leg* objects the delay is to exist.

A *Leg* can be aggregated from a Transport, though it does not necessarily need to be, if the vehicle is driving without passengers.

A *Leg* has two *Timestamp* describing when the *Leg* has started and stopped, though does a *Timestamp* not need to be associated with a *Leg*.

9.3.4 Execution

The *Execution* packages contain classes holding information of how a transport is executed in practice, which should reflect how a transport was planned. This package stores information of what vehicle was used for the transportation, which type the vehicle is, how the vehicle was scheduled along with information of additional services and timestamps of start and stops.

Description

A class diagram of the *Execution* packages can be seen below in Figure 12. Please note, that the *Timestamp* class is imported from the *Utilities* package.

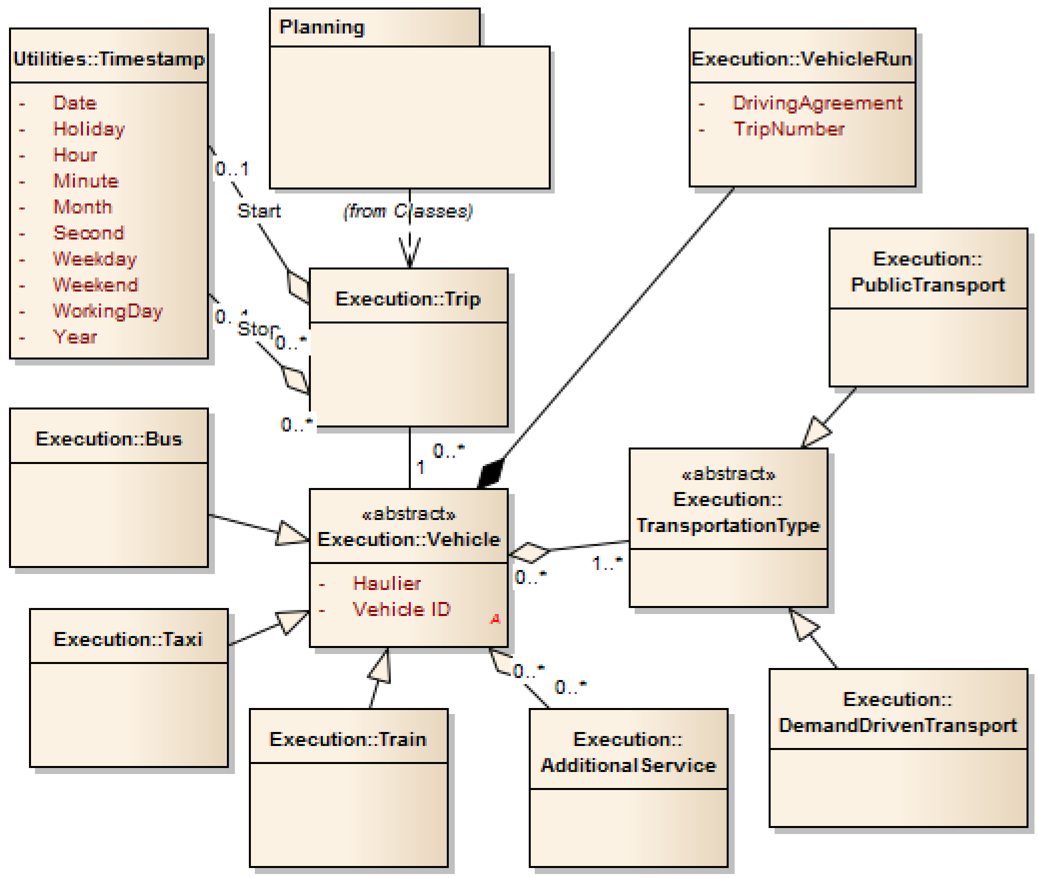

Figure 12 Execution Classes

When a *Trip* is to be executed, it is being executed one *Leg* at a time, fetched from the *Planning* package. A *Trip* stores the start and stop *Timestamp* of the executed *Trip*. A *Trip* is executed by a *Vehicle*, which is an abstraction of a *Bus*, a *Taxi*, or a *Train* and for every *Trip* an *AdditionalService* can be attached the *Vehicle*. The route a *Vehicle* has been driving over time is called a *VehicleRun*. A *Vehicle* can be executed as one of two kinds of *TransportationType*, namely as *PublicTransport* or *DemandDrivenTransport*.

Classes

The classes of the *Execution* package are describing how a *Trip* is being performed. The purpose of the contained classes is to describe in details what kind of *Vehicle* is being utilized along with when the *Trip* is going to start and end.

AdditionalService

This class describes whether some additional services are required from the *Vehicle*. This could be wide entrance, a wheelchair lift, or possibility of lying down.

Bus

A *Bus* is a specialization of the class *Vehicle*, describing that the *Vehicle* is a *Bus*. A *Bus* can often have several passengers and might have additional advantages of other means of transport.

