eCO-friendly urban Multi-modal route PlAnning Services for mobile uSers

FP7 - Information and Communication Technologies

Grant Agreement No: 288094
Collaborative Project
Project start: 1 November 2011, Duration: 36 months

D4.1.1 – Communication Interfaces supporting the most prominent standards

Work package:	WP4 – Content Gateway Module
Due date of deliverable:	30 April 2012
Actual submission date:	30 April 2012
Responsible Partner:	PTV
Contributing Partners:	PTV, CTI, CERTH

Nature: ☒ Report ☐ Prototype ☐ Demonstrator ☐ Other

Dissemination Level:

☒ PU : Public
☐ PP : Restricted to other programme participants (including the Commission Services)
☐ RE : Restricted to a group specified by the consortium (including the Commission Services)
☐ CO : Confidential, only for members of the consortium (including the Commission Services)

Keyword List: Transportation data, formats and standards, data representation, data transmission, web-based traffic info systems

The eCOMPASS project (www.ecompass-project.eu) is funded by the European Commission, Information Society and Media Directorate General, Unit G4-ICT for Transport, under the FP7 Programme.

The eCOMPASS Consortium

Computer Technology Institute & Press "Diophantus" (CTI) (coordinator), Greece

Centre for Research and Technology Hellas (CERTH), Greece

Eidgenössische Technische Hochschule Zürich (ETHZ), Switzerland

Karlsruher Institut fuer Technologie (KIT), Germany

TOMTOM INTERNATIONAL BV (TOMTOM), Netherlands

PTV PLANUNG TRANSPORT VERKEHR AG. (PTV), Germany

Document history			
Version	Date	Status	Modifications made by
0.1	07.03.2012	First draft	Dionisis Kehagias
0.2	13.03.2012	Second draft	Dionisis Kehagias, Florian Krietsch
0.3	24.03.2012	Third draft	Christos Zaroliagis
0.9	18.04.2012	Fourth draft	Dionisis Kehagias, Florian Krietsch
1.0	18.04.2012	Sent to internal reviewers	Dionisis Kehagias, Florian Krietsch
1.1	25.04.2012	Reviewers' comments incorporated (sent to PQB)	Dionisis Kehagias, Florian Krietsch
1.2	27.04.2012	PQB's comments incorporated	Dionisis Kehagias, Florian Krietsch
1.3	30.04.2012	Final (approved by PQB, sent to the Project Officer)	Dionisis Kehagias, Florian Krietsch

Deliverable manager

- Florian Krietsch, PTV

List of Contributors

- Florian Krietsch, PTV
- Christos Zaroliagis, CTI
- Dionisis Kehagias, CERTH

List of Evaluators

- Moritz Baum, KIT
- Felix Koenig, TomTom

Summary

The purpose of this deliverable is to review all appropriate data representation formats and relevant standards, which are associated with the information sources that will be received as input by the Common Gateway Module (to be developed in WP4), as well as algorithms (to be developed in WP2 , WP3). The goal of this review is to provide sufficient information for the selection of the data formats that will be supported by eCOMPASS. The deliverable contains a classification of multi-modal traffic and travel data types by describing the data resource categories that should be addressed in eCOMPASS for providing appropriate input data. The presented categories include: Public Transportation Data, Traffic Data, User Location Data, Points of Interest, Fleet Management Data, and Environmental Data. The characteristics of each of the aforementioned categories are presented and then a survey of the state of the art on relevant standards and existing representation formats follows. Furthermore, the deliverable describes proprietary and other relevant data formats and provides a list of publicly available traffic and transportation data provision services and relevant environmental data, as well as available APIs, if supported. The deliverable concludes by summarising the most prominent data representation formats and relevant standards, highlighting those aspects that are deemed most appropriate for eCOMPASS, and suggests a preliminary selection of standards and formats.

Table of Contents

1 Introduction

1.1 Purpose and Scope

The purpose of this deliverable is to review all appropriate data representation formats and relevant standards, which are associated to the information sources that will be received as input by the integrated eCOMPASS components. Based on this review, the most appropriate data representation formats will be selected that will be supported by eCOMPASS. In the next version of this deliverable (i.e. deliverable D4.1.2) the corresponding software interfaces will be developed and described that will provide support for the selected data representation formats and also for supported relevant standards. This deliverable conducts a survey of the available data representation formats and standards. The final selection will also take into account the functional, data and non-functional requirements of the eCOMPASS platform that are recorded in Deliverable D1.1. Special care will be taken in order to address the most prominent standards and representation formats for the various data types that will be supported by the project.

1.2 Summary and Objectives of WP4

Deliverable D4.1.1 is part of WP4, whose main is to implement the so-called *Content Gateway Module (CGM)* and all required interfaces that will enable interoperability with external services and information resources. The role of CGM (see also Section 1.3) is to enable interoperability with external services and information resources that are required by the algorithms to be developed in WP2, WP3. The objectives of WP4 are:

- The development of appropriate interfaces to enable interoperability with external data, web applications, and services.
- The development of a querying mechanism and the supporting application programming interface in order to allow the submission of requests for content by the integrated components and the delivery of requested content to the requesting party.
- The establishment of a distributed semantic repository of all registered searchable content.
- The development of a security mechanism to safeguard data exchange between the CGM and the external information resources.

1.3 Content Gateway Module

The CGM is a key component in the overall eCOMPASS architecture (see Figure 1), whose purpose is to handle all incoming requests for data from the integrated eCOMPASS components that require communication to external information resources, and also to manage all required activities in order to deliver the right content to the right requesting peer.

The CGM will be implemented as an interconnected environment of distributed nodes that will enable interoperability between the eCOMPASS framework and external data sources. Within the CGM a set of appropriate software and data representation interfaces will be developed to ensure that the most accepted data representation standards and access protocols will be supported.

The CGM is equipped with appropriate data representation interfaces for all common data types that will be supported by the end user applications. Supported data include information from web services and web applications, Global Position System (GPS)-based localisation sensors and data from other environmental sensors (e.g. to measure CO_2 levels). Other relevant information may become available through a web-based secure interface. The

CGM will be developed as a distributed network of gateway nodes that will allow remote interoperability between available information resources and the applications to be developed for the end users. CGM resides on top of these applications and allows integration of data and services into them. The supported services to be developed within eCOMPASS include:

- Vehicle navigation services
- Fleet management services
- Multimodal Routing services

All of the above services will be developed within WP5 and will integrate the optimised vehicle routing algorithms and multimodal human mobility algorithms to be produced as the outcome of the technical workpackages WP2 and WP3, respectively. All of the above services will be provided in the form of mobile applications in order to allow interaction to the user on-the-go. As the user requests real time content on the go (e.g. the urban traffic status) the CGM is activated and ensures that the right content is delivered to the right user on time. Appropriate information is also stored in a supported data repository that is used for caching purposes in order to allow more efficient and faster access to the requested data.

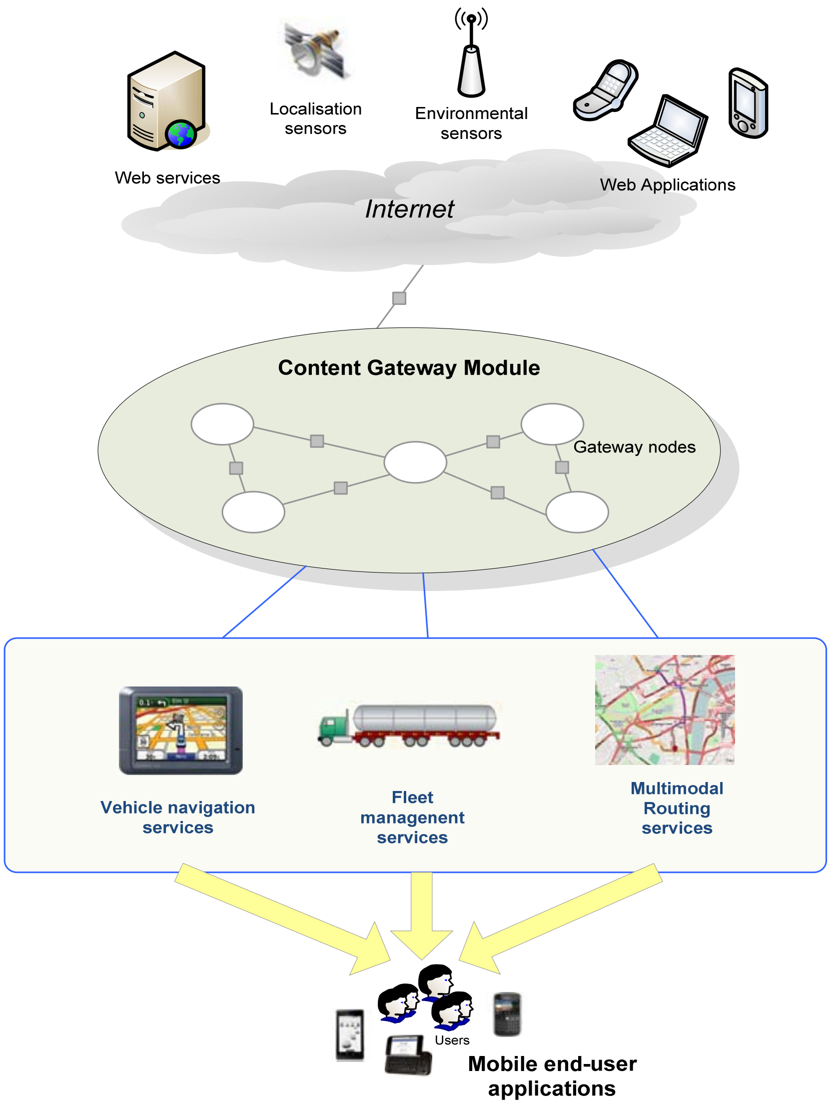

Figure 1: The eCOMPASS architecture

1.4 Structure of Deliverable

The rest of this deliverable is structured as follows. Section 2 contains a classification of multi-modal traffic and travel data by describing the types of such data that should be taken into account for the selection of the eCOMPASS supported data types. The presented categories include: Public Transportation Data, Traffic Data, User Location Data, Points of Interest (POI), Fleet Management Data and Environmental Data. This section describes each one of the aforementioned categories by underlining each data type characteristics. Section 3 provides a survey of the state of the art on standards and existing representation formats for each one of the data categories presented in Section 2. Section 4 describes proprietary or other relevant data formats. Section 5 gives a list of publicly available traffic and transportation data provision services and relevant environmental data, and also reports on the possibly available APIs. Finally, Section 6 concludes on the most prominent data representation formats and relevant standards, with focus on those standards that are deemed the most suitable for eCOMPASS, and suggests a preliminary selection of standards and formats.

2 Classification of Multimodal Traffic and Travel Data

2.1 Data Categories

This section introduces the various categories of data that are examined for possible support by the eCOMPASS system, and describes their characteristics. All categories adhere to the information needs of the various components and integrated applications that are going to be developed within the project, as these are derived by the eCOMPASS Description of Work (DoW). Of course, further refinement and a final selection will be performed after taking into account the user needs and functional, as well data and non-functional requirements that are recorded in the context of deliverable D1.1.

Section 3 reviews state of the art on data formats for the following types of data:

- ***Public Transportation Data***, including data from timetables about buses, trains, etc., as well as possible software or web interfaces via which the data may be accessed in a public or other manner.
- ***Traffic Data***, real time traffic data for specific areas in which the drivers are driving/moving.
- ***User Location Data***, including representation standards for information on user localisation, e.g. GPS data.
- ***Points of Interest***, information about specific places of particular interest that the user meets en route, or places that are located within the broader area, in which the users are moving/travelling, etc. Examples include restaurants, gas stations, rest areas, etc.
- ***Fleet Management Data***, i.e., data concerning communication between drivers in vehicles that are part of a fleet (e.g. trucks, buses, etc.) and the control centre, as well as additional data relevant to the fleet (e.g. travel logistics, mission-related information, etc.).
- ***Environmental Data*** that can be taken into account in order for the optimal eco-friendly route to be calculated (e.g. CO_2 emission, weather info, etc.).

More specific descriptions and data characteristics for each one of the aforementioned data types are presented in the remaining Section.

2.2 Public Transportation Data

About **60** billion passenger journeys were made by public transport in **2008** in the EU-27. This figure refers to local public transport, including urban, suburban and regional public transport services. Considering the total EU population, this represents about **120** public transport journeys per inhabitant per year (Berg Insight, 2012). Demand for public transport is not distributed evenly across the European territory. For instance, in the medium and large sized cities included in the International Association of Public Transport (UITP) Mobility in Cities Database (UITP, 2012) the number of public transport journeys was about **300** per inhabitant per year.

Public transport ridership has increased steadily in the last 10 years in many countries. Between **2004** and **2008,** ridership rose by about **11%** in Spain, the UK and the USA. Cities such as London and Brussels recorded particularly high ridership increases - about **20%** - during the same period. In France, excluding Paris, the number of passenger journeys increased by about **12%** between **2006** and **2008** alone. The contribution of public transport to the economy can be estimated at between **130** and **150** billion EUR, which represents **1 - 1.2%** of the EU's GDP. This figure represents the value created along the public transport supply chain. It does not include the wider impact of public transport on the economy.

Public transport services rely increasingly on information systems to ensure reliable, efficient operation and widely accessible, accurate passenger information. Well-defined, open interfaces have a crucial role in improving the economic and technical viability of Public Transport Information Systems of all kinds. Using standardised interfaces, systems can be implemented as discrete pluggable modules that can be chosen from a wide variety of suppliers in a competitive market, connecting diverse systems; rather than as monolithic proprietary systems from a single supplier. Interfaces also allow the systematic automated testing of each functional module, vital for managing the complexity of the increasingly large and dynamic systems. Furthermore, with a well defined, version interface, individual functional modules can be replaced or evolved, without unexpected breakages of obscurely dependent function.

The information on the public transport supply mainly consists of three categories of data:

- spatial data: city and network maps, description of routes and service patterns, location of points of sale, etc.
- temporal data: frequency of services, departure and passing times at stop points, service time spread, etc.
- information on fares: fare structure principles, discounts and passes, fare of a particular trip, etc.

2.2.1 Google Transit

Google transit is a tool set, which is freely available and well documented. It supports both static and real-time public transit information. Information such as stops, routes, schedule, service alerts, and other details are included.

The following two primary specifications for Google transit are of interest:

GTFS: The General Transit Feed Specification can be used to exchange *static* transit data.

The General Transit Feed Specification (GTFS) defines a common format for public transportation schedules and associated geographic information. GTFS "feeds" allow public transit agencies to publish their transit data and developers to write applications that consume that data in an interoperable way (Anonymous-1, 2012).

GTFS-real time: An extension to GTFS, this specification can be used to exchange *real time* transit data.

GTFS-real time is a feed specification that allows public transportation agencies to provide real time updates about their fleet to application developers. It is an extension to GTFS (General Transit Feed Specification), an open data format for public transportation schedules and associated geographic information. GTFS-real time was designed around ease of implementation, good GTFS interoperability, and a focus on passenger information[1]. More detailed documentation about GTFS is included in Section 4.2.1 and Annex I.

[1] Google Transit Developers' Website: https://developers.google.com/transit/gtfs-realtime/

2.2.2 HAFAS

HAFAS is the name of a journey planner software for short- and long-distance travel as well as air traffic. It was developed by HaCon[2] (Hannover Consulting).

HAFAS is used extensively by Deutsche Bahn AG (DB). It is the most widespread journey planner software in the world. For the purpose of interoperability HAFAS is used to be able to transfer the PocketNavigator (a mobile journey planner) to as many cities as possible[3].

The HAFAS Public Access Gateway is implemented as a Representational State Transfer (ReST) interface. The following methods are provided: Location, Trip, DepartureBoard, ArrivalBoard, JouneyDetail and Geometry. The first four methods can be called directly. For the JourneyDetail method a reference from the Trip-method or the DepartureBoard method is needed. A result from the JourneyDetail or Trip method provides the necessary reference needed for the Geometry method.

Requests to the system are answered in the the eXtensible Markup Language (XML) or JavaScript Object Notation (JSON). XML responses can be converted to JSON following some simple rules explained in the annex. They are called by a given service URL and multiple GET parameters to specify the desired information.

Some general principles such as coordinates, date and time formats, route indexes, stateless service versus data dependency can also be found in the annex.

The Location Service is used to search for a certain location with a name. It can also be used to retrieve a list of all available stops (very time consuming) in the journey planner and a list of the nearest stops to a given coordinate.

The calculation of a trip from origin to destination is done by the Trip Service. Several parameters can be used to specify the desired trip (such as means of transportation, maximum walking distance, walking speed…).

The DepartureBoard Service and ArrivalBoard Service deliver a list of outgoing / incoming services at a specified journey stop. These requests can also be specified by parameters.

Complete information about a journey can be requested from the JourneyDetail Service.

The Geometry Service returns a polyline that can be used to display the course of a journey on a map.

2.3 Traffic Data

Traffic data refer to real time data that provide information about current traffic conditions in urban areas. Additionally, traffic data provides information on potentially occurred accidents, roadworks, and traffic jams. The transmitted data has to be provided in a standardised way in order to be compatible with the receivers available in the market.

[2] http://www.hacon.de

[3] The HAFAS Journey Planner, http://www.hacon.de/hafas-en

One of the most common ways of transmitting and delivering traffic data is through the Traffic Message Channel (TMC). The Traffic Message Channel is a specific application of Frequency Modulation (FM) Radio Data System (RDS) used for broadcasting real time traffic and weather information. Data messages are received silently and decoded by a TMC-equipped navigation system, and delivered to the driver, typically by offering dynamic route guidance - alerting the driver of a problem on the planned route and calculating an alternative route to avoid the incident[4].

The service provider sends the coded messages to the appropriate FM radio broadcaster for transmission as an RDS signal within normal FM radio transmissions.

The message is coded according to the Alert C standard. It contains a list of up to 2048 event phrases (1402 as of 01.02.2007) which can be translated by a TMC receiver into the language of the user. Some phrases describe individual situations such as a crash, while others cover combinations of events such as construction causing long delays. The TMC data are received by the vehicle radio and antenna, and decoded by a TMC decoder (Werner, 1995).

This reconstructs the original message, using a database of event and location codes, which is presented to the driver as a visual or spoken message. Standard TMC user messages provide five basic items of broadcast information: event description, location, direction and extent, expected duration, diversion advice.

Sources of traffic information typically include police, traffic control centers, camera systems, traffic speed detectors, floating car data, winter driving reports, roadwork reports and others.

Given so few bits to work with, one major design challenge of RDS-TMC was to find a way of using 16 bits (about 65000 locations) to describe an entire state or country. Such a system could not convey latitude-longitude data (so easily available 25 years later with great precision using GPS). Instead, RDS-TMC has to rely on the use of relatively lean location tables, pointing only to significant road junctions along defined national and regional highways. The precision of each traffic event's location is low compared to modern GPS devices. The user's navigation system locates a driver to about 3 metres, but only knows that the crash is between Exit 3 and Exit 4, northbound on the motorway. This limitation is because traffic events (accidents, congestion, burst water mains, faulty traffic lights, etc.) have to be superimposed onto the maps in users' GPS devices by matching the reported location into the location table. If the nearest location table point (nearest significant junction) is located some distance from the point of the crash, then the traffic event might be shown on the device as being on a section of main road between two junctions instead of its real location. Even if the lack of accuracy causes an error of only a short distance, this can make a significant difference as to how the program in the GPS device will interpret the event in relation to the user's planned route. For example, if there is roadwork close to a junction, the position of the roadwork may be interpreted as being a short distance away from its true location. This could place the apparent roadwork on the other side of the junction. The consequence would be that the GPS device might not divert the user from the roadwork because it was assumed to not affect the route[5].

[4] The Traffic Message Channel (TMC) by TISA (Traveller Information Services Association), http://www.tisa.org/technologies/tmc/

[5] Wikipedia: http://en.wikipedia.org/wiki/Traffic_Message_Channel

Another weakness of TMC worth mentioning is that it does not include expected delays, making accurate rerouting impossible.

In order to overcome the previous shortcomings of TMC, nowadays more advanced and accurate proprietary traffic data solutions have been developed. Just to name a few: INRIX Traffic, Navteq Traffic, “3D Traffic”, Google Navigation, TomTom HD Traffic. According to a recent study carried out in Berlin (Sohr and Wagner, 2011) by the Institute of Transport Systems of the German Aerospace Centre, TomTom HD Traffic system outperformed the competition on traffic data systems.

2.4 User Location Data

User Location Data belongs to the broader category of geographical location (aka ”Geolocation”) data supported by most GIS (Geographic Information Systems). By user location data we mean data which includes information that determines the current user’s location with respect to a reference coordinates system (e.g. as a pair of latitude and longitude coordinates). The most well known representative system for providing user location data is the GPS that allows the determination of one’s location when there is an unobstructed line of sight to four or more GPS satellites. Many end-user devices, including personal navigation devices (PND), as well as mobile phones (“smartphones”) are equipped with a GPS sensor. In addition to this, mobile technology has used for advancing the accuracy of current GPS systems. Such an example is the Assisted-GPS (A-GPS) that combines information provided by the mobile network with GPS information for reducing the response time and increase accuracy especially in poor GPS signal conditions, e.g. in cities. Other location services include WiFi Positioning System and cell-site triangulation which can be used in combination, resulting in a hybrid positioning system.

User location information should be provided within eCOMPASS in order for the navigation applications to be aware of the user’s location and be able to calculate the optimal route in a wide range of conditions and circumstances and using different transportation modalities. A variety of use cases require user location data. For instance, User Location Data is necessary in order to:

- Find points of interest in the user's area
- Annotating content with location information
- Show the user's position on a map
- Support turn-by-turn route navigation
- Produce alerts when points of interest are in the user's vicinity
- Provide updated traffic information,

and so on and so forth.

Access to user location data is enabled in different ways, depending on the execution environment and the user’s device. For instance, for GPS sensor-equipped devices, such as smartphones, native or programming language-based APIs are available (e.g. Android Location API) in order to enable reading of data from GPS sensors. Also high-level interfaces have been defined to describe how location associated information can be accessed by any device (in a vendor-agnostic way). An example of such a high-level API is the W3C Geolocation API specification (Popescu, 2010).

Common sources of location information in addition to GPS, include location inferred from network signals such as IP address, RFID, WiFi and Bluetooth MAC addresses, and GSM/CDMA cell IDs, as well as user input.

2.5 POI Data

A POI is a specific point location that someone may find useful or interesting. An example is a point on the Athens Google Map, or any other representing the location of the Parthenon, or a point on the route map representing the location of a gas station[6]. A POI has the following attributes:

1. A name
2. A current Location
3. A category and/or type
4. A unique identifier
5. A URI
6. An address
7. Contact information

Some of the POI attributes are listed below:

- A POI is loosely coupled with the location, i.e. a POI can change location. When this occurs, the loose coupling with the previous location is removed and, providing the POI continues to exist, it is then coupled with its new location.
- A POI has temporal boundaries, i.e. it starts when the human activity at the POI's location commences and ends when human activity ceases, such as when a company or organisation goes out of business.

The term is widely used in cartography, especially in electronic variants including GIS, and GPS navigation software. In this context the synonym waypoint is common.

A GPS point of interest specifies, at minimum, the latitude and longitude of the POI, assuming a certain map datum. A name or description for the POI is usually included, and other information such as altitude or a telephone number may also be attached. GPS applications typically use icons to represent different categories of POI on a map graphically.

Moreover, websites exist which specialize in the collection, verification, management, and distribution of POI, which end-users can load onto their devices to replace or supplement the existing POI. While some of these websites are generic, and will collect and categorize POI for any interest, others are more specialised in a particular category (such as speed cameras) or GPS device (e.g. TomTom/Garmin). End-users also have the ability to create their own custom collections.

Commercial POI collections, especially those that ship with digital maps, or that are sold on a subscription basis are usually protected by copyright. However there are also many websites from which royalty-free POI collections can be obtained.

The applications for POI are extensive. As GPS-enabled devices, as well as software applications that use digital maps become more available, the applications for POI are also expanding. Newer digital cameras for example can automatically tag a photograph with the GPS location where a picture was taken using Exif; these pictures can then be overlaid as POI on a digital map or satellite image such as Google Earth. Geocaching applications are

[6] W3C Points of Interest Working Group Wiki: http://www.w3.org/2010/POI/wiki/Main_Page

built around POI collections. In vehicle tracking systems POIs are used to mark destination points and/or offices that can be also used for monitoring the relative position of vehicles.