DemandDrivenTransport

This class is a specialization of the *TransportationType* class, describing that the current transportation type is demand-driven. Demand-driven transportation is defined as when a transport is going to take place from one location to another, at a specific timestamp, and the location and timestamp is defined by the user. Demand-driven transportation is though very flexible.

PublicTransport

This class is a specialization of the *TransportationType* class, describing that the current transport type is public transportation. Public transportation is defined as a vehicle that travels between fixed pre-defined destinations at a static schedule. This kind of transportation type is very unflexible but has the ability of being reliable and predictable, and might also have the advantage of being able to carry several passengers.

Taxi

A *Taxi* is a specialization of the class *Vehicle*, describing that the *Vehicle* is a *Taxi*. A *Taxi* can fast and easily transport passengers between very specific places along with the driver can help servicing the passenger.

Train

A *Train* is a specialization of the class Vehicle, describing that the Vehicle is a Train. A *Train* is quite a non-flexible means of transport, having fixed departures from fixed locations which can be counted on. A *Train* is often fast when traveling over greater distances and can be cheaper to utilize due to great capacity.

TransportationType

This class is an abstract class describing what kind of means of transport the *Vehicle* is behaving as, scheduling and planning wise.

Trip

A *Trip* is the execution of a *Leg* from the *Planning* package. This class holds information of the execution of the *Trip* is started and stopped and a trip is executed using some kind of *Vehicle*.

Vehicle

A *Vehicle* is an abstract class for any kind of transportation medium that will carry a passenger from one location to another. The *Vehicle* is defined by the specializations of the vehicles and described by the associations of the *Vehicle.*

VehicleRun

A *VehicleRun* is a description of where a *Vehicle* has been driving over a period of time. This period could for example be a complete day of work for a *Taxi*, though for one day of work the path of a *Taxi* could also be divided into several *VehicleRun*.

Connections

A *Trip* can have two *Timestamp*, namely for when a *Trip* is started and when a *Trip* is ended.

A *Trip* must be associated with one *Vehicle*, while a *Vehicle* does not necessarily have to be associated with any *Trip* but can be associated with several *Trip* if several passengers (several *Trip*) are being executed by the *Vehicle* at the same time.

A *VehicleRun* is owned by a *Vehicle* in the sense that if a specific *Vehicle* ceases to exist the *VehicleRun* does not exist anymore either.

A *Vehicle* is an abstract class which is means it has to be specialized as a *Bus*, a *Taxi*, or a *Train*.

A Vehicle has at least one *TransportationType* and can actually have more than one if it changes between different *TransportationType.* A *TransportationType* does though not necessarily have to be associated with any *Vehicle.*

A *TransportationType* is an abstract class, which can be implemented as either *DemandDrivenTransport* or *PublicTransport*.

10. Related Work

10.1 EU Funded Projects

EuroFOT -- European Field Operational Test on Active Safety Systems (http://www.eurofot-ip.eu/)

The main purpose of the EuroFOT project is to assess the impact of advanced driver assistance devices on drivers. The main focus areas are to increase the safety, comfort and fuel efficiency of vehicles. Eight types of driver assistance devices are tested, and their interaction with the driver is monitored in order to determine that the outcome of these devices is as desired. The devices range from active safety devices, to adaptive cruise control, to passively assisting devices, e.g., fuel efficiency advisor.

This overlap of this WP with EuroFOT is related to fuel efficiency. EuroFOT focuses on training drivers to behave more environmentally friendly, whereas this WP focuses on constructing environmental profiles of existing infrastructure, and informing the driver of eco-routes.

CITYLOG -- Sustainability and efficiency of city logistics (http://www.city-log.eu/)

CITYLOG focuses on improving the efficiency and sustainability of what is termed *last mile transportation*. The project considers several challenges such as load optimization, optimizing delivery routes, and customer feedback on delivery times.

CITYLOG is complimentary to this WP, because whereas CITYLOG considers the efficiency of the last mile transport, this WP considers the overall efficiency of the transportation.

CVIS -- Cooperative Vehicle-infrastructure Systems (http://www.cvisproject.org)

The CVIS project aimed to provide open specifications for Vehicle to Infrastructure (V2I) communications and Vehicle to Vehicle (V2V) communications. By establishing open specifications for this communication the goals were to improve efficiency, lower congestion, and improve fleet management.

CVIS aimed at creating an infrastructure for V2V and V2I communications first, and building sample applications that leverages infrastructure. This is in contrast to this WP, which uses existing V2I for improving the performance of fleet management and lowering environmental impacts.

SafeTRIP -- Satellite Applications for Emergency handling, Traffic alerts, Road safety and Incident Prevention (http://www.safetrip.eu)

SafeTRIP exploits a new S-band satellite communication frequency, and integrates this with a GNSS signal. This results in a two-way global communication system able to perform vehicle tracking, automatic incident alerting, and speed-limit service. The primary focus of SafeTRIP is on improving safety, and constructing an infrastructure for this.