Many different file formats, including proprietary formats, are used to store point of interest data, even when the same underlying WGS84 system is used. Reasons for variations to store the same data include:

- A lack of standards in this area (GPX is a notable attempt to address this).
- Attempts by some software vendors to protect their data through obfuscation.
- Licensing issues that prevent companies from using competitors file specifications.
- Memory saving, for example, by converting floating point latitude and longitude co-ordinates into smaller integer values.
- Speed and battery life (operations using integer latitude and longitude values are less CPU-intensive than those that use floating point values).
- Requirements to add custom fields to the data.
- Use of older reference systems that predate GPS (for example UTM or the British national grid reference system)

The following are some of the file formats used by different vendors and devices to exchange POI (and in some cases, also navigation tracks):

- ASCII Text (.asc .txt .csv)
- Topografix GPX (.gpx)
- Garmin Mapsource (.gdb)
- Google Earth Keyhole Markup Language (.kml .kmz)
- Pocket Street Pushpins (.psp)
- Maptech Marks (.msf)
- Maptech Waypoint (.mxf)
- Microsoft MapPoint Pushpin (.csv)
- OziExplorer (.wpt)
- TomTom Overlay (.ov2)
- OpenStreetMap data (.osm)

Third party and vendor-supplied utilities are available to convert point of interest data[7] between different formats to allow them to be exchanged between otherwise incompatible GPS devices or systems. Furthermore, many applications will support the generic ASCII text file format, although this format is more prone to error due to its loose structure as well as the many ways in which GPS co-ordinates can be represented (e.g. decimal vs. degree/minute/second)[8].

2.6 Fleet Management Data

For the part of fleet management data, we have to differentiate between planning data (plan trips) and execution data (monitor trip execution).

For the planning part it is common to have a planning system that processes transport orders coming from, e.g. a pre-system. The planning system executes the planning algorithms and generates tours. These tours consist of the transport order stops locations. The planning system interfaces with a 'Transfer Database' (TDB) and stores results to it.

Further actions, be it on the planning part or, later on, during the tour execution, are communicated via the 'Transfer Database' (TDB).

[7] An online list of POI Converters, http://www.gps-data-team.com/convert.php

[8] Wikipedia: http://en.wikipedia.org/wiki/Point_of_interest

A TDB consists of transfer objects (import and export objects), which are modeled in the transfer database architecture. These objects are interpreted by every system involved in the planning and in the execution, e.g. planning system, vehicle client, web service (WS). A defined handshake procedure forms the underlying basis of the physical process. Flags (time stamps and action codes) are used to depict the logical data exchange.

Example: a host planning system exports planned tours to the database. For the tour execution, a mobile client later on imports trips exported from TDB via WS. During the tour execution, the mobile client reports events via WS to the TDB. The update in the TDB triggers the planning system to update its data. The planning system receives the TDB update and processes the data. This enables the planning system user to monitor the event.

The transfer DB is often integrated into the application data model of planning applications, but can also be implemented into other instances or to other DB servers.

2.6.1 Data Export from Transfer DB

Every **export object** consists of an entry in the header table **EXP_HEADER** and, depending on the object type, further entries in one or more detail tables with the same functions as for imports.

Depending on the export object, the export process is set up as a permanent process which responds either

- once a particular status of a business object is reached

or

- a user explicitly initiates an action

Please see Annex **3** for more detailed documentation.

2.7 Environmental Data

The estimation of the cars CO_2 emissions can be approximately calculated by counting the distance that the car covered during the travelled route. Thus in this way we have the first parameter in order to make an estimation of the total CO_2 emissions during the journey. Another parameter that has to be taken into account is the prediction-estimation of the traffic congestion that might occur during the car journey. Also another important factor is the type of the road that the car is travelling e.g. rural, motorway, highway and/or city road. It is obvious that the fuel consumption and consequently the car emissions are different depending on the type of roads on which the car travelling. And finally depending on the type of car, average fuel consumption (measured in litres of fuel per kilometre) differs. The amount of CO_2 emission also depends on the type of fuel such as petrol, gas or diesel. As no common data storage or representation format has been developed specifically for exchange of data about CO_2 emissions, in most cases data are stored in proprietary databases or become available in various formats. Examples include: plain text, comma separated values and Excel spreadsheets.

In addition to CO_2 other environmental data include atmospheric and meteorological (i.e., weather) data. In this context a number of scientific data standards exist for storing and transmitting oceanographic and atmospheric data. These include:

- **GRIB** (General Regularly-distributed Information in Binary Form) is a World Meteorological Organisation (WMO) standard designed for transmitting and archiving large volumes of two-dimensional meteorological and oceanographic data.

It is the standard used by the world's largest operational meteorological centers (NMC and ECMWF). A new GRIB format, known as GRIB2, was declared operational in November 2001. GRIB2 is more flexible than the original GRIB. It can be used to handle radar and satellite data and allows for better data compression.
- **CDF** (Common Data Format) was initially developed by NASA Goddard about 1980 as an interface and a toolkit for archival and access to multidimensional data. Over the years it has evolved into a machine-independent standard and is often used by different NASA groups for storing space and earth science data.
- **netCDF** (network CDF) is a commonly used data format used for scientific data. It was developed by Unidata (UCAR) around 1986, using CDF as a starting point. netCDF emphasizes a single common interface to data, implemented on top of an architecture independent representation. netCDF's data model is a relatively simple multidimensional array structure. Access to data in netCDF format is enabled by user written software or by many public domain and commercial tools.
- **HDF** (Hierarchical Data Format) is a general, extensible scientific data exchange format created by NCSA. There are two main versions: HDF4 and HDF5 (introduced in 1999). HDF4 is incompatible with HDF5. HDF emphasizes a single common format for the data, on which many interfaces can be built. HDF is often used to archive and transmit raster images.
- **BUFR** (Binary Universal Format Representation) is a WMO standard data format for the representation of meteorological and oceanographic data. Although it could be used for any type of data, its primary function is to represent observational data (e.g., from stations, raobs and ships). It was designed to reduce redundancy for efficient transmission over the GTS and to reduce the computer time required to decode the information.

The aforementioned data formats are basically binary. On the other hand only a few standardisation efforts have been carried out about data representations using text-based and metadata-enabling formats, such as XML. A first effort for the creation of an XML-based representation of environmental data resulted in the development of the so-called *Environment-XML* which is now deprecated in favour of its descendant called **EEML** (Extended Environments Markup Language[9]), a protocol for sharing sensor data between remote responsive environments, both physical and virtual. It can be used to facilitate many-to-many connections as implemented by a web service called Pachube[10], which enables people to tag and share real time sensor data from objects, devices and spaces around the world. The EEML XML-based protocol supports data stream sources that respond to and exchange data with facilities, such as buildings and devices, as well as events through data stream tagging. EEML is designed primarily for the construction industry, however it has the potential to be used for exchanging data from devices and sensors (such as CO_2 emissions sensors).

Another effort towards standardisation of environmental data is the Climate Science Modelling Language (CSML[11]) (current version: 3.0), originally developed as part of the NERC DataGrid Project. CSML is a data model for encoding climate, atmospheric and oceanographic data in terms of geometry-based observation classes such as Points, Profiles, Trajectories and Grids. In addition to static observation it also supports time-series observation data. Version 3.0 is based on the ISO 19156 Observations and Measurements

9 http://www.eeml.org
10 https://pachube.com
11 http://csml.badc.rl.ac.uk

standard through its accompanying implementation as a GML 3.2.1 Application Schema. CSML development is currently supported by the British Atmospheric Data Centre on a best efforts basis. As far as CO_2 emission data are concerned these are mostly stored in large databases, such as the PANGAEA database[12] maintained by World Data Centre for Marine Environmental Sciences (WDC-MARE[13]) mainly used for storing marine CO_2 (Surface water CO_2) in collaboration with the Carbon Dioxide Information Analysis Center (CDIAC)[14].

2.8 Conclusions

In this section six categories of external data resources were identified and described in sufficient detail. These are: Public Transportation, Traffic, User Location, Points of Interest, Feet Management and Environmental Data. For some of these the need to access data resources via appropriate APIs is obvious. In some other cases, such as user location data there is not such a need, as data will be directly available from the device (i.e. GPS sensor) to the application level via native API calls.

The freely available and well documented Google Transit fits well the need for a exchanging both static and real time public transit information. Another interesting system is HAFAS extensively used by Deutsche Bahn AG, which supports interfacing via ReSTful web services, XML and JSON. Regarding transmission of traffic data one of the commonest ways has been TMC for many years. However, nowadays TMC is insufficient for eCOMPASS purposes, as more advanced proprietary solutions exist, such as TomTom HD Traffic, INRIX Traffic, Navteq Traffic and Google Navigation. User Location data are provided by any PND or smartphone device through the embedded GPS sensor or assisted by mobile or network technologies, such as A-GPS, WiFi Positioning System, etc. Due to the lack of standards for the representation of POIs a variety of representation and storage file formats exist, ranging from plain ASCII text to more proprietary formats such as Google Earth Keyhole Markup Language (.kml .kmz) or TomTom Overlay (.ov2). Regarding Fleet management, data exchange is performed by means of a 'Transfer Database' and a defined handshake protocol for reading and writing data. The most representative standards for describing environmental data are limited to the representation of oceanographic and atmospheric data in binary data formats. Furthermore a few metadata supporting formats exist for general purpose (e.g. EEML intended for the construction industry and CSML for climate, atmospheric and oceanographic data). As far as CO_2 emission data are concerned these are mostly available in various databases and formats maintained by environmental organisations such as WDC-MARE and CDIAC.

[12] http://www.pangaea.de
[13] http://www.wdc-mare.org
[14] http://cdiac.esd.ornl.gov/oceans/home.html

3 Survey of Data Representation and Transmission Standards

The purpose of this section is to provide a survey of the state of the art of standards specified on Data Representation and Transmission for Transportation and Route Planning services, according to the project needs for data types, specified in the previous Section. At the end of this Section a reference is attempted to cover non-standard or proprietary formats that might be used by vendors in the area of transportation services.

3.1 TPEG

3.1.1 Introduction

The Transport Protocol Experts Group (TPEG) was founded in 1997 by the European Broad Casting Union (EBU). It was a group of experts led by the EBU coming from all areas of the Traffic and Travel Information businesses, as well as broadcasting. The group developed the TPEG specifications for transmission of language independent multi-modal Traffic and Travel Information. Validation of the initial work was undertaken by the TPEG Project which had a three year duration, partially funded by the European Commission.

The TPEG work was partially based on the work done with RDS-TMC, but TPEG data is human understandable as well as machine readable. In particular TPEG does not assume any large scale location or pre-coded phrase databases in any client receiving device. Its standards were then used as a starting point for the mobile.info project in Germany to develop an enhanced version of TPEG that is optimised for dynamic route guidance.

In 2008 the Traveler Information Services Association[15] (TISA) was established as a not-for-profit company (ASBL under Belgian law) to ensure an international framework for market-driven, coordinated, proactive implementation of traffic and travel information services and products based on existing standards such as RDS-TMC and TPEG. It also works towards the development of future standards and services. TISA has taken over all the activities undertaken by the previous TMC Forum, TPEG Forum and the German Mobile.Info project. It also supports standards that provide elements or a framework for services and products covering traffic and travel information, including roads, public transport, and related information needs such as points of interest, weather, and environmental information.

In the early days of the TPEG development, applications were designed in two halves. On the one hand, there were coding experts, and on the other designers. At that time the primary focus was on binary delivery using DAB. With XML becoming more main stream it was very quickly adopted by the TPEG developers who unified the two separate approaches. At the same time it was clear that XML provided another channel to deliver Traffic and Travel Information. More recently TPEG developers have turned to UML to hold the conceptual content and to build new applications. Current work programmes include building UML extracts that will automatically generate the XML and binary specifications.

3.1.2 State of the Art

Figure 2 shows the TPEG world as a whole. Currently the TPEG standards and specifications are classified in TPEG-1 (the 18234 series) and TPEG-2 (the 21219 series), respectively. The dark green boxes refer to already standardised TPEG applications and services (RTM, PTI, SNI, LOC, LRC, SSF,INV), while the light green boxes are already specified and currently following the ISO's approval procedure

[15] http://www.tisa.org

Besides the applications and services TISA has specified a set of profiles. Among the latter the TAP (TPEG Automotive Profile) is probably the most relevant. A profile defines specific selections out of possible application and parameter combinations within the TPEG base standards. TPEG Automotive Profile will define a distinct choice which applications and what transmission settings are to be used to provide a traffic information service to car navigation systems using DAB. Another interesting profile is the TPSMP (TPEG Public Service & Media Profile). The TPSMP will use TPEG-RTM for road traffic messages and TPEG-PTI for public transport information.

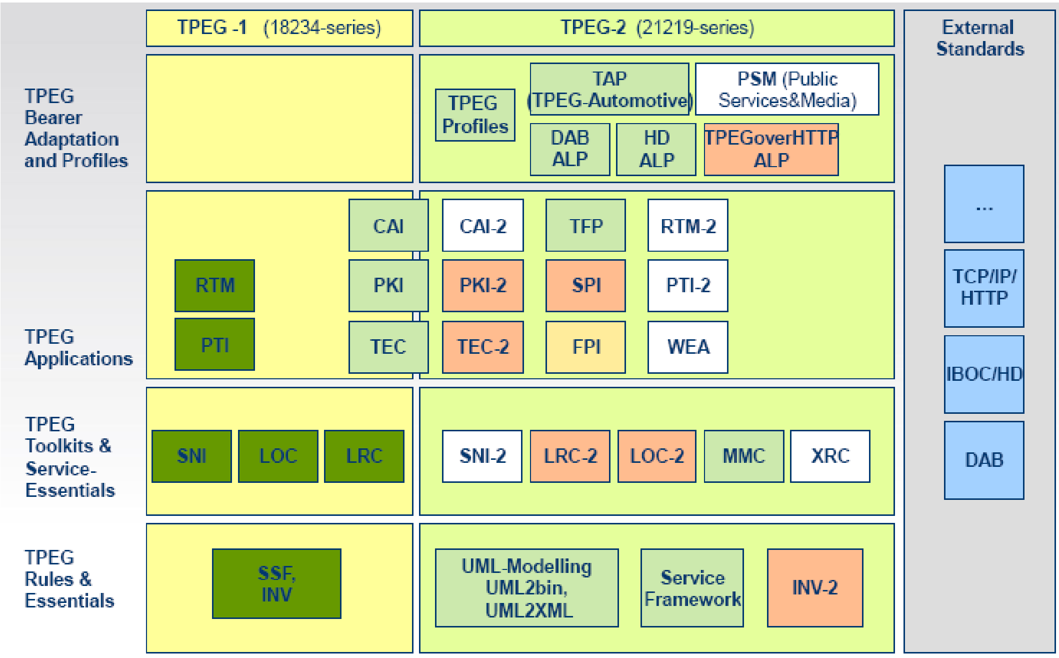

Figure 2: TPEG State of the Art

The development of TPEG has gained significant momentum over the last two years as commercial service providers in France, Germany, Italy and the UK begin to launch trials and plan next generation traffic services. The commercial traffic information sector in Europe is showing a clear commitment to launching TPEG services for the automotive industry. In addition to the commercial TPEG services there are also a number of public providers considering TPEG in Europe, particularly in Germany, Switzerland and the UK. These public services are unlikely to offer the same applications or use the same location referencing techniques as the automotive-focused commercial services. The next paragraphs provide an overview of commercial TPEG services available across Europe.

3.1.3 Technical Specification

TPEG has undergone a number of changes since it was conceived by the TPEG Forum. However, the fundamental construction of all TPEG messages is still made up of three building blocks that identify the message, describe the event and locate the incident, as shown in Figure 3 below.

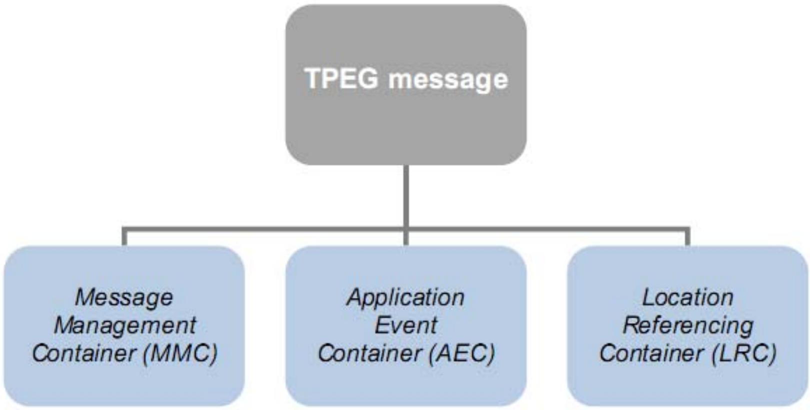

Figure 3: Fundamental TPEG Message Structure

Every TPEG message includes a Message Management Container (MMC) that provides information relating to its origins and features. The MMC is intended to allow devices to quickly determine the relevance and lifetime of the content without having to analyse the entire TPEG message. TPEG enables a range of applications to be delivered by using an Application Event Container (AEC). There are currently only three applications that have been standardised by ISO to date but there are already plans for another thirteen to be standardised in the future (see Figure 4 below).

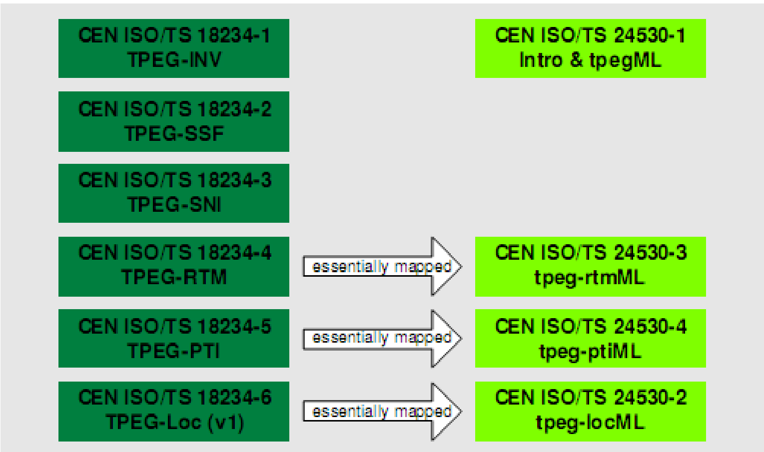

Figure 4: TPEG Standardisation

The need for TPEG RTM and TPEG TEC

Road Traffic Management (RTM) was the first application to be developed by the TPEG Forum but the automotive industry did not consider it to be suitable for route guidance purposes. A consortium of German vehicle manufacturers and suppliers therefore developed the traffic Event Compact (TEC) application (through the mobile.info project) to address their navigation-specific needs. The TPEG TEC specification is currently at the last stage of the ISO proposal phase.

The differences between TPEG RTM and TPEG TEC

RTM and TEC both use look-up tables, as the ones shown in Figure 5 to describe the parameters of road traffic incidents (see Figure below). This allows the message contents to remain language independent by having different look-up tables for each language.

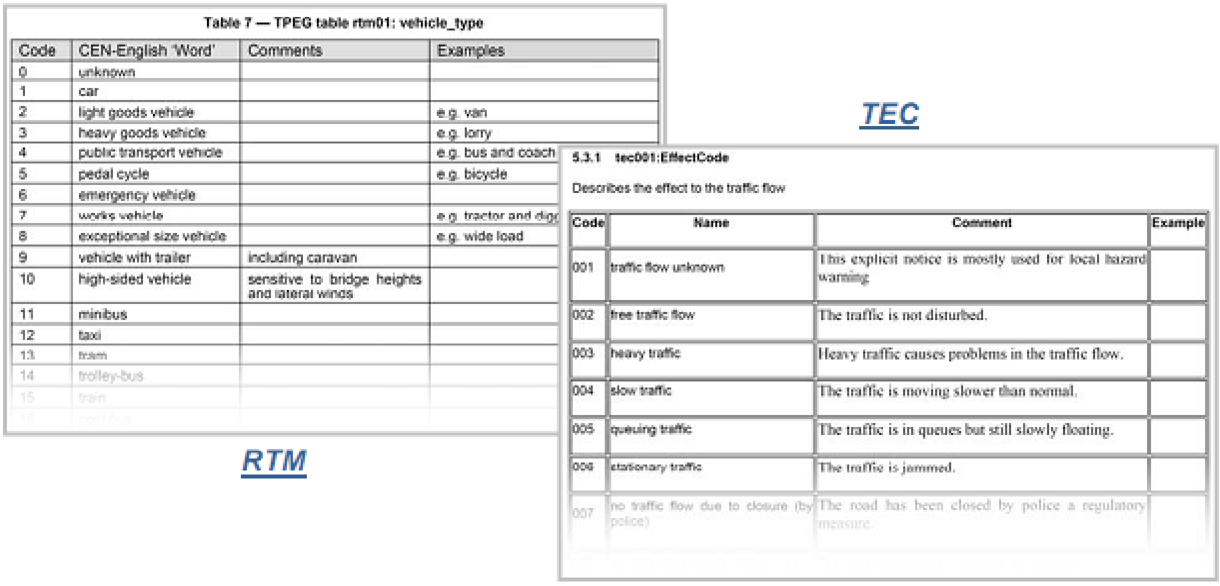

Table 7 — TPEG table rtm01: vehicle_type

Code	CEN-English 'Word'	Comments	Examples
0	unknown		
1	car		
2	light goods vehicle		e.g. van
3	heavy goods vehicle		e.g. lorry
4	public transport vehicle		e.g. bus and coach
5	pedal cycle		e.g. bicycle
6	emergency vehicle		
7	works vehicle		e.g. tractor and dig
8	exceptional size vehicle		e.g. wide load
9	vehicle with trailer	including caravan	
10	high-sided vehicle	sensitive to bridge heights and lateral winds	
11	minibus		
12	taxi		
13	tram		
14	trolley-bus		
15	train		

RTM

TEC

5.3.1 tec001:EffectCode

Describes the effect to the traffic flow

Code	Name	Comment	Example
001	traffic flow unknown	This explicit notice is mostly used for local hazard warning	
002	free traffic flow	The traffic is not disturbed.	
003	heavy traffic	Heavy traffic causes problems in the traffic flow.	
004	slow traffic	The traffic is moving slower than normal.	
005	queuing traffic	The traffic is in queues but still slowly floating.	
006	stationary traffic	The traffic is jammed.	
007	no traffic flow due to closure (by police)	The road has been closed by police a regulatory measure.	

Figure 5: Lookup Tables for TPEG TEC and TPEG RTM

Although RTM and TEC both use look-up tables to describe traffic incidents, the way in which the descriptions of the incident are structured varies significantly. These differences are summarised below and also illustrated in Figure 6.

The TPEG-RTM application is designed to deliver road information to any kind of end-user client (not just navigation systems). It is based on RDS-TMC event codes yet it provides considerably more detail and richness for a service provider to deploy. The information that RTM provides relates to event and status information on the road network and on associated infrastructure affecting a road journey. RTM does not, however, provide any specific structure to this content and there is no mandatory content. This situation could result in wide variations in the contents of messages created by different service providers across Europe. The TPEG-TEC application is designed to deliver road information specifically to navigation systems for dynamic route guidance purposes. Unlike RTM, TEC has a minimum level of mandatory information relating to the effect that an incident has on traffic flow. TEC also has a well-structured format for providing cause information using a linked set of encoding rules. This application is optimised for navigation systems because the content is delivered in a consistent and concise manner. The adoption of a single TPEG application for road traffic information across all services in Europe would be the best solution for vehicle manufacturers because it would provide a uniform structure for decoding messages. There are a number of barriers, however, that could prevent public service providers from adopting TPEG TEC. The most important obstacle is that public service providers need to cater to a much wider range of applications. While RTM is much broader and flexible this can be a drawback if applications need a concise information (that is why it was necessary to design TEC for the routing related Use case of Navigation). As ITS applications most likely need a routable interpretation, TEC is a serious candidate for applications in a cooperative system environment.