This WP distinguishes itself from SafeTRIP in that it considers eco-routes and eco-routing.

eCoMove -- Cooperative Mobility Systems and Services for Energy Efficiency (http://www.ecomove-project.eu)

The eCoMove project focuses on constructing a common infrastructure for V2I and V2V communication. Using this infrastructure, devices installed in vehicles will report GNSS data as well as energy consumption data. These data are then used in a number of services, e.g., eco-route planning, eco-coaching, eco-monitoring.

The main focus in eCoMove is very different from the focus of this WP, in that they focus on constructing an infrastructure, and build services on top of that. In this WP, existing infrastructure and data is used to estimate environmental impact. This significantly reduces the time to utilization of the work, because the support needed from vehicles is already in place.

TeleFOT -- Field Operational Tests of Aftermarket and Nomadic Devices in Vehicles (http://www.telefot.eu)

In the TeleFOT project, the impact of a range of consumer devices with different capabilities is evaluated. This impact is measured on fuel consumption, travel time, traffic jam avoidance, or percentage of trip where the speed is below the speed limits. The types of devices range from navigational to informational devices with speed limit and traffic information to green driving support devices.

This WP is similar to TeleFOT in that it also focuses on using existing devices, but whereas TeleFOT analyze the impact onto several parameters of these devices, this WP focuses on collecting data from such devices and constructing metrics for the road infrastructure.

10.2 Academic Papers

H. Ayed et al gives in [27] an abstract logical data model for calculating shortest paths in a so-called transfer graph. The transfer graph consists of a graph for each mode of transportation, and a set of vertices that are shared between graphs. These vertices are denoted transfer vertices, which mean that they allow for switching between transport modes. A similar model could be exploited in this project to allow for switching between different modes, with the advantage that each subgraph can be annotated with special features such as capacity, reliance and price. This will allow for shortest path queries that obey certain quality parameters.

C.S. Jensen et al propose in [28] a framework for distributed continuous nearest neighbor queries using network distance. The data models described throughout the paper is relevant and will to some extend be replicated in this project, especially the conversion from road network to graph.

H. Samet et al proposes in [29] a fast and scalable method of querying nearest neighbors in road networks. The method involves precomputing all shortest paths and an innovative use of quadtrees for compression of shortest paths. These combined allow for recomputing shortest paths in O(n), n being the number of edges in the shortest path. Additionally the paper describes a fast method of finding nearest neighbors in network distance, using these structures. In this project similar data structures, will be implemented and used for clustering pickup destinations which are close to each other, as measured by network distance.

Y. Ding and H. Rakha correlates in [30] different statistical variables with fuel consumption and emission of greenhouse gases is researched. They use a regression model and try to minimize the R2 error, i.e., the sum of all errors squared. The main parameters they discovered is speed and speed variability, which alone seem to explain up to 90% of the total emissions and fuel consumption. A number of additional important parameters are found such as kinetic energy and acceleration noise (disturbance of the traffic flow). Part of this project will be extending the set of parameters, by analyzing correlations between fuel consumption (obtained through Car Area Network) and parameters such as air condition, temperature and tire pressure.

Basic algorithms are given for solving the Multicriteria Shortest Path Problem (MSPP) in [31]. The most interesting part of the paper is its treatment of theoretical aspects of MSPP, in which several properties of the solution are given. These properties are relevant for this project and will be exploited and possibly extended.

In [32], F. Marchal et al propose an efficient but not highly accurate algorithm for map-matching large volumes of GNSS points. The algorithm uses a multi-hypothesis model, in which multiple hypothesis of the route followed by the driver is maintained. Eventually the best candidate is selected as the route actually followed by the driver. The authors claim a throughput of 2000 points per seconds for a realistic configuration. In [33] F.C. Pereira et al construct an accurate genetic map-matching algorithm with the capability of handling incomplete maps. They recognize that the algorithm described in [32] and their genetic algorithm are complimentary, and therefore integrate the two approaches. The result is an algorithm for map-matching that is both very efficient and very accurate. The code developed in [33] is open source, and will be exploited where possible in this project. Most of the data available in this project is not high frequency, which is required by [33], thus other approaches will be used when high frequency data is unavailable.

In [34] W. Wang et al describe a map-matching algorithm for low frequency GNSS data points, and use it to develop travel-time estimates along a large number of interstate road segments. The problem of map-matching GNSS points has been examined much, but mostly in high frequency logging settings. In this work the authors generate an error bounding box for each GNSS point, and find intersecting segments. From this set of segments candidate routes are generated and finally filtered using a fuzzy logic model. Evaluation shows an accuracy of 98.6% of the map-matching. The performance of their algorithm is measured to be around 175 points per second. While this approach is only tested on interstate road networks, a similar approach may have to be taken in this project in order to map-match the low frequency part of the data. It will be interesting to see the extent to which this is also applicable in road networks of finer granularity, as well as finding opportunities for lowering the time consumption of the algorithm.