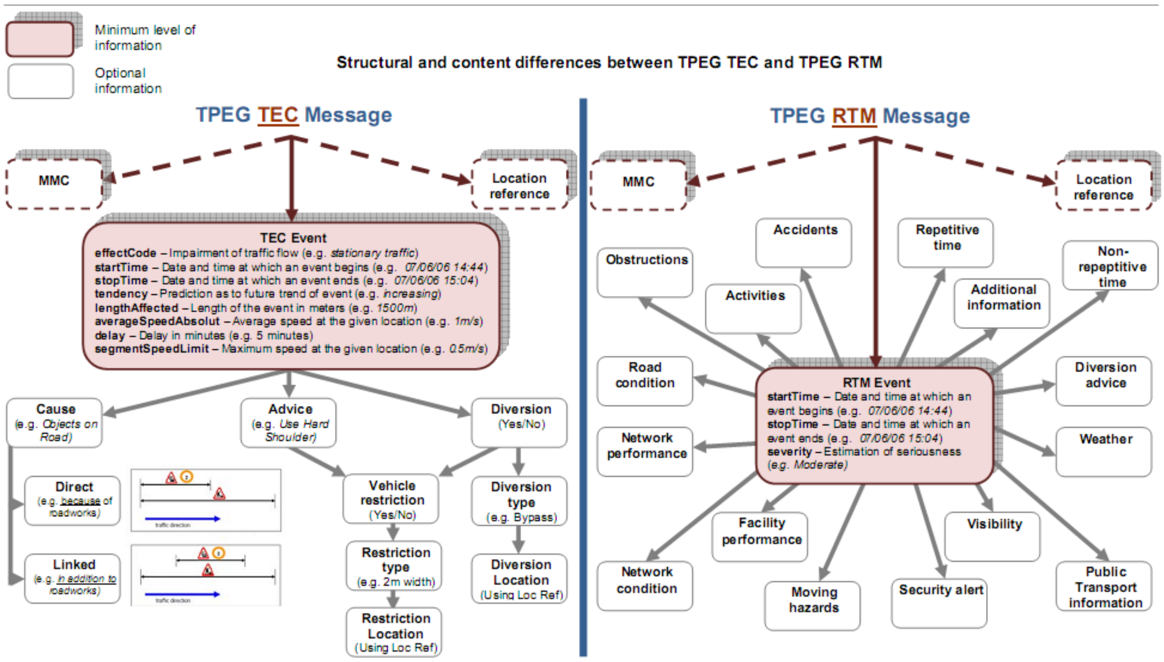

Figure 6: Structural and Content Differences between TPEG TEC and TPEG RTM

3.2 Transmodel

Transmodel is the European Reference Data Model for Public Transport; it provides an abstract model of common public transport concepts and structures that can be used to build many different kinds of public transport information system, including for timetabling, fares, operational management, real time data, etc. (CEN TC278, Reference Data Model For Public Transport, ENV1289)[16].

Transmodel which is a pure European initiative, undertaken by CEN (European Committee for Standardisation), has been proposed in order to deal with the different systematic information models and the management of distributed data sets of many different types, which underpin the development of transport information systems.

Transmodel has articulated a comprehensive conceptual model for public transport information systems, considering not just the public facing data that is the main focus of GTFS, but also the other back office and operational systems needed to manage, produce and update both reference and real time data, so that end-to-end electronic systems can be developed. Transmodel has been used to underpin a number of message sets to exchange particular types of data, such as the Service Interface for Real Time Information (for real time transport data).

The Transmodel standardisation program has been running since the early 1990s, and has been able to benefit from the extent and the diversity of European transport networks. Examples of almost every different mode, network topology, constraint, fare model, operational model can be found in Europe, which has both single and multiagency configurations operated by both the public and private sector organisations. Many different systems have been compared to establish a common set of flexible abstractions, systematically documented as the Transmodel Corpus (Knowles & Milles, 2008).

[16] Transmodel Standard, Official Website http://www.transmodel.org

3.3 SIRI (Service Interface for Real time Information)

Public transport services rely increasingly on information systems to ensure reliable, efficient operation and widely accessible, accurate passenger information. These systems are used for a range of specific purposes: setting schedules and timetables, managing vehicle fleets, issuing tickets and receipts, providing real time information on service running, and so on.

The Service Interface for Real Time Information (SIRI[17]) specifies a European interface standard for exchanging information about the planned, current or projected performance of real time public transport operations between different computer systems. It is intended for exchange of information between servers containing real time public transport vehicle or journey time data. These include the control centres of transport operators and information systems that utilise real time vehicle information to operate the system, and the downstream systems that deliver travel information to the public over stop and onboard displays, mobile devices, etc. SIRI is can also be used both for the bulk pipelining of large amounts of data between different computer systems, and for lower traffic ad-hoc queries.

The most important characteristics of SIRI are the following:

- SIRI define its messages using XML. A careful separation is made between **Transport** (how the data is transported) and **Payload** (the domain data exchanged), so that SIRI messages may be exchanged as either XML documents with HTTP-POST or using Simple Object Access Protocol (SOAP) A Web Service Definition Language (WSDL) binding is also defined for the latter.
- The payload model is wrappered in a **Mediation** layer, also described with XML, that both provides common management functions and also formally describes as policies the parameterised aspects of mediation or exchange behaviour or that can be carried out by a service.
- CEN **Transmodel** terminology and relationships are followed in the underlying PT application data model.
- A robust XML schema is available. The schema is encoded as a W3C .xsd schema, and is modularised into a number of reusable sub schemas and type packages. The schema has been validated against mainstream validators and there are working applications using common tools such as JAXB.
- Supports WSDL. Binding the schema is accompanied by a WSDL binding for creating SOAP services.

SIRI uses a consistent set of general communication protocols to exchange information between client and server. The same common patterns of message exchange are used in all the different functional interfaces. Two well-known specific patterns of client server interaction are used: *Request/Response* and *Publish/Subscribe:*

- *Request/Response* allows for the ad hoc exchange of data on demand from the client.
- *Publish/Subscribe* allows for the repeated asynchronous push of notifications and data to distribute events and Situations detected by a real time Service. This can be much more efficient for some types of communication as the client does not need to poll to detect changes to the data; rather the notifying service triggers a data exchange only when it detects an event. The SIRI *Publish/Subscribe* Protocol prescribes particular

[17] http://www.siri.org.uk

mediation to filter the number of messages returned, for example, only creating updates if real time predictions change by more than a certain threshold from a previous value.

The use of the *Publish/Subscribe* pattern of interaction follows that described in the Publish-Subscribe Notification for Web Services (WS-PubSub) specification, and as far as possible, SIRI uses the same separation of concerns and common terminology for publish/subscribe concepts and interfaces as used in WS-PubSub.

For the delivery of data in responses (to both requests and subscriptions), SIRI for efficiency, supports two common patterns of message exchange, as realised in existent national systems:

- A one step '*Direct Delivery*', as per the classic client-server paradigm, and normal WS-PubSub publish subscribe usage and
- A two step '*Fetched Delivery*' which elaborates the delivery of messages into a sequence of successive messages pairs to first notify the client, and then to send the data when the client is ready.

The Interaction patterns and the Delivery patterns are independent aspects of the SIRI protocol and may be used in any combination in different implementations. Care is taken to separate concerns of message transport from those of functional service content. For a given SIRI Functional Service type (Connection Monitoring, Stop Monitoring etc.), the message payload content is the same regardless of whether information is exchanged with a *Request/Response* or *Publish/Subscribe* pattern, or whether it is returned by *Direct* or *Fetched* Delivery.

Several different sources of documentation are available to support the use of SIRI, such online documentation at the SIRI website[16], SIRI specifications (SIRI documentation, prepared as an electronic document to CEN standards) and examples of use with sample XML documents.

SIRI also addressed a growing need to update various national and proprietary standards to conform with modern methodologies and technologies, notably modern data exchange standards (XML), to use standard Public Transport application domain terms and modelling concepts in line with the European TransModel standard, and to build on other relevant CEN, ISO and W3C technical standards for example, for geospatial position, language codes, and so forth (Knowles, 2008).

All SIRI services are provided over a standardised Communications layer (see Figure 7), based on a Web Services Architecture. The Production Timetables service (PT) exchanges information about the expected operation of a transport network for a specified day in the near future. This service is suited for provisioning AVL systems and smart devices with base timetables. The Estimated Timetable service (ET) provides details of the operation of the transport network for a period within the current day, detailing real time deviations from the timetables and control actions affecting the Timetable (cancellations, additional Journeys and Detours). This service is suited for provisioning AVL systems and smart devices with real time timetables.

The Stop Timetable (ST) and Stop Monitoring (SM) services provide **stop-centric information** about current and forthcoming vehicle arrivals and departures at a nominated stop or Monitoring Point, typically for departures within the next 20-60 minutes for display to the public. The SM service is suited in particular for providing departure boards on all forms of device.

The Vehicle Monitoring service (VM) provides information about of the current location and expected activities of a particular vehicle, and can give the current and subsequent journey and the calling points on each journey, together with the scheduled and expected arrival times. The VM service is suited in particular for onboard displays, and visualisation of vehicle movement, and for exchanging information on roaming vehicles between different control systems. It also constitutes a detailed logging feed suitable for collecting historic data about performance against schedule.

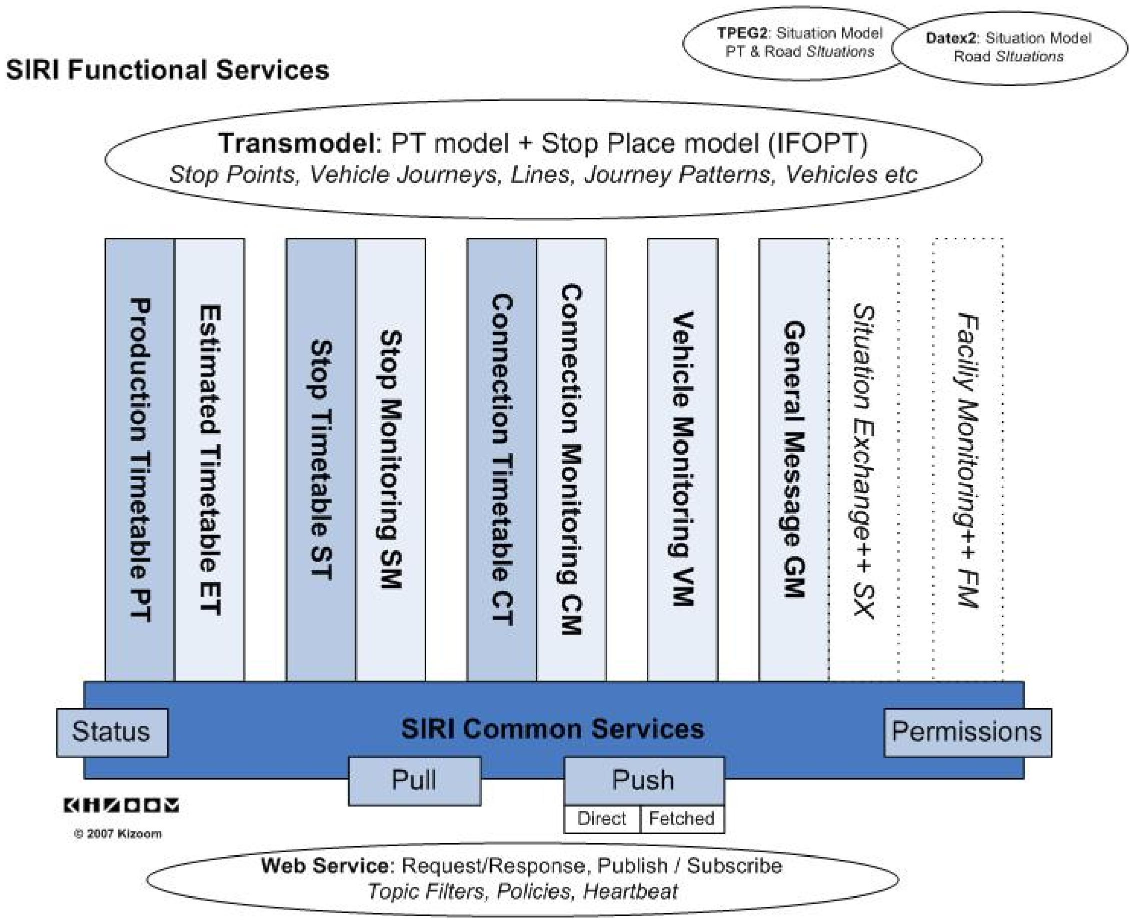

Figure 7 : SIRI Functional Services

The Connection Timetable service (CT) and Connection Monitoring service (CM) allow transport operators to exchange information about the real time management of interchanges between feeder and distributor vehicles arriving and departing at a connection point. The General Message Service (GM) provides a structured way to exchange arbitrary informative messages between participants, such as travel news, operational advice. It can be used to link together incident management systems in a store and forward architecture.

3.4 DATEX II – CEN TS 16157

DATEX II[18] has been developed to provide a standardised way of communicating and exchanging traffic information between traffic centres, service providers, traffic operators and media partners. The specification provides for a harmonised way of exchanging data

[18] DATEX II Official Website: http://www.datex2.eu/

across boundaries, at a system level, to enable better management of the European road network. DATEX II is at this time developed and maintained under the umbrella of the EasyWay project supported by the European Commission. DATEX II data model contains different data categories, such as traffic situations, measured data, calculated data and traffic views.

DATEX II provides web service support for accessing transportation data. In particular, SOAP Web services are supported according to the regular DATEX II profile. In term of delivery modes, both PULL and PUSH services are supported. PULL is used to retrieve all active situations of a certain type e.g. Accidents. For enabling PULL Web Services, Web Service Security features are enabled (Web Service secured with Basic Authentication and a simple HTTP Get). Secure call of available web services enable authenticated access to the following data: Traffic data, Weather, Various metadata. In addition to the PULL protocol, a simple HTTP GET is supported with basic authentication with user name and password.

All content available through DATEX II standard is described in XML schema with extensions. The information which is enabled by the extensions include: user location, road network management, traffic events, weather info (e.g. humidity). An example of an XML file that describes traffic info is illustrated in tree form in the following Figure 8.

Interoperability issues between disparate DATEX II systems are given high priority. A first version (1.0) was produced at the end of 2006 and was then quickly disseminated among countries. The corresponding implementations raised a number of mistakes and requests for Management Overview - White Paper Prepared by CEN TC 278 Working Group 3 Sub Group 7 Version 1.0 change. After more than one year of work, a first proposal for the version 2.0 was issued in July 2009. In parallel, CEN TC278 circulated the first standard parts for comments. The comment resolution as well as the integration of the work carried out in TG involved producing a new proposal called "release candidate 2" or "RC2".

3.5 IFOPT (Identification of Fixed Objects in Public Transport)

IFOPT (*Identification of Fixed Objects in Public Transport*) is a prCEN Technical Specification that provides a Reference Data Model for describing the main fixed objects required for public access to Public transport, that is to say Transportation hubs such as airports, stations, bus stops, ports, and other destination places and points of interest, as well as their entrances, platforms, concourses, internal spaces, equipment, facilities, accessibility etc. Such a model is a fundamental component of the modern Public transport information systems needed both to operate Public transport and to inform passengers about services.

IFOPT is itself built upon the CEN Transmodel standard and defines four related sub models (see Figure 9).

- *Stop Place Model:* Describes the detailed structure of a Stop Place (that is stations, airports, ferry ports, bus stops, coach stations, etc., providing a point of access to public transport) including Entrances, pathways, and accessibility limitations.
- *Point of Interest Model:* Describes the structure of a point of interest (that is tourist attractions, leisure facilities, stadia, public buildings, parks, prisons, etc.) to which people may wish to travel by public transport) including physical points of access, i.e. Entrances.
- *Gazetteer Topographical Model:* Provides a topographical representation of the settlements (cities, towns, villages etc.) between which people travel. It is used to associate Stop and Station elements with the appropriate topographic names and concepts to support the functions of journey planning, stop finding, etc.

- *Administrative Model*. Provides an organisational model for assigning responsibility to create and maintain data as a collaborative process involving distributed stakeholders. Includes namespace management to manage the decentralised issuing of unique identifiers.

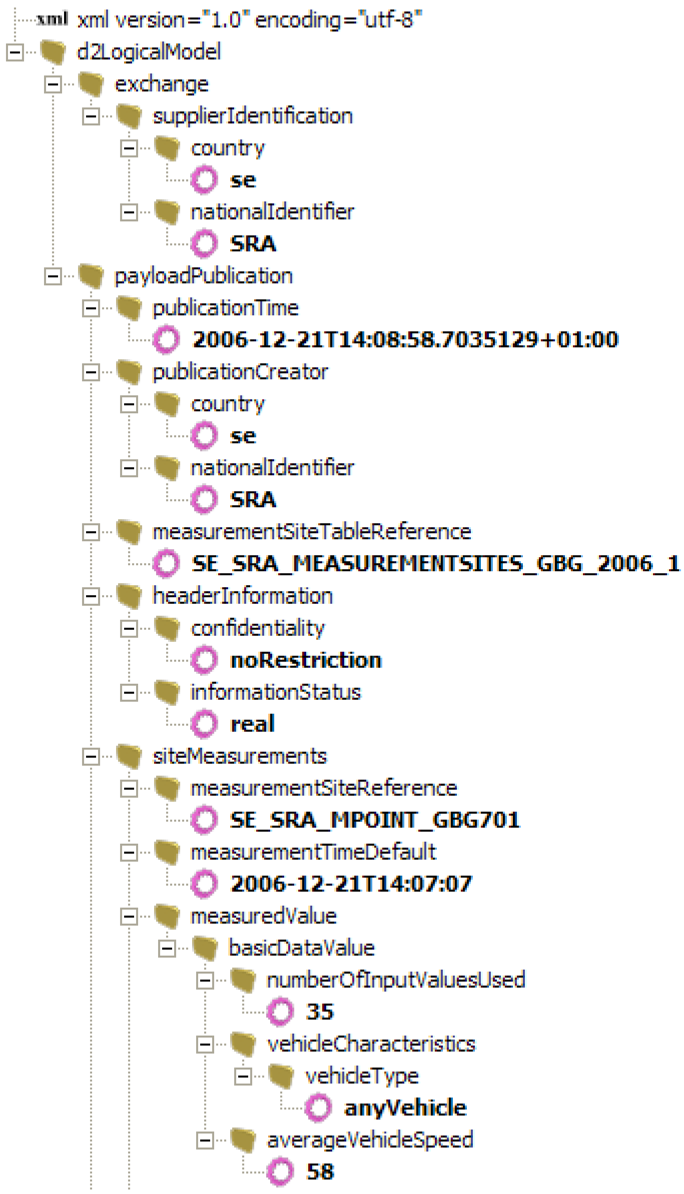

Figure 8 : An example of an XML file for measuring traffic data based on DATEX II schema

This partitioning of Fixed Object into distinct sub-models is in particular of significance for data exchange. For data exchange, a model held on one computer system must typically be serialised into an XML document or other flat file format and then, after transmission, be de-serialised and referenced back into another model on a different system. In order to exchange data efficiently it must be possible to partition the data of a large model (for example all the bus stops in a country) into smaller coherent subsets (for example all the bus stops in a single area within a country) that include references to objects that are not included in the export (for example stops in adjacent areas, or the full definitions of the areas). This raises considerations for ensuring integrity of reference and in particular for the management of the identifiers that are used to implement the reference across different systems.

In practice the coherent subsets of data that are needed for efficient exchange must reflect the operational processes and frequency of change of the data. The four Fixed Object sub-models represent four primary sets of data that usually will be exchanged as distinct documents between different parties on different timescales. Thus for example, the Administrative model is a small model that typically needs to be set up centrally with a view to coordinating the work of different stakeholders: once created, its data will change only occasionally, but it will be extensively referenced by other documents. At the other extreme, the Point of Interest and Stop Place models will need to be managed as discrete large data sets for each locality, each requiring detailed geographical surveying and local access knowledge for its creation and maintenance.

A second reason for modularisation is that it allows a more flexible and incremental approach to adoption of the standards (Anonymous-2, 2012).

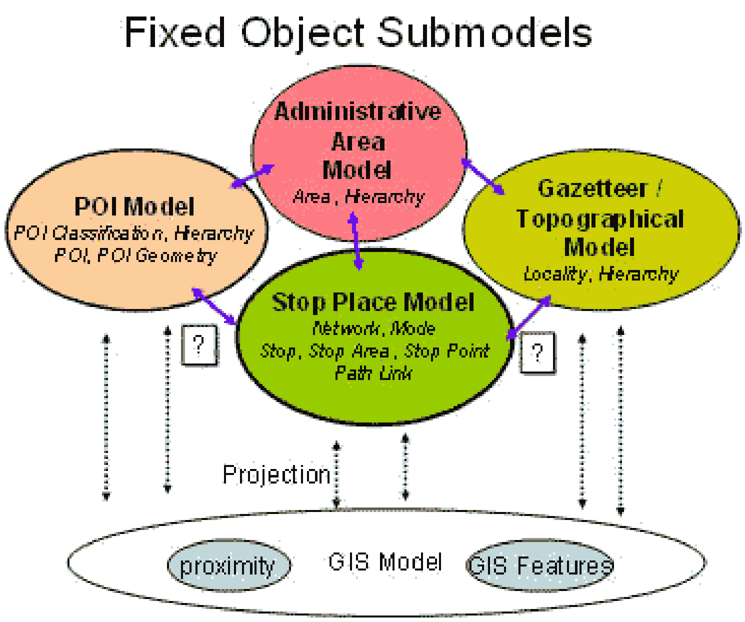

Figure 9 : IFOPT submodels

3.6 NeTEx (Network Exchange) CEN TC 278 WG9

NeTEx is a prCEN/ Technical Standard currently in development[19]. The goal of NeTEx is to provide an efficient European wide standard for exchanging Public Transport schedules and related data. NeTEX is intended to be a general purpose format capable of exchanging timetables for Rail, Bus, Coach, Ferry, Air or any other mode of public transport. It includes full support for rail services and can be used to exchange UIC (International Union of Railways) data

NeTEx is based on the CEN Transmodel standard which specifies a conceptual model for Public Transport data, extended with additional concepts for stops and stations from the CEN Technical standard IFOPT (Identification of Fixed Objects in Public Transport).

[19] NeTEx Official Website: http://www.netex.org.uk

NeTEx uses a fully articulated model that represents PT concepts as well characterised, layered abstractions; the format is designed for the efficient, updateable exchange of complex transport data between distributed systems. This allows the data to be used in modern web services architectures and to support a wide range of passenger information and operational applications. The NeTEx schema is free to use and its development is managed by the CEN standards process.

NeTEx provides a means to exchange stops, routes and timetables between different computer systems, together with related operational data.

NeTEx comprises the following main components:

1. A Specification.
2. A modular NeTEx XML Schema.
3. Descriptions of protocols for exchanging documents created with the schema.
4. Documentation on the NeTEx schema and the processes to use it.
5. Examples of schedules encoded as NeTEx XML documents.

NeTEx defines a standard for exchanging Timetable related data for public transport services, including:

- **Network Topologies** (e.g. routes, journey patterns)
- **Timetables** (including journey patterns, vehicle journeys, operating days)
- Data to support **real time** operations
- Basic **Fare** data

XML documents based on the NeTEx schema can thus be used to exchange the following information:

- Public Transport **schedules** including stops, routes, departures times / frequencies, operational notes, and map coordinates.
- Routes may have complex topologies such as circular routes, cloverleaf and lollipops, and complex workings such as short working and express patterns. Connections with other services can also be described.
- The **days** on which the services run, including availability on **public holidays** and other **exceptions**.
- Journeys may include composite journeys such as train journeys that merge or split trains.
- Information about the **Operators** providing the service.
- Additional operational information, including, **positioning runs**, **garages**, **layovers**, **duty crews,** useful for AVL and on-board ticketing systems.
- Data about the Accessibility of services to passengers with restricted mobility.
- Data is versioned with management **metadata** allowing updates across distributed systems

NeTEx describes data as XML document that can be exchanged by many different communication protocols (e.g. FTP, HTTP, SNMP) etc. It can therefore be used both for the bulk exchange of XML documents based on the Schema and for the dynamic exchange of individual data objects or groups of objects, for example as HTTP request using the SIRI framework.

NeTEx XML serialises complex PT models into a standard flat file format that can be processed cheaply and efficiently using mainstream modern computer technologies.

3.7 Conclusions

This Section presented in sufficient detail the following standards for transportation data representation formats: TPEG, Transmodel, SIRI, DATEX II, IFOPT, NeTEx. All of them provide adequate support for the various data categories presented in Section 2. In addition, TPEG and SIRI, both European initiatives, show the greatest support for environmental data. Among these two, TPEG has started gaining increased interest from commercial transportation service providers over Europe since the last couple of years.

4 Proprietary and Other Formats

For the sake of completion, after describing the most well-known (possibly open) standards and representation formats in the previous sections, this section outlines the proprietary formats that are supported by eCOMPASS commercial partners. In the project, two commercial partners participate: PTV and TomTom. Among these, PTV's navigation system does not rely on any proprietary format, it rather makes use of the open access TPEG standard described in Section 3. On the other hand, TomTom's web-based fleet management system frontend (namely WebFLEET) uses the company's proprietary format that we present in Section 4.1. Finally, we present some non-standard formats (available representation schemes that have not been proposed in the context of any standardisation activity, etc.).

4.1 TomTom WEBFLEET.connect API

TomTom WEBFLEET is the web-based application for managing vehicle fleets and orders accessing TomTom tracking and tracing functionality. WEBFLEET.connect is an API that allows integration of WEBFLEET services into any third party web-enabled application, under a specified license of use (Anonymous-3, 2012).

These are the primary functionalities accessible through WEBFLEET.connect:

- **Reports** – Retrieve data that correspond to the information contained in the reports generated within WEBFLEET
- **Messaging** – Send text to mobile units and retrieve incoming messages.
- **Addresses** – Insert, update and delete addresses and address groups as well as relations between addresses and address groups.
- **Orders** – Insert, send, update and delete orders and retrieve order status information.
- **Drivers** – Insert, update and delete drivers and retrieve driver status information.

WEBFLEET.connect can be made available to every customer with a valid WEBFLEET account. There should be at least one active object to make full use of the functionality offered by WEBFLEET.connect. WEBFLEET.connect provides access to the functionality of the WEBFLEET application in the following ways:

- ***Via HTTP requests***. WEBFLEET.connect uses the standardised transport protocol HTTP 1.1 for which compliance with IETF RFC 2616 is necessary, because it includes proper evaluation and handling of all HTTP response header values, e.g. character set, content and transfer encoding including chunked transfer encoding.
- ***Via SOAP requests***. WEBFLEET.connect supports the XML-SOAP message protocol for issuing requests over an HTTPS connection. The easiest way to use the SOAP interface is to use a SOAP toolkit appropriate for the desired programming platform (e.g. NetBeans or .net).

WEBFLEET.connect provides appropriate web service operations for returning various types of data. Some examples of supported operations that return information relevant to eCOMPASS include:

- *showTripReportExtern*: returns information on trips, such as duration, geographic latitude and longitude of source and destination points, fuel consumption in litres, carbon dioxide emissions in grams, distance measured by ecoPLUS in meters, etc.
- *showTripSummaryReportExtern*: as above

- *showWorkingTimes*: returns work time states changes of the (co)driver, the locations and vehicle
- *showAccelerationEvents*: This action shows unwanted driver behaviour. It shows a list of events with information on excessive acceleration, breaking or cornering, based on a threshold defined by the user.
- *showSpeedingEvents*: This action shows unwanted driver behaviour. It shows a list of trips or part of trips with information on excessive speeding.
- *geocodeAddress*: receives an address provided as request parameters and returns all possible matches.
- *calcRouteSimpleExtern*: Determines the route from a start location to an end location and calculates the resulting estimated time of arrival for a specific route-type.

4.2 Non-standard Formats

4.2.1 GTFS (Google Transit Data Feed Specification)

In summer 2006 the Google Transit website[20] published a new data exchange format, GTFS, (already introduced in Section 2.2.1) through which Transport authorities can make their data available to the Google transit site. The GTFS defines a common format for public transportation schedules and associated geographic information. GTFS "feeds" allow public transit agencies to publish their transit data and developers to write applications that consume that data in an interoperable way. Google Transit is one of a number of innovative Google services such as Google Maps, which provide rich search functions using a free-to-user business model funded by Google location aware advertising.

Since 2006 when it was first proposed, the GTFS specification has been enhanced through several versions. It has gained considerable momentum, with a number of data sets made available in the USA and many other different countries, and also a growing body of tools such as validators. The original focus of Google transit has been metropolitan area transport networks, comprising primarily bus, metro and ferry. Intercity rail is available and bus information is available in a few countries, such as Switzerland, Austria and Japan.

4.2.2 UNETRANS - ESRI data models

UNETRANS has been developed and funded by an American company called ESRI[21], as a generic data model for transportation applications, using ESRI's ArcGIS 8 software. It can be treated more as a general template to be used as a starting point for transportation data representation and not as an imposed standard. Users can modify the template as needed to suit particular purposes (e.g. transit, aviation). Now UNETRANS is supported as ESRI data Transportation Model.

UNETRANS is now hosted by ESRI[22]. A set of data templates are available in various formats, such as: XML, gdb, etc., with support for visual representation. Additional ESRI supported data models include: the Carbon Footprint Data Model (available in XML) and Environmental Regulated Facilities Data Model (in UML).

4.3 Conclusions

Among the eCOMPASS commercial partners, TomTom provides an API for public use some specific license. The provided API is web-based supporting both HTTP and SOAP requests. Google GTFS and ESRI UNETRANS comprise two non-standardised transportation data representation formats.

---

[20] http://www.google.com/transit

[21] http://www.esri.com

[22] http://www.ncgia.ucsb.edu/vital/unetrans

5 Online Available Traffic and Transportation Data

This section presents a list of publicly available traffic and transportation data provision services (e.g. TfL, SNCF, DB, etc.) with relevant environmental data where applicable, in most representative European cities. This survey reports on the possibly available APIs (e.g. RESTful web services) where supported for any of the data types presented in Section 2.

5.1 Public Transportation Data

5.1.1 Transport Direct (UK journey planner)

Transport Direct is a division of the UK Department for Transport to develop better information technology systems to support public transport. It developed and maintains the Transport Direct Portal[23], which is a public facing multi-modal journey planner. The front page of the UK journey planner is shown in Figure 10.

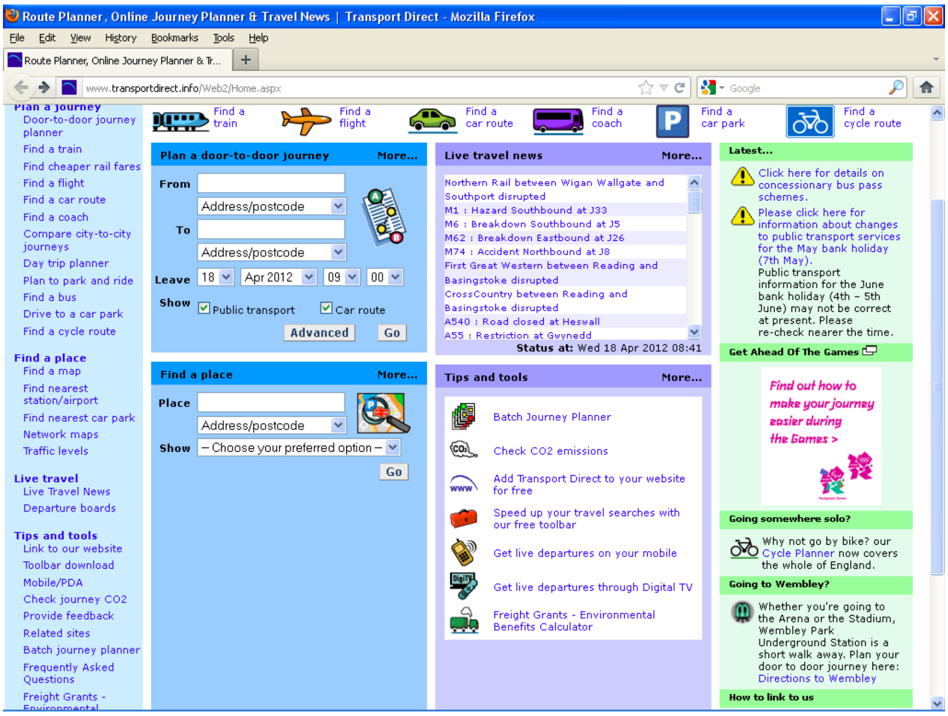

Figure 10: Main page of the Transport Direct Portal

Figure 10 shows the results of a query about available journey options from London to Edinburgh. The results can be shown in the form of a diagram, table or map. In the screen shot of Figure 10 information is provided in tabular format.

The table illustrated in Figure 11 also contains valuable information regarding accessibility issues of the transport modes that the passenger is going to use, time data of the trip duration and departure and arrival times. Moreover, for all places on the table, appropriate maps are provided showing the locations of each bus or metro stop and/or train station. Also the trip or its parts can be depicted on a map. This service is also available as a mobile application. A set of updated (2011) transport related data that the Travel Direct Service is using can be obtained in UK standardised formats in the National Public Transport Data

[23] http://www.transportdirect.info

Repository[24] (NPTDR) and is available for free to public developers. The NPTDR is a snapshot of all public transport schedules for the UK including bus, tram, train, ferry coach and rail.

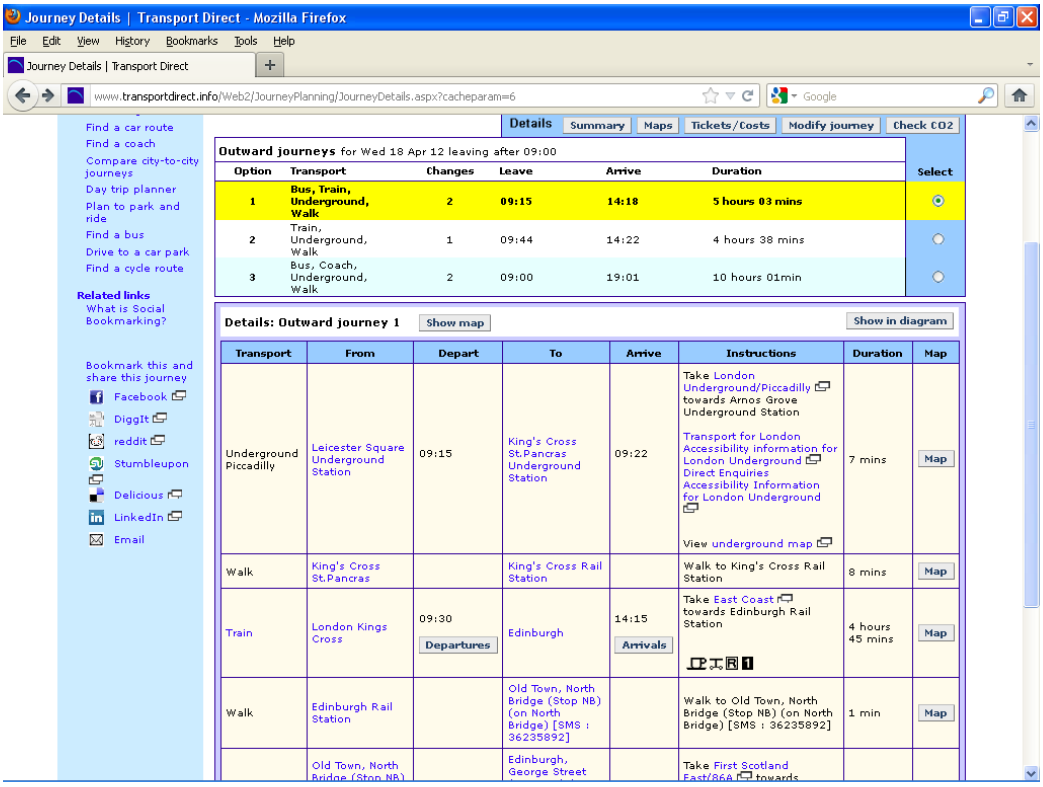

Figure 11: London to Edinburgh journey planning results

5.1.2 Google Transit and Athens Urban Transport Organisation - OASA S.A.

OASA S.A[25]. is the responsible authority for the planning, co-ordination and control of all public transport modes in the greater Athens area, including thermal buses, trolley buses, and metro. The OASA affiliates, i.e., ETHEL S.A. (thermal buses), ILPAP S.A. (trolley buses) and ISAP S.A. (Metro line 1), are responsible for the execution of transport services.

On the organisation's website there is a link which directs to Google Transit System in order to obtain information about a journey planning in Athens City territory. This means that the public transportation data from the various transport modes of Athens are collected and presented in GTFS format. The Google transit system supports public transportation data of more than 400 cities all over the world.
A test query which was submitted to the Google maps asking for directions regarding a journey within Athens territory and particularly from Agia Paraskevi to Nea Filadelfia (regions of Athens) returns the results that are depicted in Figure 12.

At the beginning of the journey planning, the system gives the opportunity to the users to select the most appropriate type of journey for them with respect to the estimated time duration of the journey.

[24] http://data.gov.uk/dataset/nptdr
[25] http://www.oasa.gr

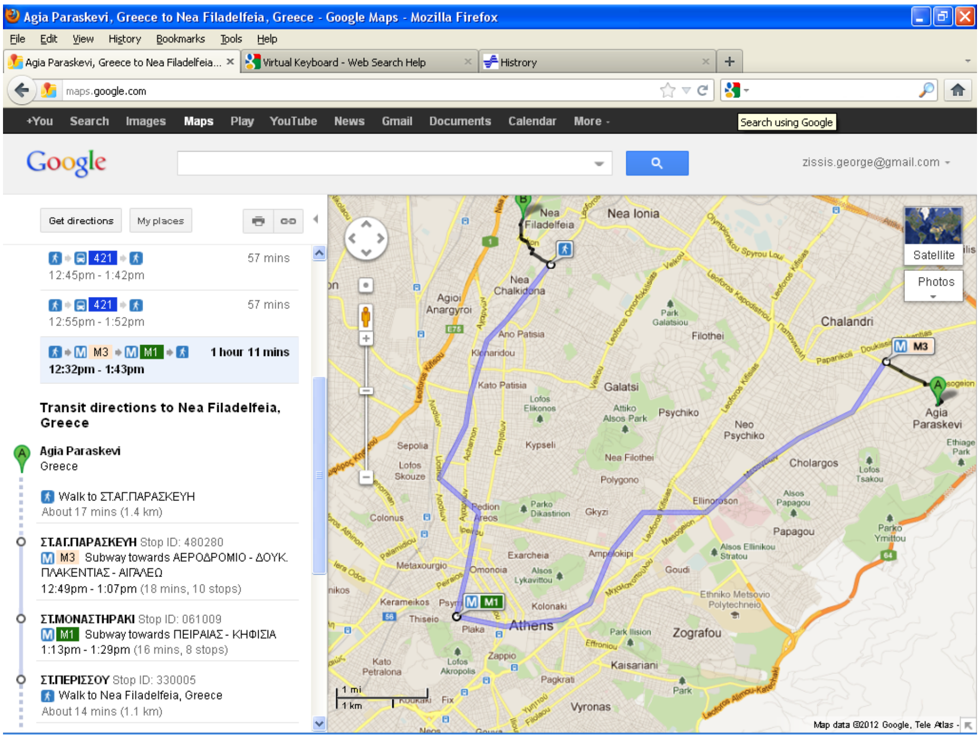

Figure 12: Agia Paraskevi to Nea Filadelfia journey planning results

As it is shown on the left of Figure 12, a list of journey directions and particularly transport mode changes is presented with additional information regarding the estimated travel duration and distance and also the name and the ID of the station and the direction that the passenger has to follow in order to arrive to the desired destination. On the right side of the screen a Google map is shown which highlights the outlined travel line that the passenger has to follow using the public transport. Also there are icons that represent the transport mode and the places where change of mode has to take place. Unfortunately ticket fare data is missing but there is a note referring to the phone number and name of the corresponding organisation, e.g. OASA.

Many of the existing GTFS public transportation data is available online[26]. The Athens Urban Transport Organisation GTFS data is available, and so is Madrid's (Spain) public transport data in GTFS format.

5.1.3 Transport for London

The main source of information about London transport service is the Transport for London (TfL) official site[27]. The website provides information on all types of public transport in London, routes, maps, journey planner and online tickets sales, and it also includes a guide to what to see and do. The official source for trains in London is the same as the official source for UK trains[28].

In Figure 13 the results of a test query for a journey planning from Victoria to London Zoo is presented. As it is seen there are four options for the current journey. If we select one of them we can see the details for the journey as it is shown in Figure 14. The detailed results

[26] http://www.gtfs-data-exchange.com

[27] http://www.tfl.gov.uk

[28] http://www.nationalrail.co.uk

contain information regarding availability of escalators, works that are taking place in the various stations, duration between stations, transportation frequency etc.

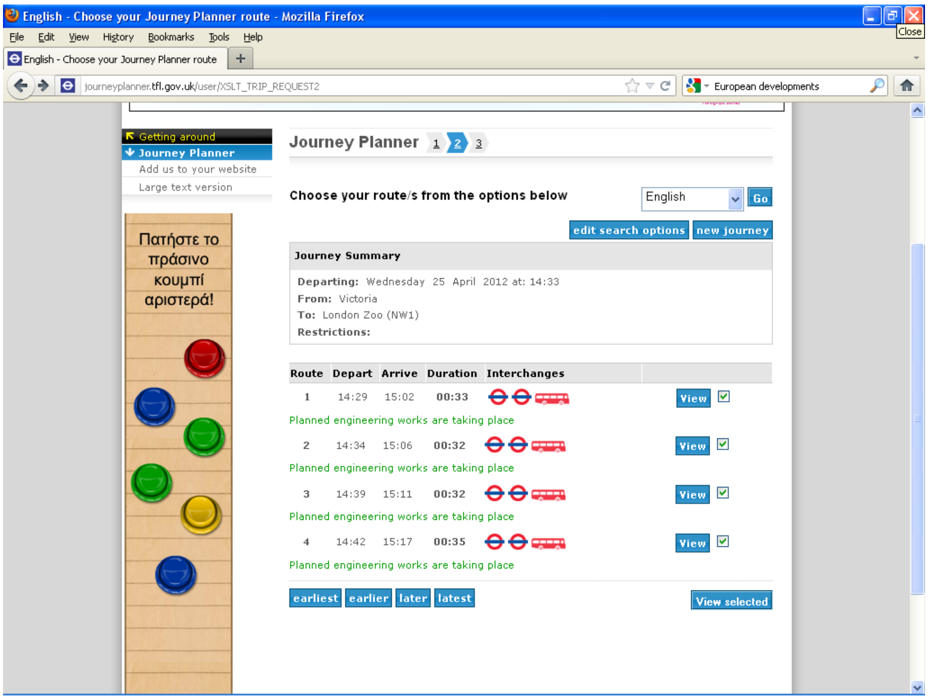

Figure 13 : from Victoria to Lonon Zoo journey planning

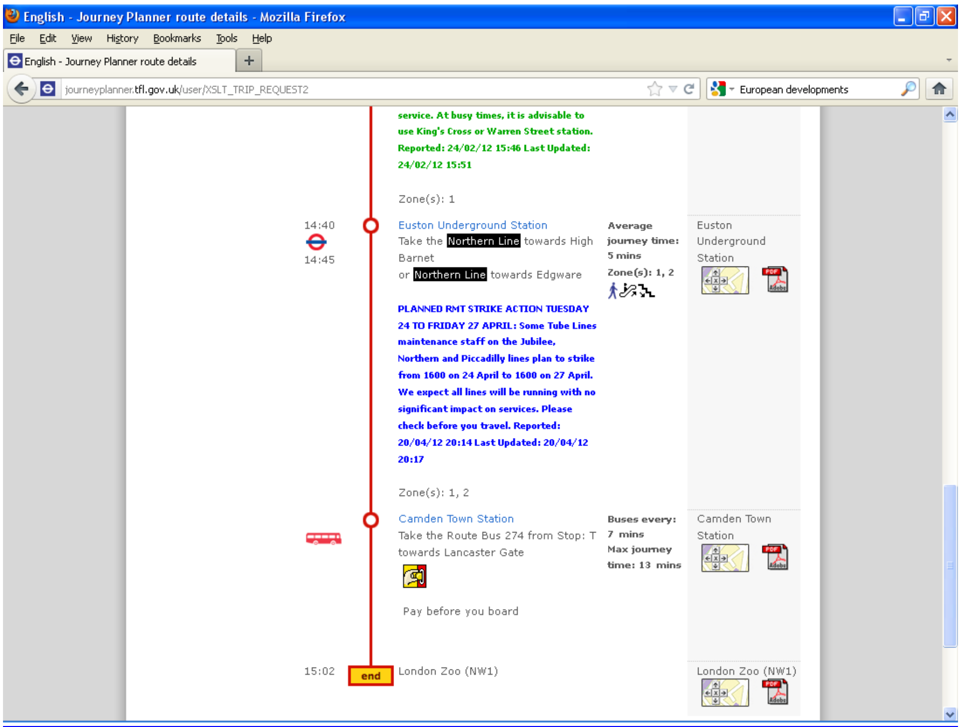

Figure 14 : Victoria to London Zoo route details

The official TfL data is available for developers. Transport for London feeds are supplied as XML and are provided online[29] (requires registration). Also, developer guidelines on how to

[29] http://www.tfl.gov.uk/businessandpartners/syndication/default.aspx

use these feeds can be found in (Anonymous-4, 2012). Access to data provided by TfL, is made available via appropriate RESTful web service interfaces, as well as XML data formats.

5.1.4 SNCF Transilien (French National Railway Company)

The French National Railway Company provides online transportation services, namely SNCF Transilien at its official website[30]. This website provides detailed information on all forms of public transport (train, RER, metro, tram, bus) in Paris urban and suburban areas, including routes, maps, a journey planner and ticket prices.

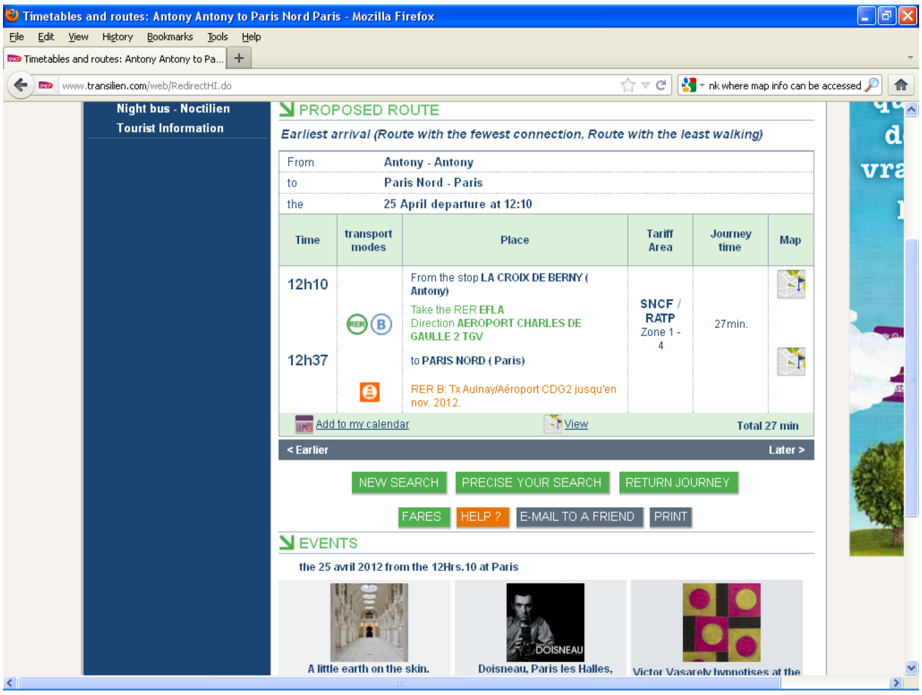

Figure 15: Antony to Paris Nord Journey planning results

Figure 15 shows the results returned on submission of a user query about planning a journey from Antony to Gares du Nord. For each part of the route the journey planner provides map info with the location of the stations, requested addresses etc., as well as an overall map of the journey route. The Transilien web service provides also information on the events that are taking place at nearby locations and on the current date.

All the data regarding the transportation in Paris are available only through the Transilien web service. The data provided by SNCF Transilien are proprietary and not available to public developers.

5.1.5 Berlin

The Berlin Transport Authority website[31] provides information for all the available transport modes in Berlin. The Berlin journey planner contains also information on POI (such as museums, historical places, stadiums etc.). At this website the user can also find information regarding the traffic problems in the city of Berlin.