G. Marketos et al propose in [35] a new type of data warehouse, namely the Trajectory Data Warehouse (TDW). Many database systems have been extended with data types for location based data, but the amount of data involved with many scenarios require special consideration in order to provide useful query times. To accommodate this, the authors propose an architecture and process for loading and processing input data before storing in a TDW. The authors propose a number of abstractions over the data, which allow for indexing in a database using tried and tested techniques. We will employ their techniques where relevant, especially their proposed process of loading data, dividing data into separate trips and map-matching the data before storage.

Ahn et al propose in [36] a fuel consumption and/or greenhouse gas emission estimation model that employs six categories of factors/variables: 1. Travel-related factors account for the distance and number of trip traveled within an analysis period, and average speed. 2. Weather-related factors account for temperature, humidity and wind effects. 3. Vehicle-related factors account for numerous variables [37]including the engine size, the condition of the engine, etc. Normally, different types of vehicles have different vehicle-related factors. 4. Roadway-related factors account for roadway grade, surface roughness, local neighborhood where the roadway is located, speed limit, etc. 5. Traffic-related factors account for the congestion status, which can be reflected by instantaneous speed levels. 6. Driver-related factors account for differences in driver behavior and aggressiveness, which can be reflected by instantaneous acceleration levels.

Different fuel consumption/ greenhouse gas emission estimation models can be classified into two general categories: macroscopic estimation models and microscopic estimation models. **Macroscopic estimation models** usually work well for predicting emissions for large fleets in large regional areas. It has been accepted by transportation planners for the evaluation of network-wide highway impacts on the environment; it is not suitable for the evaluation of energy and environmental impacts of operational-level projects. The typical macroscopic estimation models include MOBILE6 [38] and MOVES [39] developed by the U.S. Environmental Protection Agency (EPA), and EMFAC [40] developed by the California Air Resources Board (CARB). These models normally employ travel-related factors (e.g., average speed, route length, etc.), vehicle-related factors (e.g., vehicle types, model years, etc.), and weather-related factors (e.g., ambient air temperature, etc.). However, the macroscopic models do not consider traffic-related and driver related factors, such as transient vehicle behavior along a route (e.g., instantaneous speed and acceleration levels).

Microscopic estimation models are designed to evaluate the environment impacts of operating modes of individual vehicles. The primary microscopic estimation models include VeTESS [39] developed in the European 5th Framework Programme project DECADE, CMEM [41] (Comprehensive Modal Emissions Model) developed by the University of California, Riverside, and VT-Micro [36] developed by Virginia Tech. These models usually estimate the environmental impact second by second for a specific trip traversed by a particular vehicle. They normally employ the detailed speed profile of the trip, including instantaneous speed and acceleration every second, which belong to traffic-related and driver-related factors. VeTESS and CMEM models also employ vehicle-related factors, meaning that they utilize different parameters for different types of vehicles. On the contrary, VT-Micro model only utilizes the instantaneous speed and acceleration to predict the environmental impacts.

MOBILE6 and MOVES: These two models offer simplified mathematical expressions to compute fuel and emission rates based on average link speeds. Meanwhile, they maintain a well-categorized vehicle types and corresponding emission factors at different model years. The models can also distinguish variable start operation, such as cold start and hot start.

Microscopic Engine-map Model: Microscopic engine-map based estimation model are capable of estimating second by second tailpipe emission and fuel consumption. The model normally requires a vehicle’s detailed speed profile (second by second), an engine map based on the type of the vehicle’s engine, and some other vehicle related parameters. An engine map takes as input engine speed, engine output torque, torque change and some engine related parameters, and it outputs estimated fuel consumption and emissions. Different engine maps should be maintained for different types of engine.

This category of estimation models include VeTESS [39] which is developed in the European 5th Framework Programme project DECADE; CMEM [41] (Comprehensive Modal Emissions Model) developed by the University of California, Riverside; VETO which is developed at the Swedish National Road and Transport Research Institute; Phem which is developed at Graz University.

We take VeTESS model as an example to show how such model works. Given a trip (speed profile and gradient over the whole route) of a vehicle, VeTESS model first computes the total force on the vehicle by considering the acceleration resistance, climbing resistance, etc. Based on different gear change rules recorded in VeTESS, the engine torque and the speed of engine flywheel are calculated. Finally, a corresponding engine map (based on the engine type of the vehicle) is employed to estimate the fuel consumption and emission of the trip for the vehicle.