[30] http://www.transilien.com/web/site
[31] http://www.bvg.de

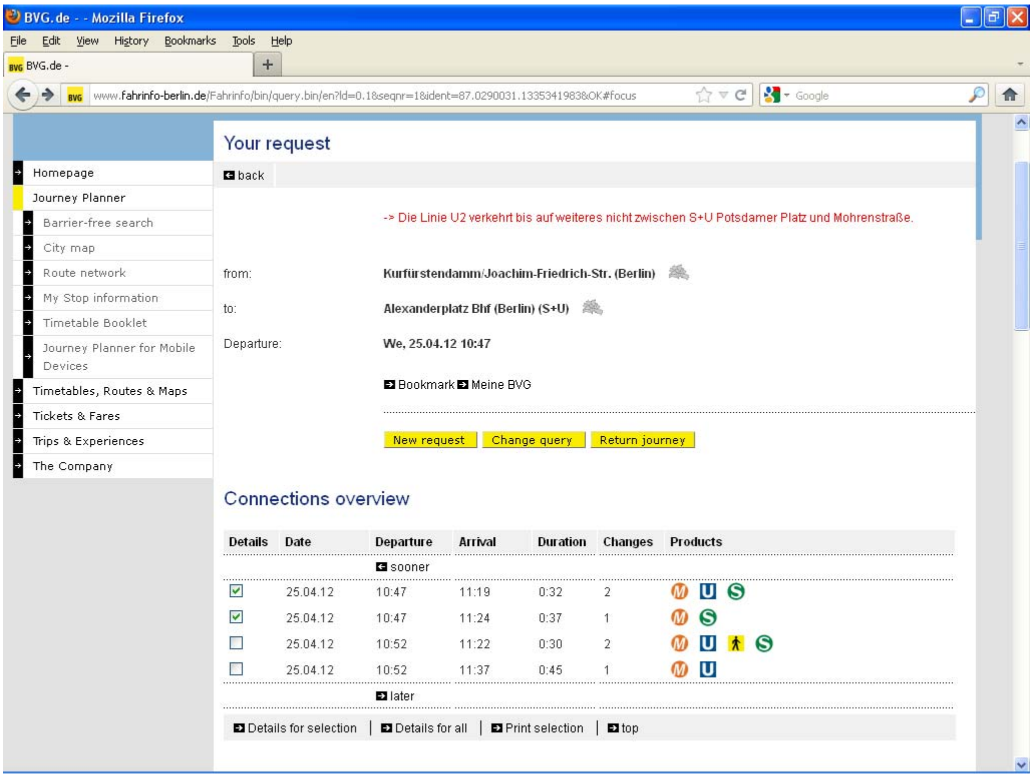

Figure 16: Kurfürstendamm/Joachim-Friedrich-Str. to Alexanderplatz Bhf. Journey planning results

Figure 16 shows the results that are listed after submitting a query for planning a journey from Kurfürstendamm/Joachim-Friedrich-Str. to Alexanderplatz Bhf. As it is seen we can select from a list of journey options with different transport modes and different travel time durations. In Figure 17 the results of the first journey option are presented in detail. The journey planner provides also information regarding potential problems due to dense traffic and lifts availability at the stations. All the addresses, places, stations that are shown on the journey planner, can be also viewed on a map.

The German national rail service is operated by the online service of Deutsche Bahn (DB)[32]. On the DB website there is also a journey planner where the user can use in order to obtain information regarding the available trains, travel fares and also to select to travel using various types of trains (e.g. fast vs. local).

Figure 18 shows the results of a query submitted for calculating a trip from Berlin to Bitburg using local trains. In the same figure information about the arrival and departure times for each transit station is also provided. Travel fare information is also included. DB also provides bicycle sharing information services[33].

The data hosted by Berlin Transport Authority and DB websites is proprietary and available only for online view using a web browser. No support for developers is provided.

5.1.6 Madrid

The transport information services for Madrid are provided online by the "Consorcio Transportes de Madrid" (CTM)[34]. The provided journey planner can combine information on different transport modes such as bus, metro and suburban trains.

[32] http://www.bahn.de/p/view/index.shtml

[33] http://www.callabike-interaktiv.de

[34] http://www.ctm-madrid.es

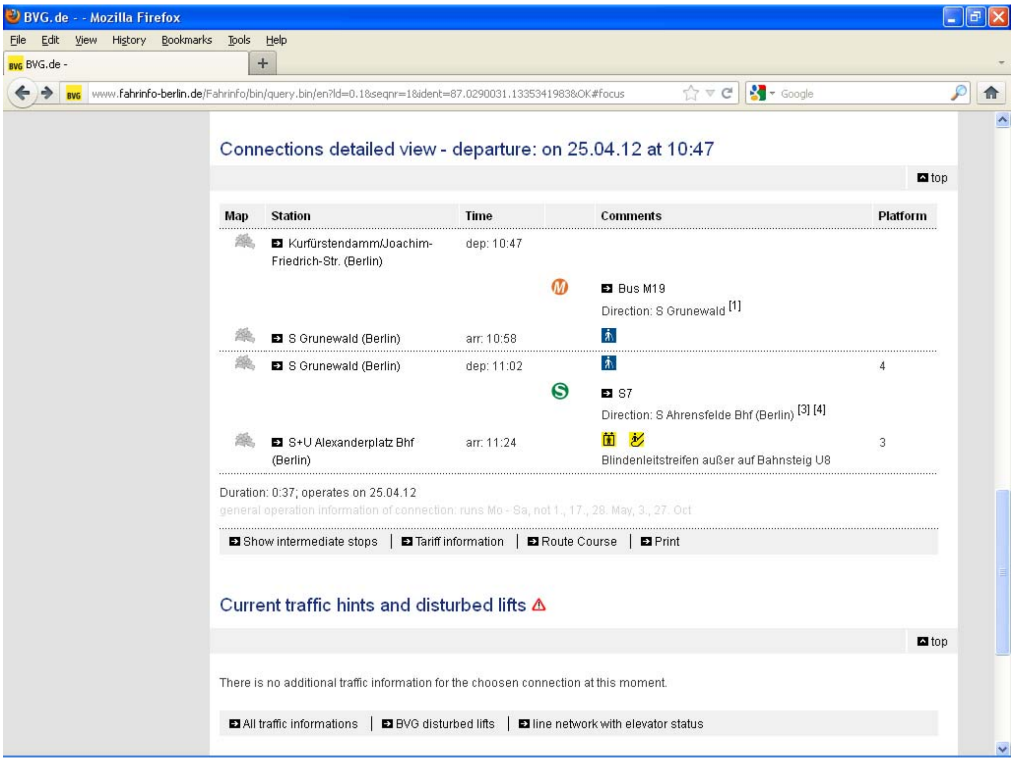

Figure 17 : Kurfürstendamm/Joachim-Friedrich-Str. to Alexanderplatz Bhf. Journey planning results (analytical presentation)

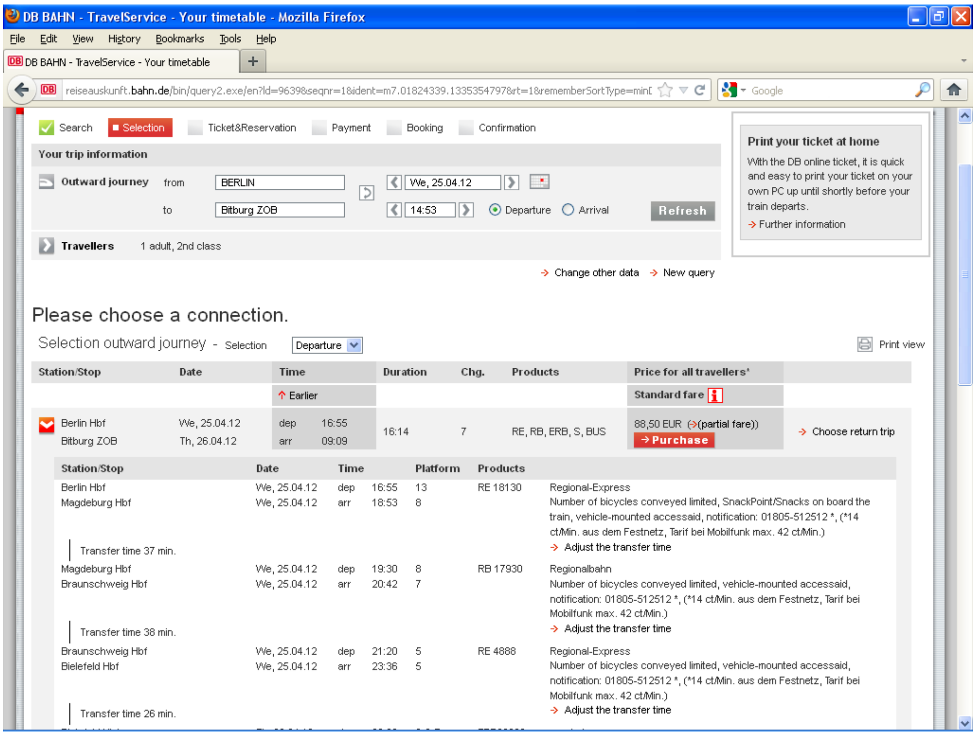

Figure 18 : Train travel planning from Berlin to Bitburg

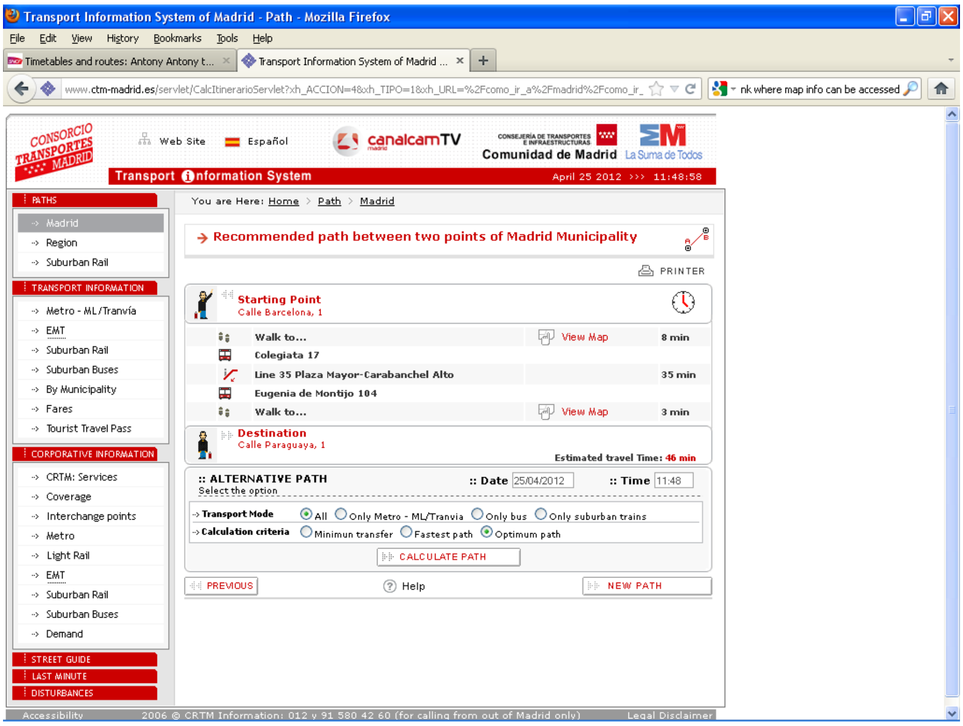

Figure 19: Calle Barcelona to Calle Paraguaya journey planning

Figure 19 shows the CTM website and specifically the results of a query submitted about the calculation of the route from Calle Barcelona to Calle Paraguaya. The system provides maps of the starting and destination points, travel times and means of public transport to be used.

At the official website of the Madrid's metro[35] information regarding metro lines, tickets and fares is available. On this website there is also a journey planner which provides info only for travellers that are going to use the metro. The transportation data is available in data in GTFS format.

5.1.7 ATAC Roma

The official website for the transportation services in Rome is provided by ATAC S.p.A[36]. and contains information for bus lines, metro lines, route planner, timetables, routes and maps, tickets and fares. The route planner provides information both for public transport and for private transport modes e.g. cars and motorbikes.

Figure 20 shows the web page that is returned after submitting a query for calculating a journey in Rome from via Brugnato to via Abbateggio. The results are presented on the map and in the form of a list of stop points. At each point of the journey there is available info about timetables and estimated travel duration from the previous point.

The official Italian Railways website[37] provides a journey planner with information about train stations, timetables, trains availability, travel fares etc. There is also a website for bicycle sharing[38].

The transport information both in Rome Transportation and Italian Railways websites is proprietary and available only for view on the official website user interface.

[35] http://www.metromadrid.es
[36] http://www.atac.roma.it
[37] http://www.trenitalia.com
[38] http://www.roma-n-bike.com

5.1.8 Transportation Services in other European cities

Barcelona. The main publicly available transportation services for Barcelona include

- The Transports Metropolitans de Barcelona official site[39] which provides information regarding bus lines, metro lines, tickets and travel cards. There is also a journey planner that combines information of different transport modes.
- Train lines information services[40]. All data regarding the transportation in Barcelona is available only through the web service of the official website.
- Sharing bicycles services[41]. This web service provides a journey planner. The route can be viewed on a Google Map along with information regarding the bicycle availability.

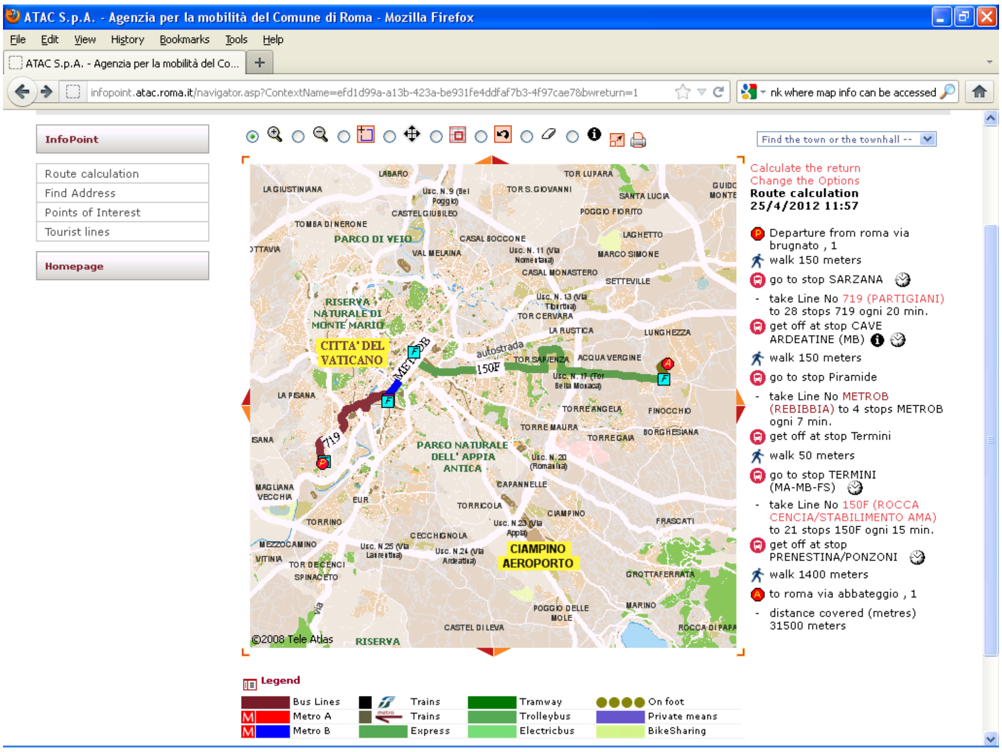

Figure 20: via Brugnato to via Abbateggio journey planning

All the transportation data for all the transport modes are proprietary and can be viewed only by humans on the browser. No support for developers is provided.

Venice. The local transport services in Venice are available at the ACTV, the Venice Public-Transit Agency (ACTV) website[42] en and provides information for land and water public transport routes. The transport information is proprietary and available only through the official website.

Amsterdam. The main publicly available transportation services for Amsterdam include tram, bus and metro and are presented on the GVB website[43]. This website provides information for bus lines, metro lines, timetables, routes, maps, tickets and fares. The information here is static and available only on the website.

[39] http://www.tmb.cat/en/home
[40] http://www.renfe.com
[41] http://www.bicing.cat
[42] http://www.actv.it
[43] http://www.gvb.nl/Pages/default.aspx

Greater Manchester. The official website of the Transport for Great Manchester[44] provides a journey planner for buses, trains, metrolink and accessible transport. Bus stop and bus time data are available to the developers for all services within the Greater Manchester boundary known technically as "ATCO-CIF[45]".

5.2 Traffic Data

5.2.1 Real time traffic information System (England) - Highways Agency

The Highways Agency[46] is responsible for the construction and maintenance of motorways and major trunk roads in England. This is the strategic network of roads used to move people and freight around the country. As it is mentioned at their website, the organisation is not responsible for all the roads in the UK. Roads in England that are not shown on the map and presented on the website are managed by Local Authorities. Also trunk roads and motorways in Scotland belong to the responsibility of Transport Scotland and those in Wales to the responsibility of the Welsh Assembly Government. It is worth mentioning that TomTom HD Traffic includes all information from the Highways Agency[47].

Real time traffic information is provided by a dedicated Highway Agency's webpage[48] depicted in Figure 21. The Highway Agency's webpage provides the following types of real time data to the user.

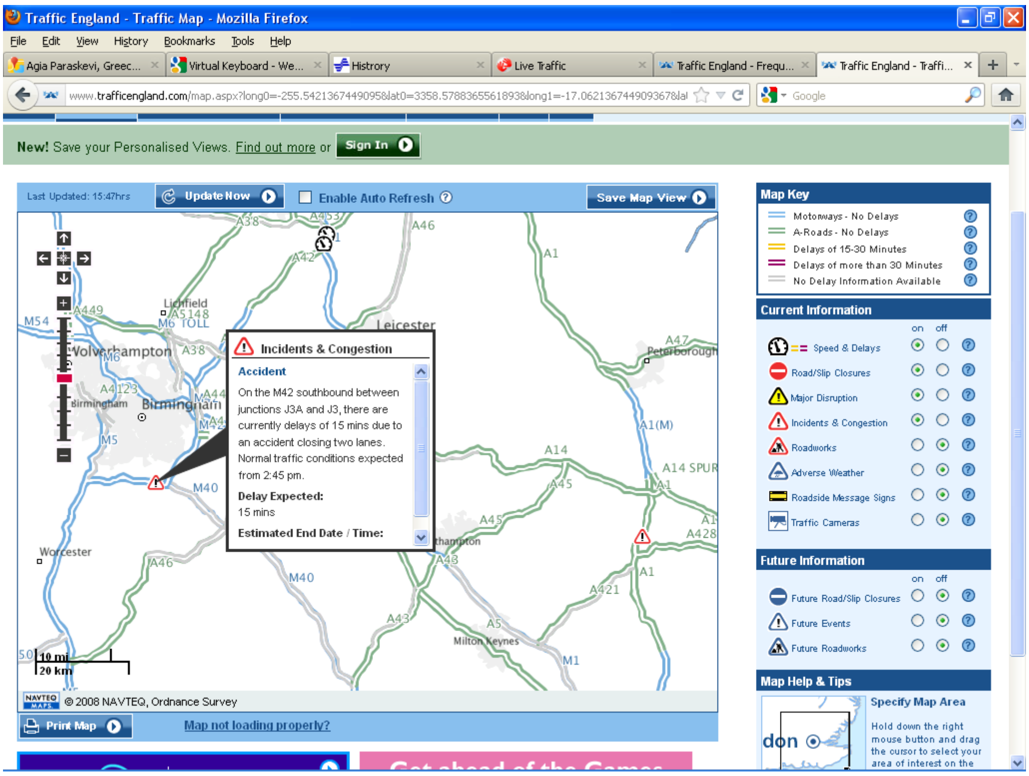

Figure 21: Accident message on Real time traffic data Highway Agency's web service

- Speed & Delay
- Road/Slip Closures
- Major Disruption

44 http://www.tfgm.com
45 http://datagm.org.uk/package/atco-cif
46 http://www.highways.gov.uk
47 http://www.highways.gov.uk/news/pressrelease.aspx?pressreleaseid=420364
48 http://www.trafficengland.com

- Incidents & Congestion
- Roadworks
- Adverse Weather
- Roadside Message Signs
- Traffic Cameras

and also provides future information such as

- Future Road/Slip Closures
- Future Events
- Future Roadworks

All these types of information that are available to the end user appear on the map that the web service offers. For each kind of the selected information a specific icon appears on the map on mouse over, which contains a certain, clear and quite informative message related to the provided type of data.

As an example, during our investigation of the England real time traffic data system an accident (Incidents & Congestion type of data) appeared on the map and the message was the following:

> *"Accident: On the M42 southbound between junctions J3A and J3, there are currently delays of 15 mins due to an accident closing two lanes. Normal Traffic conditions expected from 2:45 pm. Delay expected: 15 mins, Estimated End Date/Time: WED 18 APR 2012 14:45".*

Additional information related to the accident, such as Road/Slip Closures etc. are shown on the same web page in sufficient detail.

The web page is also available as a mobile application.

The Highways Agency, like all other government departments in UK is subject to the Freedom of Information Act 2000 ("the FOI Act") therefore the data provided in the web service are available but upon request. As it was mentioned on the agency's website

> *"Anyone can make a request for information. You do not need to be a UK citizen or living in the UK in order to make one. You may request any information held by or on the behalf of the Agency, regardless of how it is held. We hold a wide variety of information such as maps, plans, photographs, written information and information held on computer."*

In order to facilitate the request there is a form available online[49].

5.2.2 TomTom HD Traffic

The real time traffic data information system TomTom HD Traffic is available online[50]. This system provides data for the following list of countries:

- in Europe: Austria, Belgium, Czech Republic, Denmark, Finland, France, Germany, Ireland, Italy, Luxembourg, Netherlands, Norway, Poland, Portugal, Spain, Sweden, Switzerland and United Kingdom
- in Americas: Canada and United States
- in Africa: South Africa and
- in Asia Pacific: Australia and New Zealand

The types of traffic data provided by TomTom HD Traffic are the following:

---

[49] http://www.highways.gov.uk/aboutus/documents/foi_pplication%282%29.pdf
[50] http://www.tomtom.com/livetraffic

- Slow traffic
- Queuing traffic
- Stationary traffic
- Possible delays
- Traffic Jam
- Accident
- Roadworks
- Road Blocked
- Lane closure
- Number of traffic incident or warnings
- Traffic information available

For each type of traffic data there is a specific icon which represents the content of the current data. All the types of data are covering the whole national road network of the countries.

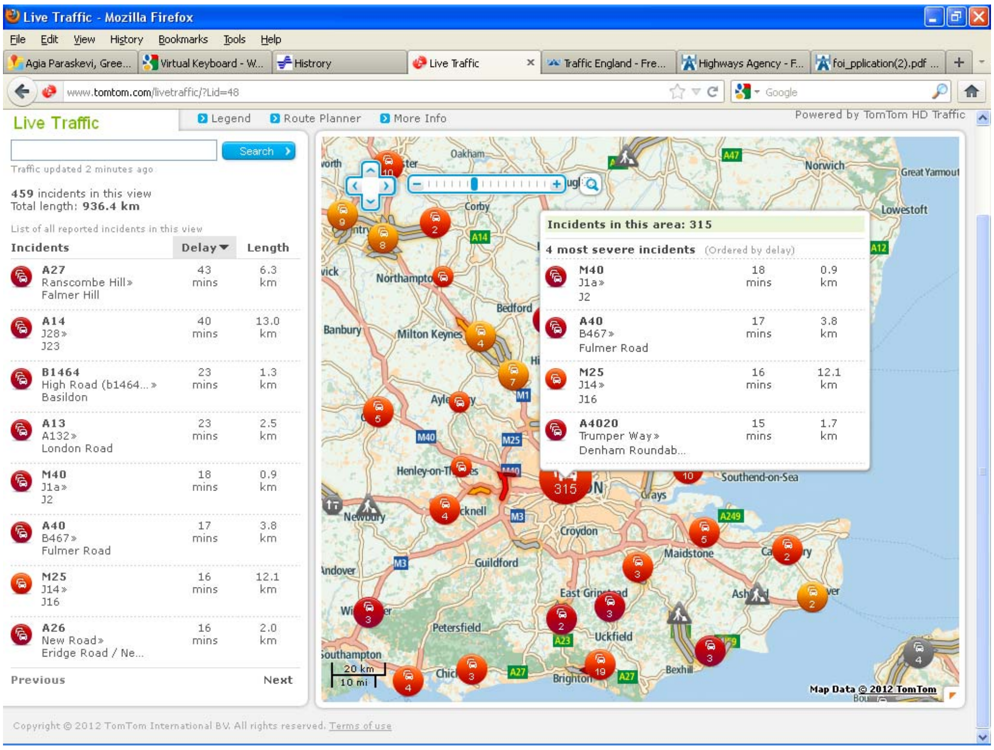

Figure 22: Live Traffic web system by TomTom

In Figure 22 TomTom's Live Traffic system in the South region of Britain covering also the London territory is presented. On the circles presented with red, yellow and orange colours appears a number which informs the user about the number of incidents which took place in this specific area of the map. This is only the "bird's eye view" when zoomed out very far. When the user zooms in, the actual incidents precisely with their length and intensity are shown on the map. It appears that during our investigation on the TomTom's web service 315 incidents took place only in the city of London. As mouse passes over, a message appears on the map with the four most serious (related to the traffic delays) incidents showing the location of the incident, the estimated delay and the impact in terms of distance coverage of the incident. On the left side of the screen, a list of the most serious (again related to the traffic delays) incidents are shown. The system also informs the users about the total number of incidents within the territory shown on the map (right side). The data provided by this web service is property of TomTom and not available for public use.

5.3 POI data

There is a tremendous amount of royalty free POI sets of data available on the Internet regarding different types of POI datasets for different manufacturing systems. For example on the website http://downloadpoi.com there are POIs for a list of countries such as Australia, United States, Germany, France, Austria, Russia, Slovakia, Poland, Romania, Czech Republic, Belgium, Italy and Canada. The available sets of POI data for Germany are the following: German Hotels, German Postbank ATM, German Pubs, German McDonalds, German Pharmacy – Apotheke, German Museums, German Campings, German Citibank ATM-s. The sets of the above mentioned POI data are compatible with the following manufacturing systems Garmin CSV, Garmin GPX, TomTom OV2, iGO 8 KML.

List of other related Web Sites:

- roadmapgps.com
- brighthub.com
- gps-poi-us.com
- poi.gps-data-team.com
- downloadpoi.com
- forums.cnet.com
- gps-software-hub.com
- tomtom.gps-data-team.com
- poiconvert.free.fr

5.4 Fleet Management Data

Companies do not in general share their transportation data, as this data would allow competitors to get company internal information. In case of co-operation between companies, e.g. sharing of one fleet, fleet management data is being exchanged either on a bilateral, proprietary interface level, or via a common database. In any case, fleet management data is restrictive and tends to be not available to the public. Major truck manufacturers agreed to give third parties access to vehicle data through the Fleet Management Standard[51] (FMS) group.

Some recent research activities in the field of 'City Logistics Authorities', developed city logistics concepts, to force trucking companies, to announce their trips to these 'City Logistics Management' by establishing processes to request access to the city area. (see e.g. eCoMove[52]).

5.5 Environmental Data

5.5.1 CO_2 Transport emissions Data system

The previously presented Transport Direct Portal[23] shown in Figure 10 provides also CO_2 emission data. As in the previous case let us assume that a the user plans a journey from London to Edinburgh. Along with the public transport data the system provides data regarding the CO_2 emissions of different transport modes, as it is shown in Figure 23.

[51] http://www.fms-standard.com/index.htm

[52] http://www.ecomove-project.eu

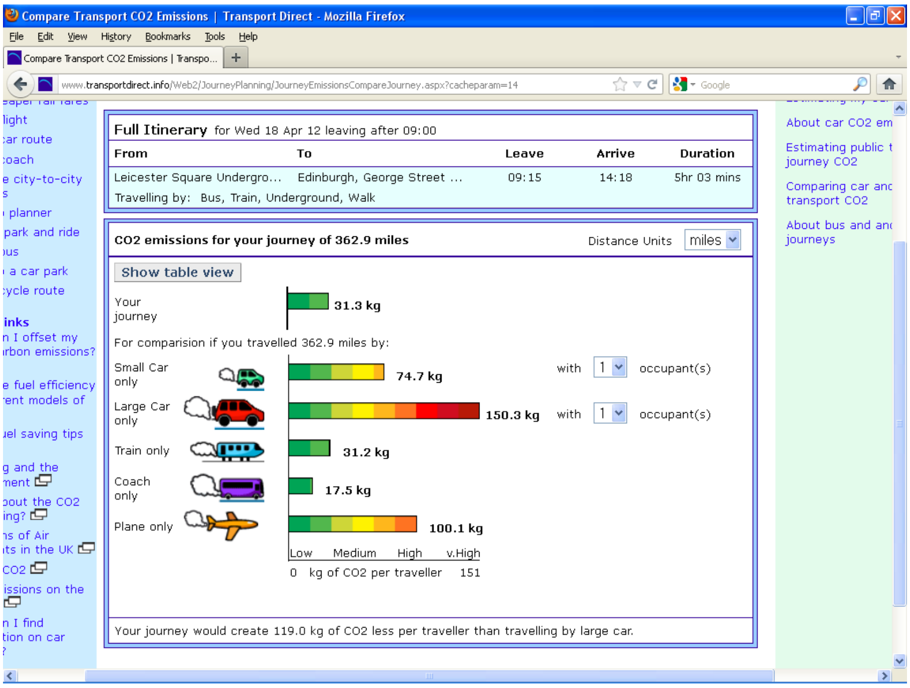

Figure 23: CO_2 emissions by different transport modes

It is clearly depicted that the CO_2 emissions by using other modes of transport other than public transport is quite high. The visualisation of these data was made using a standard A-G chart, from green to red as it is used for the energy efficiency rating in electrical goods. This chart's usage has now been widened and is recommended for use by the Vehicle Certification Agency (VCA) for all new cars. Transport Direct assumes that cars with one occupant and low emissions are likely to be in the green and yellow area of the bar and cars with one occupant and high emissions will be in the orange and red area. This “rating” will remain constant irrespective of the distance of the journey, but the amount of emissions generated will increase as the journey distance increases.

5.5.2 L’ ECOcomparateur by SNCF

The ECOcomparateur is a journey planning system provided by SNCF (National Railway Company of France) which mainly provides data for journey planning using French national railways but also makes an estimation of travel time, costs and CO_2 emissions using other transport modes and particularly car or airplane. The web service is available online[53] (see Figure 24 above for a screenshot of the query submission user interface).

Figure 24: L’ ECOcomparateur journey Paris to Marseille planning

In Figure 25 the ECOcomparateur system is shown for planning a journey from Paris to Marseille. Adding the data of the starting and the destination point of the journey we can

[53] http://www.voyages-sncf.com

proceed to the calculation of the journey. The system provides travel data such as cost and duration of the travel and also train data related to the journey. In addition to the returned data the estimation of travel data with respect to the duration, cost and CO_2 emissions using different mode of transport (car and airplane) are shown.

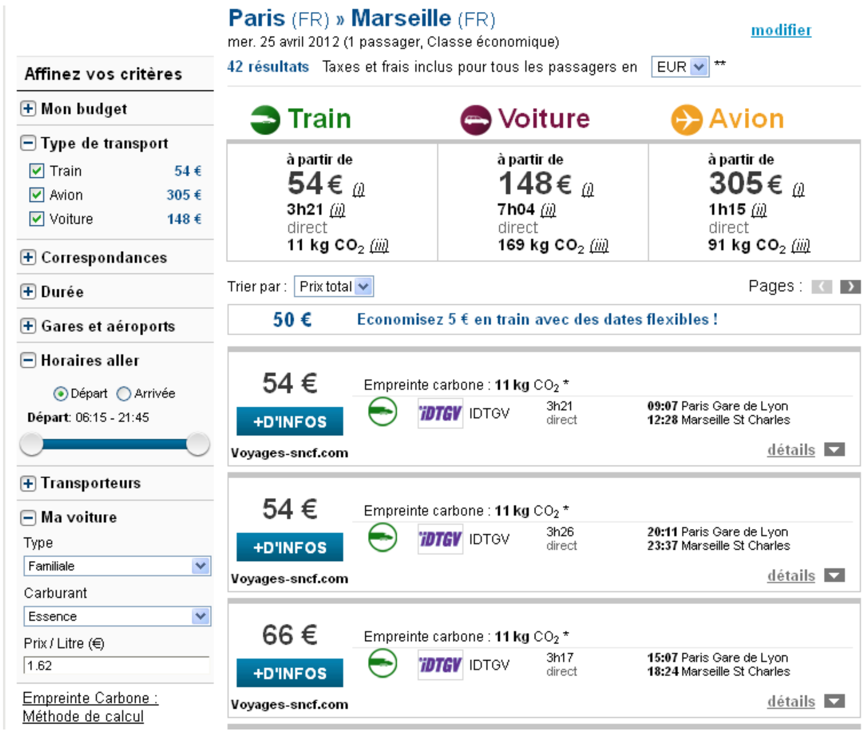

Figure 25: L' ECOcomparateur system results for the Paris to Marseille journey

It is more than obvious that the travel using the railways it is more effective in terms of cost and environmental impact. In the website the methodology for calculating fuel costs, CO_2 emissions and Airplane fares is described.

5.5.3 Real time weather data by Highway Agency (England)

The Highway Agency in England[48] besides the real time traffic data provides quite accurate frequent and up to date data regarding the weather conditions on the depicted map and roads which are under Agency's responsibility. Figure 26 shows this option.

It is worth mentioning that the provided data is also related to the current location. For example the message that is shown in Figure 26 is the following:

M1 J6 on A405 (location data), Current Temp: 8 C, Current Wind: 11 mph and Condition: Raining.

All these weather related data are very important for the traveller or the driver of a vehicle. Taking into account the weather conditions one can decide whether to travel or not under the current conditions. According to the Highway Agency statement on the Web Site data is available to the public.

5.6 Conclusions

Based on the previous survey, public access to public transportation data is only supported in all sites in the UK via Restful web service API and XML. Also Madrid and Athens provide data in GTFS format. As for the remaining cities the public transportation services provide no support for developers, and no access to data is allowed by any public means, i.e., data is only available for view through the corresponding web interfaces. An exception is the information services system for Paris, which provides the option for embedding the journey planner component within a third-party web page (in a “web-mashup” fashion). Most of the sites offer a journey planner capable of providing multi-modal route planner with further information available about the various parts of the trip. Some of them also provide environmental information. Regarding Berlin, which is going to be one of the eCOMPASS pilot sites, the Berlin Authority website as well as DB provide information on all available transport modes. Unfortunately no direct programmatic access to their data (i.e. via a specific API) is available. Regarding traffic data, the Highways Agency is responsible for maintaining relevant data included in TomTom’s HD Traffic system. Traffic data, as well as fleet management data are kept as proprietary and no direct access is provided.

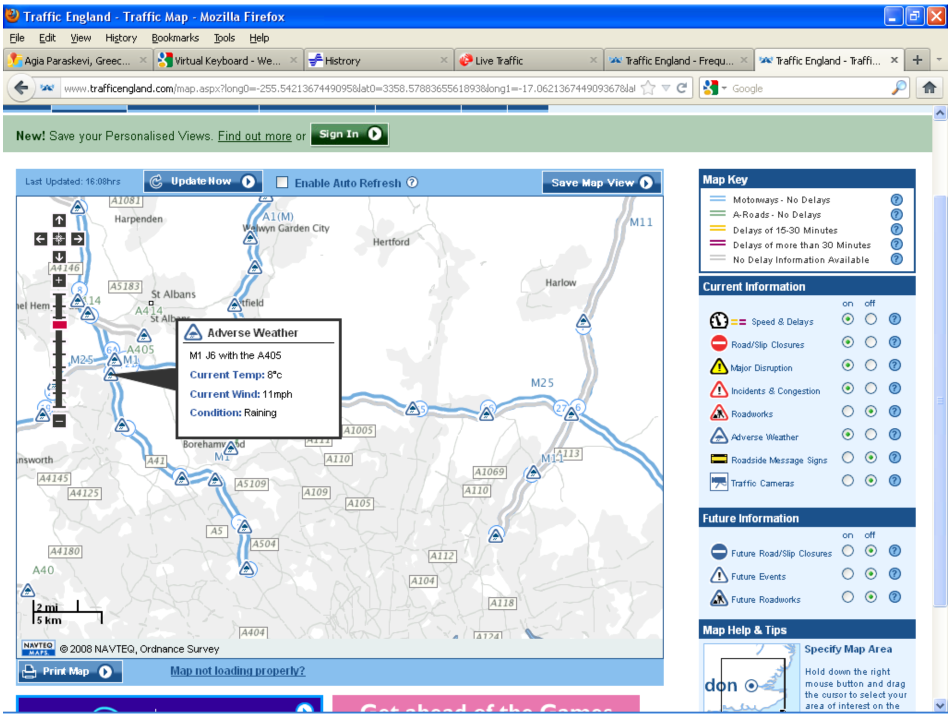

Figure 26: Real Time Weather data provided by Highway Agency

6 Summary & Conclusions

The first part of this deliverable describes the various categories of data that are required as input data by the eCOMPASS routing algorithms, to be implemented in WP2, WP3 and therefore by the applications that will be implemented in WP5 and integrate the aforementioned algorithms. A first try is attempted here for the identification of those data categories that should be supported. The categories that have been examined are the following: Public Transportation Data, Traffic Data, User Location Data, Points of Interest, Fleet Management Data, Environmental Data. Some of these categories can be supported via appropriate APIs and remote (web-based) requests via e.g. SOAP and HTTP communication and message exchange protocols. Others do not require any mechanism to obtain the data via remote requests because they come from sources that are coupled with the device, such as location data received by a GPS sensor which is integrated in the user's PND. Others do not provide public access to data.

This deliverable presented standard and non-standard transportation data representation formats and also reviewed online transportation services by examining potential access to real data, available online in various ways, e.g. via web interfaces or services. We tried to cover the most relevant to eCOMPASS cases in order to get a sufficient overview of the current online information status, underlying the various possibilities for accessing data (e.g. open access availability vs. commercial use, available APIs vs. human access only, etc.). This part of the survey is very important because it will highlight the real possibilities for getting access to data especially during the deployment of the pilot sites, which is planned after the development phase of the project. For example, the survey revealed that the pilot site in Berlin should be provided proprietary access to public transportation data, as not public access to data is available.

Based on the review on online available data, the aim of this deliverable is to provide a recommendation of how the aforementioned categories can be supported. In particular, the following options are foreseen:

- ***Public Transportation Data***. In the case of London real time online access is provided, however Berlin (eCOMPASS pilot site) services do not provide support for accessing data via any API. Therefore, no data interfaces can be provided, as the data can be available only on a contract basis. For Madrid and Athens static data can be accessed in the form of GTFS.
- ***Traffic Data***: only access to proprietary data is feasible. In this case, we recommend to use TomTom HD Traffic data and to examine the conditions (e.g., license issues, etc.) under which access to this data can become available.
- ***User Location Data*** are supported on the device, hence no special treatment of user location data is required.
- ***Points of Interest***. Royalty free access is available on the Internet, hence appropriate interfaces can be developed to gather POI-related data.
- ***Fleet Management Data***: only proprietary access. No interfaces can be developed, but the possibilities and the conditions should be examined for getting access to data by PTV.
- ***Environmental Data:*** no public access is available. The real possibilities and the conditions under which access to environmental data can be achieved (e.g.

potentially through the proprietary systems provided by PTV and TomTom) should be examined.

Regarding the aforementioned categories the eCOMPASS working team should make the final decision on which data categories will be supported based on the above recommendations, as well as on the standards that were reviewed in section 4.

In particular, this deliverable, after describing the eCOMPASS data categories, underlies the importance of adherence to standards, and it therefore presents the most representative and well-known standards for various types of transportation-related data. The standards presented here include: Transmodel, TPEG, SIRI, DATEX II, IFOPT, NeTEx and cover several aspects of the aforementioned data categories. For the sake of completeness, we present along with the previous list of standardisation efforts, a list of proprietary and non-standardised data representation formats that are supported by significant companies in the field (TomTom, Google, ESRI).

To summarise there are generally open standards for representing data in any of the data categories to be supported in eCOMPASS, as well as open but not standardised formats, of which Google's GTFS is the most significant one. Most of them support data representation in XML. Regarding access to online data resources, it seems that transportation data are available for London and Manchester (as well as for all UK cities) via SOAP and RESTful interfaces, whereas Madrid and Athens provide public access to their data but in GTFS format. For Berlin, Paris (expect bike sharing data), Amsterdam, Rome, Vienna and all remaining European cities there is no support for public access to transportation data.

For the underlying formal foundations of the future work within eCOMPASS, we identified TPEG and SIRI as robust and well-developed basis for being able to deal with the diversity of information required as well as the strong structuring and modularity that will be required. Both of these standards mark a significant point of departure for future work as they provide the most significant support for environmental data compared to the current state of the art, in which coverage by the remaining standards for environmental and eco-friendly data is limited to less expressive formalisms and for which structuring mechanisms are extremely limited. Among the two standards, TPEG has gained significant momentum over the last couple of years by commercial service providers in France, Germany, Italy and the UK who have begun to launch trials and plan next generation traffic services. The commercial traffic information sector in Europe is showing a clear commitment to launching TPEG services for the automotive industry. In addition to the commercial TPEG services there are also a significant number of public providers considering TPEG in Europe. Based on this analysis we recommend that eCOMPASS should provide support for TPEG, as well as any proprietary data supported by the commercial partners, TomTom and PTV.

A challenge for the next phase of project development will be to enable the appropriate software modules and mechanisms to bring these eco friendliness-related aspects together with the current tools and applications for personal or fleet navigation. This combination is itself expected to advance project development practices significantly. For those cases that will be selected, appropriate software interface mechanisms will be developed and reported in Deliverable D4.1.2.

References

Anonymous-1 (2012). "Google Developers", Transit website. Online available at: https://developers.google.com/transit/gtfs/ (accessed April 25)

Anonymous-2 (2012). Road traffic and transport telematics - Public transport - Identification of fixed objects in public transport, Technical Committee CEN/TC 278, 12/2007.

Anonymous-3 (2012). *WEBFLEET.connect Reference Guide, v.1.11.0*, Copyright 2012 TomTom Business Solutions.

Anonymous-4 (2012). *Syndication developer guidelines* : *Transport for London XML feeds.*

Online available at:
http://www.tfl.gov.uk/tfl/businessandpartners/syndication/assets/syndication-developer-guidelines.pdf (accessed 24 April 2012).

Berg Insight (2012). *ITS in Public Transport*. Berg Insight's M2M Research Series.

CEN – ENV ISO 14825: 1996 Geographic Data Files (GDF3.0), CEN/TC278 (Road Traffic and Transport Telematics).

Commitee of Deputies (2005). Road Safety Grouß (CEMT/CS/SR 2005 1)

Devices (PND), Research Program TomTom-DLR, Deutsches Zentrum für Luft- und Raumfahrt e.V. in der Helmholtz-Gemeinschaft.

European Telecommunications Standards Institute (ETSI) (1995). "Radio broadcasting system; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers" ETS 300 401.

ISO 14825 : 2002 Intelligent Transport Systems – Geographic Data Files – Overall Data Specifications (GDF4.0), ISO/TC204 (Intelligent Transport Systems), 2002

Knowles N. (2008). "SIRI Handbook & Functional Service Diagrams" Version 0.13. Kizoom Limited.

Knowles, N. and Miller, P. (2008). "A Transmodel based XML schema for the Google Transit Feed Specification With a GTFS / Transmodel comparison", Crown Copyright, Kizoom Limited.

International Association of Public transport (UITP). *Public Transport Statistics Report.* Issue 1. Online available: http://www.uitp.org/mos/pics/stats/survey_bus_fleet.pdf (accessed 24 April 2012).

Popescu A. (ed.) (2010). *Geolocation API Specification - W3C Candidate Recommendation*. Online available: http://www.w3.org/TR/geolocation-API/ (accessed 24 April 2012).

Sohr, A. and Wagner P. (2011). Project drive test – comparison of five Personal Navigation

Transport Protocol Experts Group (TPEG). Binary Specifications EBU BPN 027 and CEN ISO TS 18234.

Werner H. (1995). *TMC Handbook – Location Table Exchange Format.* Traffic Message Channel Forum, 1/10/2005.

A.1 Annex I: GTFS

A.1.1 Google Transit file architecture

A.1.1.1 Feed Files

This specification defines the following files along with their associated content:

Filename	Required	Defines
agency.txt	Required	One or more transit agencies that provide the data in this feed.
stops.txt	Required	Individual locations where vehicles pick up or drop off passengers.
routes.txt	Required	Transit routes. A route is a group of trips that are displayed to riders as a single service.
trips.txt	Required	Trips for each route. A trip is a sequence of two or more stops that occurs at specific time.
stop_times.txt	Required	Times that a vehicle arrives at and departs from individual stops for each trip.
calendar.txt	Required	Dates for service IDs using a weekly schedule. Specify when service starts and ends, as well as days of the week where service is available.
calendar_dates.txt	Optional	Exceptions for the service IDs defined in the calendar.txt file. If calendar_dates.txt includes ALL dates of service, this file may be specified instead of calendar.txt.
fare_attributes.txt	Optional	Fare information for a transit organisation's routes.
fare_rules.txt	Optional	Rules for applying fare information for a transit organisation's routes.
shapes.txt	Optional	Rules for drawing lines on a map to represent a transit organisation's routes.
frequencies.txt	Optional	Headway (time between trips) for routes with variable frequency of service.
transfers.txt	Optional	Rules for making connections at transfer points between routes.
feed_info.txt	Optional	Additional information about the feed itself, including publisher, version, and expiration information.

A.1.2 Field Definitions

A.1.2.1 agency.txt

Field Name	Required	Details
agency_id	Optional	The **agency_id** field is an ID that uniquely identifies a transit agency. A transit feed may represent data from more than one agency. The **agency_id** is dataset unique. This field is optional for transit feeds that only contain data for a single agency.
agency_name	Required	The **agency_name** field contains the full name of the transit agency. Google Maps will display this name.
agency_url	Required	The **agency_url** field contains the URL of the transit agency. The value must be a fully qualified URL that includes **http://** or **https://**, and any special characters in the URL must be correctly escaped. See http://www.w3.org/Addressing/URL/4_URI_Recommentations.html for a description of how to create fully qualified URL values.
agency_timezone	Required	The **agency_timezone** field contains the timezone where the transit agency is located. Timezone names never contain the space character but may contain an underscore. Please refer to http://en.wikipedia.org/wiki/List_of_tz_zones for a list of valid values. If multiple agencies are specified in the feed, each must have the same **agency_timezone**.
agency_lang	Optional	The **agency_lang** field contains a two-letter ISO 639-1 code for the primary language used by this transit agency. The language code is case-insensitive (both en and EN are accepted). This setting defines capitalisation rules and other language-specific settings for all text contained in this transit agency's feed. Please refer to http://www.loc.gov/standards/iso639-2/php/code_list.php for a list of valid values.
agency_phone	Optional	The **agency_phone** field contains a single voice telephone number for the specified agency. This field is a string value that presents the telephone number as typical for the agency's service area. It can and should contain punctuation marks to group the digits of the number. Dialable text (for example, TriMet's "503-238-RIDE") is permitted, but the field must not contain any other descriptive text.
agency_fare_url	Optional	The **agency_fare_url** specifies the URL of a web page that allows a rider to purchase tickets or

		other fare instruments for that agency online. The value must be a fully qualified URL that includes **http://** or **https://**, and any special characters in the URL must be correctly escaped. See http://www.w3.org/Addressing/URL/4_URI_Recommentations.html for a description of how to create fully qualified URL values.

A.1.2.2 stops.txt

Field Name	Required	Details
stop_id	Required	The **stop_id** field contains an ID that uniquely identifies a stop or station. Multiple routes may use the same stop. The **stop_id** is dataset unique.
stop_code	Optional	The **stop_code** field contains short text or a number that uniquely identifies the stop for passengers. Stop codes are often used in phone-based transit information systems or printed on stop signage to make it easier for riders to get a stop schedule or real time arrival information for a particular stop. The stop_code field should only be used for stop codes that are displayed to passengers. For internal codes, use **stop_id**. This field should be left blank for stops without a code.
stop_name	Required	The **stop_name** field contains the name of a stop or station. Please use a name that people will understand in the local and tourist vernacular.
stop_desc	Optional	The **stop_desc** field contains a description of a stop. Please provide useful, quality information. Do not simply duplicate the name of the stop.
stop_lat	Required	The **stop_lat** field contains the latitude of a stop or station. The field value must be a valid WGS 84 latitude.
stop_lon	Required	The **stop_lon** field contains the longitude of a stop or station. The field value must be a valid WGS 84 longitude value from -180 to 180.
zone_id	Optional	The **zone_id** field defines the fare zone for a stop ID. Zone IDs are required if you want to provide fare information using fare_rules.txt. If this stop ID represents a station, the zone ID is ignored.

<table>
<tr>
<td>stop_url</td>
<td>Optional</td>
<td>The stop_url field contains the URL of a web page about a particular stop. This should be different from the agency_url and the route_url fields.