Roadway classifications method [42]**:** Streets in a road network (i.e., roadway-related factors) are classified into different categories based on the following criteria: 1. Street function: either as part of a main road or as part of the local road network. 2. Type of environment: defined by the settlement in the local neighborhood in which the street is located. For example, city center, residential area, suburb, etc. 3. Speed limit. 4. Density of junctions controlled by traffic lights. 5. Traffic-calming measures: with and without. 6. Traffic-flow: (1) less than 200 vehicles/hour per lane; (2) between 200 and 700 vehicles/hour per lane; (3) more than 700 vehicles/hour per lane. (4) Not available. 7. Peak time and off-peak time.

A collection of driving data (vehicle speed, engine speed, and GNSS positioning data) is first divided into different partitions according to the categories of the streets where the data is captured. For each partition of the driving data, a microscopic engine-map based estimation model makes estimation on detailed speed profiles and specific vehicle parameters for the individual vehicles. In this paper, the authors tried several different models, including e.g., VT-Micro [36] [37].

VT-Micro : This model aims at estimating environmental impacts for different operating modes of individual vehicles. It considers traffic-related and driver-related factors by using instantaneous speeds and accelerations. The fuel consumption is non-linear w.r.t. the vehicle speed and acceleration. However, it is generally true that as acceleration and speed increase, the fuel consumption also tend to increase. A linear regression model which incorporates linear, quadratic and cubic of speed (denoted as s) and acceleration (denoted as a) is proposed [36]. The environmental impact (denoted as EI) is not estimated directly by this model, but a transformed value using nature logarithm (denoted as ln(EI)).

Two different sets of coefficients (denoted as $L_{i,j}$ and $G_{i,j}$) are trained for accelerating (a > 0) and decelerating scenarios (a < 0), respectively.

$$\ln(EI) = \sum_{i=0}^{3}\sum_{j=0}^{3}(G_{i,j} \times s^i \times a^j), \quad for\ a < 0.$$

$$\ln(EI) = \sum_{i=0}^{3}\sum_{j=0}^{3}(G_{i,j} \times s^i \times a^j), \quad for\ a < 0.$$

This model uses purely instantaneous speed and acceleration, but not any other parameters w.r.t. types of vehicles, etc. Therefore, this model can be directly deployed on GNSS data.

Table 1 summarizes different factors that are used by different models. If only GNSS data is available, average speed (travel-related factor), instantaneous speed and acceleration are able to be derived. Thus, VT-Micro model seems to be the best choice in this scenario. The other models can also be applied if default values are used for the unavailable factors. If road network and corresponding metadata is also available, streets in the road network can be classified according to some criteria [16]. This may result in high accuracy.

	Category	Travel	Weather	Vehicle	Roadway	Traffic	Driver
MOVES	Macro	Yes	Yes	Yes	Yes/No	No	No
EngineMap	Micro	No	No	Yes	No	Yes	Yes
RoadwayCl	Micro	No	No	Yes	Yes	Yes	Yes
VT-Mircro	Micro	No	No	No	No	Yes	Yes

Table 7 Model versus Factor

J. Booth et al. propose a data model [43] for trip planning in multi-modality transportation systems, in which pass ways for pedestrians, road network for vehicles, bus routes for buses, railways for urban and suburban trains are modeled. The model not only maintains the connectivity and accessibility of each transportation network, but also records the geometry of each transportation network. What is more, the model can also deal with other attributes associated with the transportation network, such as POIs along road segments; and some aspects of the data that are probabilistic in nature, e.g., the speed for vehicles that traversed on specific road segments are presented as expected speed and variance. A trip – a path between an origin and destination subject to some transportation mode constraints – is proposed as the central concept in the model. The trips and transportation networks are represented as both a graph and relation model, and a set of operators are integrated within a SQL-like syntax. We may reuse such modeling method to model the multi-modality transportation network, in particular, the road network, bus routes and railways.

J Xu et al. in their recent work [44] studied modeling the infrastructures for multimodal moving objects. They are particularly interested in generating synthetic multimodal infrastructures that enable and constrain the movements of moving objects. In particular, pavements for pedestrians, zebra crossings, bus routes, bus stops, and indoor floor plans are modeled and each modal of transportation can be referenced with each other.

Modeling a road network (e.g., a single modality transportation network) is discussed in the work by C.S. Jensen [28]. In Link Node representation, a road network is simply modeled as a directed graph, where vertices (nodes) indicate intersections, and edges (links) indicate road segments (routes) connecting two vertices. A directed graph captures the topology of the road network, which is usually enough for traffic and route planning; however, it cannot maintain some information that may also be useful for other applications. Kilometer-post representation for modeling a road network is typically used for road administration, which is convenient for relating a physical location in the network to a known location stored in a database, e.g., kilometer-post. Geographical representation captures the geographical coordinates of the transportation infrastructure. Specifically, the three dimensional points on the center lines of roads are recorded into a database. Segment representation models a road network as a collection of segments that intersect at connections, which preserves the network topology and captures the complete set of roads.

R. H. Güting proposed a road network model [45] which takes routes and junctions as fundamental elements. The model offers two kinds of routes: simple routes and dual routes. Simple routes are modeled for those applications which do not care about distinguishing positions on two sides of a road, and vice versa for the dual routes. Route measure and route location are provided as two concepts for locations on different kinds of routes. Furthermore, turn information at each junction (whether U-turn is possible, left turn only, etc) can also be modeled and maintained in a so-called transition matrix.

11. Requirements

This section lists the requirements to the system.

This system must be able to input (insert or update) the following.

1. Load/update basic map
    a. The system must be able to load a basic transport network in the form of a digital map. The digital map is the foundation for computing eco-routes for various vehicle types. It must be possible to update the digital map.
2. Load/update specialized map
    a. The system must be able to load a specialized transportation networks. These maps are different from the basic map because for example for busses it must contain special bus roads and bus stop. For trains it must contain the railway tracks and the train stations.
3. Load GNSS Measurements in Batch
    a. The system must be able to load and use both low-frequency and high-frequency GNSS data.
4. Load GNSS Measurements in Real-Time
    a. The system must be able to process large quantities of GNSS data in real-time. Such GNSS data is assumed to be provided in a stream-like fashion.
5. Load CANBus data in batch
    a. The system must be able to load CANBus and associated GNSS data.
6. Load CANBus Data in Real-Time
    a. The system must be able to load CANBus and associated GNSS data fast enough to be able to provide traffic-related information in real time.
7. Load Passenger Requests
    a. The system must facilitate that it is simple to upload the transportation requests made by the users.
8. Load Tour Related Data
    a. The system must be able to load the trips made by a fleet of vehicles to produce the output. This data is from the trip scheduling system.
9. Convert GNSS Measurements to GHG Emission
    a. Providing an estimate of the GHG emission based on GNSS measurements is a basic functionality of the system.
10. Combine GNSS Measurements and CANBus Data
    a. The system must be able to take advantages of both GNSS measurements and CANBus data if both types of data is available in a region of the map.
11. The system must be able to use GHG emission estimated for one region based on GNSS measurements from another region using similarities in the road network.
    a. This ensures that coverage of the estimation is as wide as possible without requiring that GNSS measurements are available for, e.g., all motorways in Europe.
12. Annotate Existing Map Technology with Eco Routes

   a. This is to be able to disseminate the knowledge in the project to the largest possible set of interested persons and organizations. By using existing map technology and the knowledge about GHG emission the impact of the REDUCTION project can be maximized.

13. Estimate GHG Emission for a Single Trip.
    a. This is to be able to compare and contrast individual trips.
14. Compare and Contrast GHG Emission for a Single Trip
    a. The system must be able to evaluate the GHG emission of a single trip and compare it e.g., to alternative route or alternative time of day. This includes a comparison between individual traffic using taxis to multi-modal transportation using, e.g., a combination of busses and taxis or only busses.
15. Estimate GHG emission for a set of Trips
    a. This is to compare the impact of multi-modal transportation, e.g., a partial use of busses compared to using only taxis.
16. Compare and Contrast GHG Emission for a Single Trip
    a. This is to evaluate if a fleet of vehicles are being used efficiently. This includes the usage of multi-modal transportation.
17. Evolution of Eco-Routes
    a. The system must record over time if the GHG emission of a the same trip changes, e.g., after a change in the road-network infrastructure is going from point A to point B changed.

The mapping of requirements to use cases is shown in Table 8.

Requirement	Use Case
Load/update basic map	Load Basic Map
Load/update specialized map	Load Specialized Map
Load GNSS Measurements in Batch	Load GNSS Data
Load GNSS Measurement in Real-Time	Load GNSS Data
Load CANBus Data in Batch	Load CANBus Data
Load CANBus Data in Real-Time	Load CANBus Data
Load Passenger Requests	Load Passenger Requests
Load Tour Related Data	Load Tours
Convert GNSS Measurements to GHG Emission	Build Eco-Map
Combine GNSS Measurements and CANBus data	Build Eco-Map
Annotate Existing Map Technology with Eco Routes	Build Eco-Map
Estimate GHG Emission for a Single Trip	Evaluate Single Trip
Compare and Contrast GHG Emission for a Single Trip	Evaluate Single Trip
Estimate GHG emission for a set of Trips	Evaluate Fleet Trips
Compare and Contrast GHG Emission for a Set of Trips	Evaluate Fleet Trips
Evolution of of Eco-Routes	Update Basic Map

Table 8 Mapping Requirement to Use Cases

Note that in Table 8 all requirements are covered by use cases. In addition, all use cases are used. We therefore conclude that the system designed can fulfill the requirements listed and that there is no functionality of the system not being used.