The value must be a fully qualified URL that includes http:// or https://, and any special characters in the URL must be correctly escaped. See http://www.w3.org/Addressing/URL/4_URI_Recommentations.html for a description of how to create fully qualified URL values.</td>
</tr>
<tr>
<td>location_type</td>
<td>Optional</td>
<td>The location_type field identifies whether this stop ID represents a stop or station. If no location type is specified, or the location_type is blank, stop IDs are treated as stops. Stations may have different properties from stops when they are represented on a map or used in trip planning.

The location type field can have the following values:

• 0 or blank - Stop. A location where passengers board or disembark from a transit vehicle.
• 1 - Station. A physical structure or area that contains one or more stop.</td>
</tr>
<tr>
<td>parent_station</td>
<td>Optional</td>
<td>For stops that are physically located inside stations, the parent_station field identifies the station associated with the stop. To use this field, stops.txt must also contain a row where this stop ID is assigned location type=1.

<table>
<tr><th>This stop ID represents...</th><th>This entry's location type...</th><th>This entry's parent_station field contains...</th></tr>
<tr><td>A stop located inside a station.</td><td>0 or blank</td><td>The stop ID of the station where this stop is located. The stop referenced by parent_station must have location_type=1.</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><td></td><td></td><td><table>
<tr><td>A stop located outside a station.</td><td>0 or blank</td><td>A blank value. The parent_station field doesn't apply to this stop.</td></tr>
<tr><td>A station.</td><td>1</td><td>A blank value. Stations can't contain other stations.</td></tr>
</table></td></tr>
<tr><td>stop_timezone</td><td>Optional</td><td>The stop_timezone field contains the timezone in which this stop or station is located. Please refer to Wikipedia List of Timezones for a list of valid values. If omitted, the stop should be assumed to be located in the timezone specified by agency_timezone in agency.txt.

When a stop has a parent station, the stop is considered to be in the timezone specified by the parent station's stop_timezone value. If the parent has no stop_timezone value, the stops that belong to that station are assumed to be in the timezone specified by agency_timezone, even if the stops have their own stop_timezone values. In other words, if a given stop has a parent_station value, any stop_timezone value specified for that stop must be ignored.

Even if stop_timezone values are provided in stops.txt, the times in stop_times.txt should continue to be specified as time since midnight in the timezone specified by agency_timezone in agency.txt. This ensures that the time values in a trip always increase over the course of a trip, regardless of which timezones the trip crosses.</td></tr>
</table>

A.1.3 routes.txt

Field Name	Required	Details
route_id	Required	The **route_id** field contains an ID that uniquely identifies a route. The **route_id** is dataset unique.
agency_id	Optional	The **agency_id** field defines an agency for the specified route. This value is referenced from the agency.txt file. Use this field when you are providing data for routes from more than one agency.
route_short_name	Required	The **route_short_name** contains the short name of a route. This will often be a short, abstract

		identifier like "32", "100X", or "Green" that riders use to identify a route, but which doesn't give any indication of what places the route serves. If the route does not have a short name, please specify a **route_long_name** and use an empty string as the value for this field. See a Google Maps screenshot highlighting the **route_short_name**.
route_long_name	Required	The **route_long_name** contains the full name of a route. This name is generally more descriptive than the **route_short_name** and will often include the route's destination or stop. If the route does not have a long name, please specify a **route_short_name** and use an empty string as the value for this field. See a Google Maps screenshot highlighting the **route_long_name**.
route_desc	Optional	The **route_desc** field contains a description of a route. Please provide useful, quality information. Do not simply duplicate the name of the route. For example, "A trains operate between Inwood-207 St, Manhattan and Far Rockaway-Mott Avenue, Queens at all times. Also from about 6AM until about midnight, additional A trains operate between Inwood-207 St and Lefferts Boulevard (trains typically alternate between Lefferts Blvd and Far Rockaway)."
route_type	Required	The **route_type** field describes the type of transportation used on a route. Valid values for this field are: • **0** - Tram, Streetcar, Light rail. Any light rail or street level system within a metropolitan area. • **1** - Subway, Metro. Any underground rail system within a metropolitan area. • **2** - Rail. Used for intercity or long-distance travel. • **3** - Bus. Used for short- and long-distance bus routes. • **4** - Ferry. Used for short- and long-distance boat service. • **5** - Cable car. Used for street-level cable cars where the cable runs beneath the car. • **6** - Gondola, Suspended cable car. Typically used for aerial cable cars where the car is suspended from the cable. • **7** - Funicular. Any rail system designed for steep inclines. See a Google Maps screenshot highlighting the **route_type**.

route_url	Optional	The **route_url** field contains the URL of a web page about that particular route. This should be different from the **agency_url**. The value must be a fully qualified URL that includes **http://** or **https://**, and any special characters in the URL must be correctly escaped. See http://www.w3.org/Addressing/URL/4_URI_Recommentations.html for a description of how to create fully qualified URL values.
route_color	Optional	In systems that have colors assigned to routes, the **route_color** field defines a color that corresponds to a route. The color must be provided as a six-character hexadecimal number, for example, 00FFFF. If no color is specified, the default route color is white (FFFFFF). The color difference between **route_color** and **route_text_color** should provide sufficient contrast when viewed on a black and white screen. The W3C Techniques for Accessibility Evaluation And Repair Tools document offers a useful algorithm for evaluating color contrast. There are also helpful online tools for choosing contrasting colors, including the snook.ca Color Contrast Check application.
route_text_color	Optional	The **route_text_color** field can be used to specify a legible color to use for text drawn against a background of **route_color**. The color must be provided as a six-character hexadecimal number, for example, FFD700. If no color is specified, the default text color is black (000000). The color difference between **route_color** and **route_text_color** should provide sufficient contrast when viewed on a black and white screen.

A.1.4 trips.txt

Field Name	Required	Details
route_id	Required	The **route_id** field contains an ID that uniquely identifies a route. This value is referenced from the routes.txt file.
service_id	Required	The **service_id** contains an ID that uniquely identifies a set of dates when service is available for

		one or more routes. This value is referenced from the calendar.txt or calendar_dates.txt file.
trip_id	Required	The **trip_id** field contains an ID that identifies a trip. The **trip_id** is dataset unique.
trip_headsign	Optional	The **trip_headsign** field contains the text that appears on a sign that identifies the trip's destination to passengers. Use this field to distinguish between different patterns of service in the same route. If the headsign changes during a trip, you can override the **trip_headsign** by specifying values for the the stop_headsign field in stop_times.txt. See a Google Maps screenshot highlighting the **headsign**.
trip_short_name	Optional	The **trip_short_name** field contains the text that appears in schedules and sign boards to identify the trip to passengers, for example, to identify train numbers for commuter rail trips. If riders do not commonly rely on trip names, please leave this field blank. A **trip_short_name** value, if provided, should uniquely identify a trip within a service day; it should not be used for destination names or limited/express designations.

direction_id	Optional	The **direction_id** field contains a binary value that indicates the direction of travel for a trip. Use this field to distinguish between bi-directional trips with the same route_id. This field is not used in routing; it provides a way to separate trips by direction when publishing time tables. You can specify names for each direction with the **trip_headsign** field.
		
		• **0** - travel in one direction (e.g. outbound travel)
		• **1** - travel in the opposite direction (e.g. inbound travel)
		
		For example, you could use the trip_headsign and direction_id fields together to assign a name to travel in each direction on trip "1234", the trips.txt file would contain these rows for use in time tables:
		
		`trip_id, ... ,trip_headsign,direction_id`
		`1234, ... , to Airport,0`
		`1505, ... , to Downtown,1`

block_id	Optional	The **block_id** field identifies the block to which the trip belongs. A block consists of two or more sequential trips made using the same vehicle, where a passenger can transfer from one trip to the next just by staying in the vehicle. The **block_id** must be referenced by two or more trips in trips.txt.
shape_id	Optional	The **shape_id** field contains an ID that defines a shape for the trip. This value is referenced from the shapes.txt file. The shapes.txt file allows you to define how a line should be drawn on the map to represent a trip.

A.1.5 stop_times.txt

Field Name	Required	Details
trip_id	Required	The **trip_id** field contains an ID that identifies a trip. This value is referenced from the trips.txt file.
arrival_time	Required	The **arrival_time** specifies the arrival time at a specific stop for a specific trip on a route. The time is measured from "noon minus 12h" (effectively midnight, except for days on which daylight savings time changes occur) at the beginning of the service date. For times occurring after midnight on the service date, enter the time as a value greater than 24:00:00 in HH:MM:SS local time for the day on which the trip schedule begins. If you don't have separate times for arrival and departure at a stop, enter the same value for **arrival_time** and **departure_time**. You must specify arrival times for the first and last stops in a trip. If this stop isn't a time point, use an empty string value for the **arrival_time** and **departure_time** fields. Stops without arrival times will be scheduled based on the nearest preceding timed stop. To ensure accurate routing, please provide arrival and departure times for all stops that are time points. Do not interpolate stops. Times must be eight digits in HH:MM:SS format (H:MM:SS is also accepted, if the hour begins with 0). Do not pad times with spaces. The following columns list stop times for a trip and the

		proper way to express those times in the **arrival_time** field: **Time** **arrival_time value** 08:10:00 A.M. 08:10:00 or 8:10:00 01:05:00 P.M. 13:05:00 07:40:00 P.M. 19:40:00 01:55:00 A.M. 25:55:00 **Note:** Trips that span multiple dates will have stop times greater than **24:00:00**. For example, if a trip begins at 10:30:00 p.m. and ends at 2:15:00 a.m. on the following day, the stop times would be **22:30:00** and **26:15:00**. Entering those stop times as **22:30:00** and **02:15:00** would not produce the desired results.
departure_time	Required	The **departure_time** specifies the departure time from a specific stop for a specific trip on a route. The time is measured from "noon minus 12h" (effectively midnight, except for days on which daylight savings time changes occur) at the beginning of the service date. For times occurring after midnight on the service date, enter the time as a value greater than 24:00:00 in HH:MM:SS local time for the day on which the trip schedule begins. If you don't have separate times for arrival and departure at a stop, enter the same value for **arrival_time** and **departure_time**. You must specify departure times for the first and last stops in a trip. If this stop isn't a time point, use an empty string value for the **arrival_time** and **departure_time** fields. Stops without arrival times will be scheduled based on the nearest preceding timed stop. To ensure accurate routing, please provide arrival and departure times for all stops that are time points. Do not interpolate stops. Times must be eight digits in HH:MM:SS format (H:MM:SS is also accepted, if the hour begins with 0). Do not pad times with spaces. The following columns list stop times for a trip and the proper way to express those times in the **departure_time** field:

		Time **departure_time value** 08:10:00 A.M. 08:10:00 or 8:10:00 01:05:00 P.M. 13:05:00 07:40:00 P.M. 19:40:00 01:55:00 A.M. 25:55:00 **Note:** Trips that span multiple dates will have stop times greater than **24:00:00**. For example, if a trip begins at 10:30:00 p.m. and ends at 2:15:00 a.m. on the following day, the stop times would be **22:30:00** and **26:15:00**. Entering those stop times as **22:30:00** and **02:15:00** would not produce the desired results.
stop_id	Required	The **stop_id** field contains an ID that uniquely identifies a stop. Multiple routes may use the same stop. The stop_id is referenced from the stops.txt file. If location_type is used in stops.txt, all stops referenced in stop_times.txt must have location_type of 0. Where possible, **stop_id** values should remain consistent between feed updates. In other words, stop A with **stop_id 1** should have **stop_id 1** in all subsequent data updates. If a stop is not a time point, enter blank values for **arrival_time** and **departure_time**.
stop_sequence	Required	The **stop_sequence** field identifies the order of the stops for a particular trip. The values for **stop_sequence** must be non-negative integers, and they must increase along the trip. For example, the first stop on the trip could have a **stop_sequence** of **1**, the second stop on the trip could have a **stop_sequence** of **23**, the third stop could have a **stop_sequence** of **40**, and so on.
stop_headsign	Optional	The **stop_headsign** field contains the text that appears on a sign that identifies the trip's destination to passengers. Use this field to override the default trip_headsign when the headsign changes between stops. If this headsign is associated with an entire trip, use trip_headsign instead. See a Google Maps screenshot highlighting the **headsign**.

pickup_type	Optional	The **pickup_type** field indicates whether passengers are picked up at a stop as part of the normal schedule or whether a pickup at the stop is not available. This field also allows the transit agency to indicate that passengers must call the agency or notify the driver to arrange a pickup at a particular stop. Valid values for this field are: • **0** - Regularly scheduled pickup • **1** - No pickup available • **2** - Must phone agency to arrange pickup • **3** - Must coordinate with driver to arrange pickup The default value for this field is **0**.
drop_off_type	Optional	The **drop_off_type** field indicates whether passengers are dropped off at a stop as part of the normal schedule or whether a drop off at the stop is not available. This field also allows the transit agency to indicate that passengers must call the agency or notify the driver to arrange a drop off at a particular stop. Valid values for this field are: • **0** - Regularly scheduled drop off • **1** - No drop off available • **2** - Must phone agency to arrange drop off • **3** - Must coordinate with driver to arrange drop off The default value for this field is **0**.
shape_dist_traveled	Optional	When used in the stop_times.txt file, the **shape_dist_traveled** field positions a stop as a distance from the first shape point. The **shape_dist_traveled** field represents a real distance traveled along the route in units such as feet or kilometers. For example, if a bus travels a distance of 5.25 kilometers from the start of the shape to the stop, the **shape_dist_traveled** for the stop ID would be entered as "5.25". This information allows the trip planner to determine how much of the shape to draw when showing part of a trip on the map. The values used for **shape_dist_traveled** must increase along with **stop_sequence**: they cannot be used to show reverse travel along a route.

		The units used for **shape_dist_traveled** in the stop_times.txt file must match the units that are used for this field in the shapes.txt file.

A.1.6 calendar.txt

Field Name	Required	Details
service_id	Required	The **service_id** contains an ID that uniquely identifies a set of dates when service is available for one or more routes. Each service_id value can appear at most once in a calendar.txt file. This value is dataset unique. It is referenced by the trips.txt file.
monday	Required	The **monday** field contains a binary value that indicates whether the service is valid for all Mondays. • A value of **1** indicates that service is available for all Mondays in the date range. (The date range is specified using the **start_date** and **end_date** fields.) • A value of **0** indicates that service is not available on Mondays in the date range. **Note:** You may list exceptions for particular dates, such as holidays, in the calendar_dates.txt file.
tuesday	Required	The **tuesday** field contains a binary value that indicates whether the service is valid for all Tuesdays. • A value of **1** indicates that service is available for all Tuesdays in the date range. (The date range is specified using the **start_date** and **end_date** fields.) • A value of **0** indicates that service is not available on Tuesdays in the date range. **Note:** You may list exceptions for particular dates, such as holidays, in the calendar_dates.txt file.
wednesday	Required	The **wednesday** field contains a binary value that indicates whether the service is valid for all Wednesdays. • A value of **1** indicates that service is available for all Wednesdays in the date range. (The date

		range is specified using the **start_date** and **end_date** fields.) • A value of **0** indicates that service is not available on Wednesdays in the date range. **Note:** You may list exceptions for particular dates, such as holidays, in the calendar_dates.txt file.
thursday	Required	The **thursday** field contains a binary value that indicates whether the service is valid for all Thursdays. • A value of **1** indicates that service is available for all Thursdays in the date range. (The date range is specified using the **start_date** and **end_date** fields.) • A value of **0** indicates that service is not available on Thursdays in the date range. **Note:** You may list exceptions for particular dates, such as holidays, in the calendar_dates.txt file.
friday	Required	The **friday** field contains a binary value that indicates whether the service is valid for all Fridays. • A value of **1** indicates that service is available for all Fridays in the date range. (The date range is specified using the **start_date** and **end_date** fields.) • A value of **0** indicates that service is not available on Fridays in the date range. **Note:** You may list exceptions for particular dates, such as holidays, in the calendar_dates.txt file
saturday	Required	The **saturday** field contains a binary value that indicates whether the service is valid for all Saturdays. • A value of **1** indicates that service is available for all Saturdays in the date range. (The date range is specified using the **start_date** and **end_date** fields.) • A value of **0** indicates that service is not available on Saturdays in the date range. **Note:** You may list exceptions for particular dates, such as holidays, in the calendar_dates.txt file.
sunday	Required	The **sunday** field contains a binary value that indicates whether the service is valid for all Sundays.

		• A value of **1** indicates that service is available for all Sundays in the date range. (The date range is specified using the **start_date** and **end_date** fields.) • A value of **0** indicates that service is not available on Sundays in the date range. **Note:** You may list exceptions for particular dates, such as holidays, in the calendar_dates.txt file.
start_date	Required	The **start_date** field contains the start date for the service. The **start_date** field's value should be in YYYYMMDD format.
end_date	Required	The **end_date** field contains the end date for the service. This date is included in the service interval. The **end_date** field's value should be in YYYYMMDD format.

A.1.7 calendar_dates.txt

The calendar_dates table allows you to explicitly activate or disable service IDs by date. You can use it in two ways.

- Recommended: Use calendar_dates.txt in conjunction with calendar.txt, where calendar_dates.txt defines any exceptions to the default service categories defined in the **calendar.txt** file. If your service is generally regular, with a few changes on explicit dates (for example, to accomodate special event services, or a school schedule), this is a good approach.
- Alternate: Omit calendar.txt, and include ALL dates of service in calendar_dates.txt. If your schedule varies most days of the month, or you want to programmatically output service dates without specifying a normal weekly schedule, this approach may be preferable.

Field Name	Required	Details
service_id	Required	The **service_id** contains an ID that uniquely identifies a set of dates when a service exception is available for one or more routes. Each (service_id, date) pair can only appear once in calendar_dates.txt. If the a service_id value appears in both the calendar.txt and calendar_dates.txt files, the information in calendar_dates.txt modifies the service information specified in calendar.txt. This field is referenced by the trips.txt file.
date	Required	The **date** field specifies a particular date when service availability is different than the norm. You

		can use the **exception_type** field to indicate whether service is available on the specified date. The **date** field's value should be in YYYYMMDD format.
exception_type	Required	The **exception_type** indicates whether service is available on the date specified in the **date** field. • A value of **1** indicates that service has been added for the specified date. • A value of **2** indicates that service has been removed for the specified date. For example, suppose a route has one set of trips available on holidays and another set of trips available on all other days. You could have one **service_id** that corresponds to the regular service schedule and another **service_id** that corresponds to the holiday schedule. For a particular holiday, you would use the calendar_dates.txt file to add the holiday to the holiday **service_id** and to remove the holiday from the regular **service_id** schedule.

A.1.8 fare_attributes.txt

Field Name	Required	Details
fare_id	Required	The **fare_id** field contains an ID that uniquely identifies a fare class. The **fare_id** is dataset unique.
price	Required	The **price** field contains the fare price, in the unit specified by **currency_type**.
currency_type	Required	The **currency_type** field defines the currency used to pay the fare. Please use the ISO 4217 alphabetical currency codes which can be found at the following URL: http://www.iso.org/iso/en/prods-services/popstds/currencycodeslist.html.
payment_method	Required	The **payment_method** field indicates when the fare must be paid. Valid values for this field are: • **0** - Fare is paid on board. • **1** - Fare must be paid before boarding.
transfers	Required	The **transfers** field specifies the number of transfers permitted on this fare. Valid values for this

		field are: • **0** - No transfers permitted on this fare. • **1** - Passenger may transfer once. • **2** - Passenger may transfer twice. • **(empty)** - If this field is empty, unlimited transfers are permitted.
transfer_duration	Optional	The **transfer_duration** field specifies the length of time in seconds before a transfer expires. When used with a **transfers** value of 0, the **transfer_duration** field indicates how long a ticket is valid for a fare where no transfers are allowed. Unless you intend to use this field to indicate ticket validity, **transfer_duration** should be omitted or empty when **transfers** is set to 0.

A.1.9 fare_rules.txt

The fare_rules table allows you to specify how fares in fare_attributes.txt apply to an itinerary. Most fare structures use some combination of the following rules:

- Fare depends on origin or destination stations.
- Fare depends on which zones the itinerary passes through.
- Fare depends on which route the itinerary uses.

For examples that demonstrate how to specify a fare structure with fare_rules.txt and fare_attributes.txt, see FareExamples in the GoogleTransitDataFeed open source project wiki.

Field Name	Required	Details
fare_id	Required	The **fare_id** field contains an ID that uniquely identifies a fare class. This value is referenced from the fare_attributes.txt file.
route_id	Optional	The **route_id** field associates the fare ID with a route. Route IDs are referenced from the routes.txt file. If you have several routes with the same fare attributes, create a row in

		fare_rules.txt for each route.
		
		For example, if fare class "b" is valid on route "TSW" and "TSE", the fare_rules.txt file would contain these rows for the fare class:
		
		`b,TSW`
		`b,TSE`

origin_id	Optional	The **origin_id** field associates the fare ID with an origin zone ID. Zone IDs are referenced from the stops.txt file. If you have several origin IDs with the same fare attributes, create a row in fare_rules.txt for each origin ID.
		
		For example, if fare class "b" is valid for all travel originating from either zone "2" or zone "8", the fare_rules.txt file would contain these rows for the fare class:
		
		`b, , 2`
		`b, , 8`

destination_id	Optional	The **destination_id** field associates the fare ID with a destination zone ID. Zone IDs are referenced from the stops.txt file. If you have several destination IDs with the same fare attributes, create a row in fare_rules.txt for each destination ID.
		
		For example, you could use the origin_ID and destination_ID fields together to specify that fare class "b" is valid for travel between zones 3 and 4, and for travel between zones 3 and 5, the fare_rules.txt file would contain these rows for the fare class:
		
		`b, , 3,4`
		`b, , 3,5`

contains_id	Optional	The **contains_id** field associates the fare ID with a zone ID, referenced from the stops.txt file. The fare ID is then associated with itineraries that pass through every contains_id zone. For example, if fare class "c" is associated with all travel on the GRT route that passes through zones 5, 6, and 7 the fare_rules.txt would contain these rows:

		c,GRT,,,5 c,GRT,,,6 c,GRT,,,7 Because all contains_id zones must be matched for the fare to apply, an itinerary that passes through zones 5 and 6 but not zone 7 would not have fare class "c". For more detail, see FareExamples in the GoogleTransitDataFeed project wiki.

A.1.10 shapes.txt

Field Name	Required	Details
shape_id	Required	The **shape_id** field contains an ID that uniquely identifies a shape.
shape_pt_lat	Required	The **shape_pt_lat** field associates a shape point's latitude with a shape ID. The field value must be a valid WGS 84 latitude. Each row in shapes.txt represents a shape point in your shape definition. For example, if the shape "A_shp" has three points in its definition, the shapes.txt file might contain these rows to define the shape: A_shp,37.61956,-122.48161,0 A_shp,37.64430,-122.41070,6 A_shp,37.65863,-122.30839,11
shape_pt_lon	Required	The **shape_pt_lon** field associates a shape point's longitude with a shape ID. The field value must be a valid WGS 84 longitude value from -180 to 180. Each row in shapes.txt represents a shape point in your shape definition. For example, if the shape "A_shp" has three points in its definition, the shapes.txt file might contain these rows to define the shape: A_shp,37.61956,-122.48161,0

		A_shp,37.64430,-122.41070,6 A_shp,37.65863,-122.30839,11
shape_pt_sequence	Required	The **shape_pt_sequence** field associates the latitude and longitude of a shape point with its sequence order along the shape. The values for **shape_pt_sequence** must be non-negative integers, and they must increase along the trip. For example, if the shape "A_shp" has three points in its definition, the shapes.txt file might contain these rows to define the shape: A_shp,37.61956,-122.48161,0 A_shp,37.64430,-122.41070,6 A_shp,37.65863,-122.30839,11
shape_dist_traveled	Optional	When used in the shapes.txt file, the **shape_dist_traveled** field positions a shape point as a distance traveled along a shape from the first shape point. The **shape_dist_traveled** field represents a real distance traveled along the route in units such as feet or kilometers. This information allows the trip planner to determine how much of the shape to draw when showing part of a trip on the map. The values used for **shape_dist_traveled** must increase along with **shape_pt_sequence**: they cannot be used to show reverse travel along a route. The units used for **shape_dist_traveled** in the shapes.txt file must match the units that are used for this field in the stop_times.txt file. For example, if a bus travels along the three points defined above for A_shp, the additional **shape_dist_traveled** values (shown here in kilometers) would look like this: A_shp,37.61956,-122.48161,0,0 A_shp,37.64430,-122.41070,6,6.8310 A_shp,37.65863,-122.30839,11,15.8765

A.1.11 frequencies.txt

This table is intended to represent schedules that don't have a fixed list of stop times. When trips are defined in frequencies.txt, the trip planner ignores the absolute values of the arrival_time and departure_time fields for those trips in stop_times.txt. Instead, the stop_times table defines the sequence of stops and the time difference between each stop.