Table 9 compares the business goals to the requirements. Only the most relevant requirements are added. In addition, note that all the requirements related to establishing the data foundation, e.g., load of GNSS measurements, CANBus data, and map data is not included in the table. These requirements are a foundation for doing the computation and are a requirement for all the business goals and therefore note added to Table 9.

Based on Table 9 we conclude that the system has the functionality required to fulfill the business goals.

Business Goal	Requirements
Reduction of GHG Emission from vehicles used in flex-traffic	• Convert GNSS Measurements to GHG Emission • Combine GNSS Measurements and CANBus data
Establishing environmental profiles of vehicle types.	• Combine GNSS Measurements and CANBus data
Estimation of GHG emissions based on GNSS measurements.	• Convert GNSS Measurements to GHG Emission • Compare and Contrast GHG Emission for a Single Trip
Estimation of the GHG emissions of a single vehicle.	• Estimate GHG Emission for a Single Trip
Enabling the use of GHG emissions goals in public procurement	• Annotate Existing Map Technology with Eco Routes
Enabling of multi-modal planning of trips	• Annotate Existing Map Technology with Eco Routes • Estimate GHG emission for a set of Trips • Compare and Contrast GHG Emission for a Set of Trips

Table 9 Comparing Business Goals and Requirements

12. Conclusion

This report has listed the requirements to the software prototypes (the system) that are evaluated in the field trials in WP5 of the REDUCTION project (Task 5.2 and Task 5.5). The system has been defined including the system boundaries.

Based on the business requirements and a set of use cases the overall design of the system has been provided using the UML notation. This gives a solid foundation for the implementation of the actual software.

The main risks have been identified and estimated. Two main risks have been identified lack of CANBus data and no access to bus/train information. Actions have been started to mitigate these risks.

To the best of our knowledge this deliverable has contributed to state-of-the-art in the following areas.

- Provided an overview of existing formats for GNSS and CANBus data exchange from vehicles
- Provided an overview of related work within eco-routing for both EU-funded projects and academic papers
- Provided a design of a complete system for handling both GNSS and CANBus data that is highly flexible.
- Provided a risk assessment of what data must be available to realize a generic and flexible system for computing eco routes.

References

[1] Y. Wen, Y. Lu, J. Yan, Z. Zhou, K. von Deneen and P. Shi, "An Algorithm for License Plate Recognition Applied to Intelligent Transportation System," vol. 12, no. 3, 2011.

[2] K. Kanayama, Y. Fujikawa, K. Fujimoto and M. Horino, "Development of vehicle-license number recognition system using real-time image processing and its application to travel-time measurement," 1991.

[3] H. Ahmed, M. El-Darieby, Y. Morgan and B. Abdulhai, "A Wireless Mesh Network-Based Platform for ITS," 2008.

[4] H. Sawant, J. Tan, Q. Yang and Q. Wang, "Using Bluetooth and sensor networks for intelligent transportation systems," 2004.

[5] B. Li, "Recursive estimation of average vehicle time headway using single inductive loop detector data," vol. 46, no. 1, 2012.

[6] C. Le Bastard, D. Guilbert, A. Delepoulle, A. Boubezoul, S. Ieng and Y. Wang, "Vehicule identification from inductive loops application : Travel time estimation for a mixed population of cars and trucks," 2011.

[7] Y. Li, M. Miller and M. Cassid, "Improving Mobility through Enhanced Transit Services: Transit Taxi Service for Areas with Low Passenger Demand Density," 2009.

[8] K. Uchimura, H. Takahashi and T. Saitoh, "Demand responsive services in hierarchical public transportation system," vol. 52, no. 4, 2002.

[9] Informational Agency of the Republic of Tatarstan, "Sergei Ivanov confirms introducing import duties on GPS receivers," 18 July 2009. [Online]. Available: http://eng.tatar-inform.ru/news/2009/07/18/26149/. [Accessed 15 February 2012].

[10] "GPX: The GPS Exchange Format," [Online]. Available: http://www.topografix.com/gpx.asp. [Accessed 28 2 2012].

[11] Garmin Inc., "Garmin GPX Extensions Format v3," [Online]. Available: http://developer.garmin.com/schemas/gpxx/v3/. [Accessed 28 2 2012].

[12] Garmin Inc., "Traning Center Database XML," 12 05 2007. [Online]. Available: http://developer.garmin.com/schemas/tcx/v2/. [Accessed 28 2 2012].

[13] Garmin Inc., "Activity XML," 13 5 2008. [Online]. Available: http://developer.garmin.com/schemas/axm/1.0/. [Accessed 28 2 2012].

[14] OGC, "OGC KML Version 2.2.0," 14 4 2008. [Online]. Available: http://www.opengeospatial.org/standards/kml. [Accessed 28 2 2012].

[15] National Marine Electronics Association, "NMEA 0183 Standard," [Online]. Available: http://www.nmea.org/content/nmea_standards/nmea_083_v_400.asp.

[16] National Marine Electronics Association, "NMEA 2000® Standard," [Online]. Available: http://www.nmea.org/content/nmea_standards/nmea_2000_ed2_20.asp.

[17] W. Gurtner and L. Estey, "RINEX The Receiver Independent Exchange Format V 3.01," NASA, 2009.

[18] UNAVCO, "BINEX: Binary Exchange Format," 16 2 2011. [Online]. Available: http://binex.unavco.org/binex.html. [Accessed 28 2 2012].

[19] GPSBabel.org, "GPSBabel - Free software for GPS data conversion and transfer.," [Online]. Available: http://www.gpsbabel.org/.

[20] "On-board diagnostics," [Online]. Available: http://en.wikipedia.org/wiki/On-board_diagnostics.

[21] "Engine control unit," [Online]. Available: http://en.wikipedia.org/wiki/Engine_control_unit.

[22] "Controller area network," [Online]. Available: http://en.wikipedia.org/wiki/Controller_area_network.

[23] ARINC 825, [Online]. Available: http://www.arinc825.com/.

[24] Environmental Systems Research Institute, Inc, "ESRI Shapefile Technical Description," 1998. [Online]. Available: http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf.

[25] C. Moeller, "osm2po - openstreetmap converter and routing engine for java," [Online]. Available: http://osm2po.de/.

[26] Pitney Bowes Software Inc, "MapInfo® is now Pitney Bowes Software," [Online]. Available: http://www.pbinsight.com/welcome/mapinfo/.

[27] D. K. Z. H. P. B. a. J. M. H. Ayed, "Transfer graph approach for multimodal transport problems," 2008.

[28] J. K. T. P. I. T. C.S. Jensen, "Nearest neighbor queries in road networks," 2003.

[29] J. S. a. H. A. H. Samet, "Scalable network distance browsing in spatial databases," 2008.

[30] Y. D. a. H. Rakha, "Trip-based explanatory variables for estimating vehicle fuel consumption and emission rates," 2002.

[31] E. Martins, "On a Multicriteria Shortest Path Problem," 1984.

[32] J. H. K. A. F. Marchal, "Efficient Map Matching of Large Global Positioning System Data Sets: Tests on Speed-Monitoring Experiment in Zürich," 2006.

[33] H. C. N. P. F.C. Pereira, "An Off-line Map-matching Algorithm for Incomplete Map Databases," 2009.

[34] J. J. B. R. X. G. W. Wang, "Large-Scale Freeway Network Traffic Monitoring: A Map-Matching Algorithm Based on Low-Logging Frequency GPS Probe Data," 2011.

[35] E. F. I. N. N. P. A. R. Y. T. G. Marketos, "Building Real-World Trajectory Warehouses," 2008.

[36] K. Ahn, H. Rakha, A. Trani and M. Van Aerde, "Estimating vehicle fuel consumption and emissions based on instantaneous speed and accelerationlevels," vol. 128, no. 2, 2002.

[37] K. Ahn and H. Rakha, "The effects of route choice decisions on vehicle energy consumption and emissions," vol. 13, no. 3, p. 151–167, 2008.

[38] MOBILE6, [Online]. Available: http://www.epa.gov/otaq/m6.htm.. [Accessed 29 02 2012].

[39] l. Pelkmans, P. Debal, T. Hood, G. Hauser and M. Delgado, "Development of a simulation tool to calculate fuel consumption and emissions of vehicles operating in dynamic conditions," 2004.

[40] EMFAC, [Online]. Available: http://www.dot.ca.gov/hq/env/air/pages/emfac.htm. [Accessed 29 02 2012].

[41] M. Barth, C. Malcolm, T. Younglove and N. Hill, "Recent validation efforts for a comprehensive modal emissions model.," vol. 1750, no. 1.

[42] E. Ericsson, H. Larson and K. Brundell-Freij, "Optimizing route choice for lowest fuel consumption-potential effects of a new driver support tool.," vol. 14, no. 6, pp. 369-383.

[43] J. Booth, A. Prasad Sistla, O. Wolfson and I. F. Cruz, "A data model for trip planning in multimodal transportation systems," *EDBT,* pp. 994-1005, 2009.

[44] H. Xu and R. H. Güting, "Infrastructures for Research on Multimodal Moving Objects.," *Mobile Data Management,* vol. 1, pp. 329-332, 2011.

[45] R. H. Güting, V. Teixeira de Almeida and Z. Ding, "Modeling and querying moving objects in networks," *VLDB J.,* pp. 165-190, 2006.

[46] C. S. Jensen, T. B. Pedersen, L. Speicys and I. Timko, "Data Modeling for Mobile Services in the Real World," *SSTD,* pp. 1-9, 2003.