Field Name	Required	Details
trip_id	Required	The **trip_id** contains an ID that identifies a trip on which the specified frequency of service applies. Trip IDs are referenced from the trips.txt file.
start_time	Required	The **start_time** field specifies the time at which service begins with the specified frequency. The time is measured from "noon minus 12h" (effectively midnight, except for days on which daylight savings time changes occur) at the beginning of the service date. For times occurring after midnight, enter the time as a value greater than 24:00:00 in HH:MM:SS local time for the day on which the trip schedule begins. E.g. 25:35:00.
end_time	Required	The **end_time** field indicates the time at which service changes to a different frequency (or ceases) at the first stop in the trip. The time is measured from "noon minus 12h" (effectively midnight, except for days on which daylight savings time changes occur) at the beginning of the service date. For times occurring after midnight, enter the time as a value greater than 24:00:00 in HH:MM:SS local time for the day on which the trip schedule begins. E.g. 25:35:00.
headway_secs	Required	The **headway_secs** field indicates the time between departures from the same stop (headway) for this trip type, during the time interval specified by **start_time** and **end_time**. The headway value must be entered in seconds. Periods in which headways are defined (the rows in frequencies.txt) shouldn't overlap for the same trip, since it's hard to determine what should be inferred from two overlapping headways. However, a headway period may begin at the exact same time that another one ends, for instance: `A, 05:00:00, 07:00:00, 600`

		B, 07:00:00, 12:00:00, 1200
exact_times	Optional	The **exact_times** field determines if frequency-based trips should be exactly scheduled based on the specified headway information. Valid values for this field are: • **0** or **(empty)** - Frequency-based trips are not exactly scheduled. This is the default behavior. • **1** - Frequency-based trips are exactly scheduled. For a frequencies.txt row, trips are scheduled starting with `trip_start_time = start_time + x * headway_secs` for all `x` in `(0, 1, 2, ...)` where `trip_start_time < end_time`. The value of **exact_times** must be the same for all frequencies.txt rows with the same **trip_id**. If **exact_times** is **1** and a frequencies.txt row has a **start_time** equal to **end_time**, no trip must be scheduled. When **exact_times** is **1**, care must be taken to choose an **end_time** value that is greater than the last desired trip start time but less than the last desired trip start time + **headway_secs**.

transfers.txt

Trip planners normally calculate transfer points based on the relative proximity of stops in each route. For potentially ambiguous stop pairs, or transfers where you want to specify a particular choice, use transfers.txt to define additional rules for making connections between routes.

Field Name	Required	Details
from_stop_id	Required	The **from_stop_id** field contains a stop ID that identifies a stop or station where a connection between routes begins. Stop IDs are referenced from the stops.txt file. If the stop ID refers to a station that contains multiple stops, this transfer rule applies to all stops in that station.
to_stop_id	Required	The **to_stop_id** field contains a stop ID that identifies a stop or station where a connection between routes ends. Stop IDs are referenced from the stops.txt file. If the stop ID refers to a station that contains multiple stops, this transfer rule applies to all stops in that station.
transfer_type	Required	The **transfer_type** field specifies the type of connection for the specified (from_stop_id,

		to_stop_id) pair. Valid values for this field are: • **0** or **(empty)** - This is a recommended transfer point between two routes. • **1** - This is a timed transfer point between two routes. The departing vehicle is expected to wait for the arriving one, with sufficient time for a passenger to transfer between routes. • **2** - This transfer requires a minimum amount of time between arrival and departure to ensure a connection. The time required to transfer is specified by **min_transfer_time**. • **3** - Transfers are not possible between routes at this location.
min_transfer_time	Optional	When a connection between routes requires an amount of time between arrival and departure (transfer_type=2), the **min_transfer_time** field defines the amount of time that must be available in an itinerary to permit a transfer between routes at these stops. The min_transfer_time must be sufficient to permit a typical rider to move between the two stops, including buffer time to allow for schedule variance on each route. The min_transfer_time value must be entered in seconds, and must be a non-negative integer.

A.1.12 feed_info.txt

The file contains information about the feed itself, rather than the services that the feed describes. GTFS currently has an agency.txt file to provide information about the agencies that operate the services described by the feed. However, the publisher of the feed is sometimes a different entity than any of the agencies (in the case of regional aggregators). In addition, there are some fields that are really feed-wide settings, rather than agency-wide.

Field Name	Required	Details
feed_publisher_name	Required	The **feed_publisher_name** field contains the full name of the organisation that publishes the feed. (This may be the same as one of the **agency_name** values in agency.txt.) GTFS-consuming applications can display this name when giving attribution for a particular feed's data.
feed_publisher_url	Required	The **feed_publisher_url** field contains the URL of the feed publishing organisation's website. (This may be the same as one of the **agency_url** values in agency.txt.) The value must be a fully qualified URL that

		includes **http://** or **https://**, and any special characters in the URL must be correctly escaped. See http://www.w3.org/Addressing/URL/4_URI_Recommentations.html for a description of how to create fully qualified URL values.
feed_lang	Required	The **feed_lang** field contains a IETF BCP 47 language code specifying the default language used for the text in this feed. This setting helps GTFS consumers choose capitalisation rules and other language-specific settings for the feed. For an introduction to IETF BCP 47, please refer to http://www.rfc-editor.org/rfc/bcp/bcp47.txt and http://www.w3.org/International/articles/language-tags/.
feed_start_date feed_end_date	Optional	The feed provides complete and reliable schedule information for service in the period from the beginning of the **feed_start_date** day to the end of the **feed_end_date** day. Both days are given as dates in YYYYDDMM format as for calendar.txt, or left empty if unavailable. The **feed_end_date** date must not precede the **feed_start_date** date if both are given. Feed providers are encouraged to give schedule data outside this period to advise of likely future service, but feed consumers should treat it mindful of its non-authoritative status. If **feed_start_date** or **feed_end_date** extend beyond the active calendar dates defined in calendar.txt and calendar_dates.txt, the feed is making an explicit assertion that there is no service for dates within the **feed_start_date** or **feed_end_date** range but not included in the active calendar dates.
feed_version	Optional	The feed publisher can specify a string here that indicates the current version of their GTFS feed. GTFS-consuming applications can display this value to help feed publishers determine whether the latest version of their feed has been incorporated.

A.2 Annex II: HAFAS Interface

A.2.1 Introduction

The public interface is implemented as a ReST1 (Representational State Transfer) interface which provides different methods for the different functionalities of the journey planner, which are the following services:

- Location
- Trip
- DepartureBoard
- ArrivalBoard
- JourneyDetail
- Geometry

While Location, Trip, ArrvialBoard and DepartureBoard can be called directly the JourneyDetail-Method can only be called by a reference given in a result of the Trip, or DepartureBoard services. Geometry-Method can only be called by a reference given in a result of the Trip or JourneyDetail request.

The system only implements read-only GET requests which are called by given service URLs and multiple GET parameters to specify the requested journey planner information. The parameter values need to be in ISO-8859-1 URL encoded. The result of each request will be delivered either as XML or JSON (see 1.2.7) response. If the encoding of URL parameters is not right, the behaviour of the system might deliver unexpected results.

From now on it is assumed, that you have been provided with a base URL of the HAFAS system. The following documentation of the different requests are described based on this given base url <baseurl>.

A.2.2 General principles

There are some general principles which are valid for the different services which are described in this section.

A.2.3 Coordinates

Coordinates are always in the WGS84 system, represented as decimal degrees in the interval -90 to 90 for the latitude (lat) and -180 to 180 for the longitude (long).

A.2.4 Date and time formats

Dates are always represented in the format YYYY-MM-DD. This applies both for request parameters as for dates in responses. Times are always represented in the format HH:MM in 24h nomenclature.

A.2.5 Stateless service vs. data dependency

All services of the provided interface are stateless as it is required for a ReST protocol. But this has its limitation concerning the journey planner's timetable data. As soon as the timetable data is exchanged (in most cases daily on weekdays), IDs of stops/stations are not necessary valid anymore. The same applies for reference URLs to retrieve journeyDetails. The storage of stop/station IDs and reference URLs to journeyDetails for a longer period except the current user session is not recommended therefore and can only be done on own risk for undetermined behaviour when reusing these IDs or references.

A.2.6 Route index

A route is the list of stops/stations where a vehicle like a train or bus stops. Every stop/station on a route has its own index which can be used as a reference. This index is also used to identify distinctively if the same stop/station if it is contained several times in one route.

A.2.7 Real time information

Real time information will be included in the service as far as it is available in the web based journey planner. It is always delivered in addition to the planned departures and arrivals.

A.2.8 Versioning

Due to enhancements of the API the input parameters and the results can change over time. Different Versions of the API will be available at the same time.

The requested version can be specified by using the version number in the path info:

http://<baseUrl>/<version>/<servicename>

The version part is optional, if it is omitted, the latest version will be used. Be aware that omitting the version can break your client when a new API version is introduced. If your client should always use a special version of the api (v2 for example), your url would look like this: http://<baseUrl>/v2/<servicename>

A.2.9 Response Format

The interface returns responses either in XML (default) or JSON format.

In order to request a JSON response you have to append the following parameter to each call of the interface: format=json. If JSONP is needed you can append an additional parameter to specify the name of callback function, the JSON object will be wrapped by a function call with this name: jsonpCallback=mycallback.

The JSON content is generated by converting the xml content to JSON automatically. The conversion is done by the following simple rules:

- Element names become object properties
- Text (PCDATA) becomes an object property with name "$" <a>foo becomes { "a": { "$" : "foo" } }
- Nested elements become nested properties <a>foo<c>foo</c> becomes { "a": { "b" : { "$": "foo" }, "c": { "$": "foo"} } }
- If there are multiple elements with the same name the JSON code contains an array for these element. <a>foo1foo2 becomes { "a": { "b" : [{"$": foo1" }, {"$": "foo2" }] } }
- Attribute names become object properties foo2 becomes { "a": { "atb" : "foo1", "$" : "foo2" } }

A.2.10 Authentification

Every client using the API needs to pass a valid authentification key in every request.
The following parameter has to be appended to the url: authKey=<your_key_here>.

Please contact Västtrafik in order to request an authentification key.

A.2.11 Services

The following sub-sections give an overview of the most prominent services.

A.2.11.1 Location Service

There are 3 different types of the location service which can be used to get a list of locations using different input parameters.
The response format for all services is defined in hafasRestLocation.xsd (see also 1.4.1 for further details).

A.2.11.2 Location.name Service

The location.name service can be used to perform a pattern matching of a user input and to retrieve a list of possible matches in the journey planner database. Possible matches might be stops/stations, points of interest and addresses. For reasons of backward compatibility the service name location can be used as an alias for location.name.

The service has only one GET parameter which is called input. This parameter contains a string with the user input. The result is a list of possible matches (locations) where the user might pick one entry to perform a trip request with this location as origin or destination or to ask for a departure board or arrival board of this location (stops/stations only) .

A.2.11.3 Location.allstops Service

The location.allstops service returns a list of all stops available in the journey planner. Be aware that a call of this service is very time consuming and should be only requested when it is really needed.

The URL to call the service is the following:

http://<baseurl>/location.allstops

A.2.11.4 Location.nearbystops Service

The location.nearbystops service returns a list of stops around a given center coordinate (within a radius of 1000m). The returned results are ordered by their distance to the center coordinate.
Possible parameters:

Name	Use	Range	Default	Description
originCoordLat	Mandatory	See 1.2.1	-	Latitude of center coordinate
originCoordLong	Mandatory	See 1.2.1	-	Longitude of center coordinate
maxNo	Optional	1-1000	10	Maximum number of returned stops

The URL to call the service is the following:
http://<baseurl>/location.nearbystops?originCoordLong=11.981211&originCoordLat=57.709792&maxNo=5

A.2.11.5 Trip service

The trip service calculates a trip from a specified origin to a specified destination. These might be stop/station IDs or coordinates based on addresses and points of interest validated by the location service or coordinates freely defined by the client.

Both origin and destination are mandatory parameters for the trip service.

The parameters are named either originId or originCoordLat, originCoordLong, and originCoordName. For the destination the parameters are named either destId or destCoordLat, destCoordLong and destCoordName. The origin/dest-CoordName parameters are the names of the address at the specified coordinate.

It is possible to define a via stop/station. This forces the journey planner to search for trips which pass the defined station. The parameter is called viaId.

The departure time and date are defined with the parameters date and time. If the date is not set the current date will be used (server time). If the parameter time is not set the current server time will be used to perform the request.
To specify that the given time and date is not the departure time but the latest time to arrive at the destination you can use the parameter searchForArrival=1.
It is possible to switch off specific means of transport by using one of the following optional parameters:

- useVas=0 // Vasttågen
- useLDTrain=0 // Long Distance Trains
- useRegTrain=0 // Regional Trains
- useBus=0
- useBoat=0
- useTram=0

The default value is that all means of transport are switched on (value 1). If no parameter is set this default value applies.

In addition to the flags above you can use excludeDR=1 to exclude journeys which require tel. registration, by default they are included.

As a response the service will return a result according to hafasRestDepartureBoard.xsd. This will contain a list of departures with train/line number, type of transport, departure times (incl. real time), departure stop/stations (might be different from requested stop), direction text and a track information if available. Every departure will also contain a reference to the journey detail service.

In addition to departure boards the service arrivalBoard delivers arriving journeys at a specified stop. The parameters are identical to the parameters of the departureBoard service.
As a response the service will return a result according to hafasRestArrivalBoard.xsd. This will contain a list of arrival with train/line number, type of transport, arrival times (incl. real time), departure stop/stations (might be different from requested stop), the name of the origin stop and a track information if available. Every arrival will also contain a reference to the journey detail service.

A.2.11.6 Journey detail service

The journeyDetail service will deliver information about the complete route of a vehicle. This service can’t be called directly but only by reference URLs in a result of a trip or departureBoard request. It contains a list of all stops/stations of this journey including all departure and arrival times (with real time data if available) and additional information like specific attributes about facilities and other texts.

The response will be returned a result according to the format described in hafasRestJourneyDetails.xsd.

A.2.11.7 Geometry service

The Geometry service will return the polyline for a leg. This service cannot be called directly but only by reference URLs in a result of a trip or JourneyDetail request.

The result contains a list of WGS84 coordinates which can be used to display the polyline on a map.

The response will be returned as a result according to the format described in hafasRestGeometry.xsd

A.3 Annex III: TDB structure with FleetManagementStandard (FMS) implementation

A.3.1 IMP_HEADER

Field descriptions:

Fieldname	Typ	Pri-mary	Mandatory	Version	Description	Definition
IMP_REFERENCE						
IMP_OBJECT_TYPE	NUMERIC(2)		X	1.0.5	Type of data to be imported.	1=Order 2=Location 3=Customer 4=Order Process Status 5=Provider 6=Driver 12=Position data 13=ETA 14=FMS 15=Tour 16=Tour stop 17=Tour order
IMP_EXT_ID	VARCHAR(255)		X		ID of the object type specified in the header	
IMP_ACTION_CODE	NUMERIC(2)		X		Action code	1=Create new 2=Modification 3=Cancellation
IMP_CREATION_TIME	VARCHAR(17)		X	1.0.5	Time stamp from the facility in the transfer DB	YYYYMMDDhhmmssSSS
IMP_PROCESS_TIME	VARCHAR(17)		X	1.0.5	Time stamp from the processing in the transfer DB	YYYYMMDDhhmmssSSS
IMP_PROCESS_CODE			X			0=Initial (set by host) 1=Import successful -1=Import failed

IMP_PROCESS_RETRIES						

A.3.2 IMP_TOUR_STATUS

The IMP_EXT_ID of the dependend IMP_HEADER is the external ID of the tour.

Field descriptions:

Fieldname	Typ	Pri-mary	Mandatory	Version	Description	Definition
TOUR_IMP_REFERENCE	NUMERIC(9)	X	X	2.1.2	Unique identification of the import record	
TOUR_VEH_EXTID	VARCHAR(255)		X	2.1.4	Unique identification of the vehicle (licenceplate)	
TOUR_EXP_REFERENCE	NUMERIC(9)	X	X	2.1.2	Unique identification of the export	
TOUR_STAT_CODE	NUMERIC(2)		X	2.1.2	Status Code	0 = undefined 1 = started 2 = canceled 3 = finished 4 = accepted 5 = denied
TOUR_STAT_CREATION	VARCHAR(17)		X	2.1.2	Time stamp when the OnBoardUnit creates the state	YYYYMMDDhhmmssSSS
TOUR_COMMENT	VARCHAR(100)			2.1.2	Textual description associated with the column TO_ STAT_CODE	

A.3.3 IMP_TOURSTOP_STATUS

The IMP_EXT_ID of the dependend IMP_HEADER is the external ID of the tour.

Field descriptions:

Fieldname	Typ	Pri-mary	Mand atory	Version	Description	Definition
TOST_IMP_REFERENCE	NUMERIC(9)	X	X	2.1.2	Unique identification of the import record	
TOST_EXTID	VARCHAR(255)		x	2.1.2		
TOST_EXP_REFERENCE	NUMERIC(9)	X	X	2.1.2	Unique identification of the export	
TOST_STAT_CODE	NUMERIC(2)		X	2.1.2	Status Code	0 = undefined 1 = arrived 2 = departed
TOST_STAT_CREATION	VARCHAR(17)		X	2.1.2	Time stamp when the OnBoardUnit creates the state	YYYYMMDDhhmmssSSS
TOST_COMMENT	VARCHAR(100)			2.1.2	Textual description associated with the column TOST_ STAT_CODE	
TOST_VEH_EXTID	VARCHAR(255)		X	2.1.4	Unique identification of the vehicle (licenceplate)	

A.3.4 IMP_TOURORDER_STATUS

The IMP_EXT_ID of the dependend IMP_HEADER is the external ID of the tour.

Field descriptions:

Fieldname	Typ	Pri-mary	Mand atory	Version	Description	Definition
TOOR_IMP_REFERENCE	NUMERIC(9)	X	X	2.1.2	Unique identification of the import record	
TOOR_EXTID	VARCHAR(255)		x	2.1.2		
TOOR_EXP_REFERENCE	NUMERIC(9)	X	X	2.1.2	Unique identification of the export	

TOOR_STAT_CODE	NUMERIC(2)		X	2.1.2	Status Code	0 = undefined 1 = finished 2 = denied 3 = customer not reached 4 = damaged
TOOR_STAT_CREATION	VARCHAR(17)		X	2.1.2	Time stamp when the OnBoardUnit created the state	YYYYMMDDhhmmssSSS
TOOR_COUNT	NUMERIC(4)		X	2.1.6	Approach number	
TOOR_COMMENT	VARCHAR(100)			2.1.2	Textual description associated with the column TO_ STAT_CODE	
TOOR_VEH_EXTID	VARCHAR(255)		X	2.1.4	Unique identification of the vehicle (licenceplate)	

A.3.5 IMP_POSITION

To each header entry multiple IMP_POSITION entries are possible. The IMP_EXT_ID of the dependend IMP_HEADER is the external ID of the vehicle.

Field descriptions:

Fieldname	Typ	Pri-mary	Mandatory	Version	Description	Definition
POS_IMP_REFERENCE	NUMERIC(9)	X	X	2.1.2	Unique identification of the import record	
POS_TYPE	NUMERIC(1)		X		If POS_TYPE = NMEA only the fields POS_NMEA…, POS_CONTEXT and POS_TOUR_… where used	1 = POS 2 = NMEA
POS_NMEA_TYPE	NUMERIC(1)			2.1.2	Only RMC and GGA are supported	1 = RMC 2 = GGA
POS_NMEA_CONTENT	VARCHAR(100)			2.1.2	NMEA information delivered from GPS	
POS_COORD_TYPE	NUMERIC(2)		X	2.1.2	Coordination format	0 = Mercator 1 = Geominsec 2 = Geodecimal (prefered)

POS_COORD_X	NUMERIC(9)		X	2.1.2	x-coordinate, longitude	
POS_COORD_Y	NUMERIC(9)		X	2.1.2	y-coordinate, latitude	
POS_CREATION_TIME	VARCHAR(17)		X	2.1.2	Time stamp when the OnBoardUnit created the position message.	YYYYMMDDhhmmssSSS
POS_GPS_TIME	VARCHAR(17)		X	2.1.2	Time stamp of the GPS position.	YYYYMMDDhhmmssSSS
POS_STATUS	NUMERIC(3)			2.1.2	For later use	
POS_STATUS_TEXT	VARCHAR(30)			2.1.2	For later use	
POS_CONTEXT	NUMERIC(1)			2.1.2		0 = free (default) 1 = Tour
POS_TOUR_EXTID	VARCHAR(255)			2.1.2	The external ID of the tour, to which this message belongs.	
POS_TOUR_REFERENCE	NUMERIC(9)			2.1.2	Unique identification of the **export** record	Subject of change. Perhaps we remove this parameter.

A.3.6 IMP_ETA

The main content in the transfer DB content IMP_OBJECT_TYPE 13 is the TABLE IMP_ETA (1), optional it is possible to provide the dependend position (IMP_POSITION).

The IMP_EXT_ID of the dependend IMP_HEADER is the defined external ID of the vehicle.

Field descriptions:

Fieldname	Typ	Pri-mary	Mandatory	Version	Description	Definition
ETA_IMP_REFERENCE	NUMERIC(9)	X	X	2.1.2	Unique identification of the import record	
ETA_TOUR_EXTID	VARCHAR(255)		X	2.1.2	The external ID of the tour, to which this message belongs.	
ETA_TOUR_REFERENCE	NUMERIC(9)		X	2.1.2	Unique identification of the **export** record	Subject of change. Perhaps we remove this parameter.
ETA_TOURSTOP_EXTID	VARCHAR(255)		X	2.1.2	The id of the location (TS_LOC_EXTID)	

ETA_ORDER_EXTID	VARCHAR(255)		X	2.1.2		
ETA_CREATION_TIME	VARCHAR(17)		X	2.1.2	Time stamp when the OnBoardUnit created the ETA	YYYYMMDDhhmmssSSS
ETA_TIME	VARCHAR(12)		X	2.1.2	Estimated time of arrival calculated from the OnBoardUnit.	YYYYMMDDhhmm

A.3.7 IMP_FMS

The main content in the transfer DB content IMP_OBJECT_TYPE 14 is the TABLE IMP_FMS (1), optional it is possible to provide the dependend position (IMP_POSITION).

The IMP_EXT_ID of the dependend IMP_HEADER is the defined external ID of the vehicle or driver.

Field descriptions:

Fieldname	Typ	Pri-mary	Mandatory	Version	Description	Definition
FMS_IMP_REFERENCE	NUMERIC(9)	X	X	2.1.2	Unique identification of the import record	
FMS_CREATION_TIME	VARCHAR(17)		X	2.1.3	Time stamp when the OnBoardUnit created the FMS	YYYYMMDDhhmmssSSS

FMS_MSG_TYPE	NUMERIC(2)		X	2.1.2		1 = Cruise Control/Vehicle Speed: CCVS 2 = Electronic Engine Controller #2: EEC2 3 = Fuel Consumption: LFC 4 = Dash Display: DD 5 = Electronic Engine Controller #1: EEC1 6 = Vehicle Weight: VW 7 = Engine Hours, Revolutions: HOURS 8 = Vehicle Identification: VI 9 = FMS-standard Interface: FMS 10 = High Resolution Vehicle Distance: VDHR 11 = Service Information: SERV 12 = Tachograph : TCO1 13 = Engine Temperature 1: ET1 14 = Ambient Conditions: AMB 15 = Driver's Identification: DI 16 = Fuel Economy: LFE 17 = PTO Drive Engagement: PTODE 18 = High Resolution Fuel Consumption (Liquid): HRLFC
FMS_DATA_BYTE_1	VARCHAR(8)			2.1.2	Data Byte 1	
FMS_DATA_BYTE_2	VARCHAR(8)			2.1.2	Data Byte 2	
FMS_DATA_BYTE_3	VARCHAR(8)			2.1.2	Data Byte 3	
FMS_DATA_BYTE_4	VARCHAR(8)			2.1.2	Data Byte 4	
FMS_DATA_BYTE_5	VARCHAR(8)			2.1.2	Data Byte 5	
FMS_DATA_BYTE_6	VARCHAR(8)			2.1.2	Data Byte 6	
FMS_DATA_BYTE_7	VARCHAR(8)			2.1.2	Data Byte 7	
FMS_DATA_BYTE_8	VARCHAR(8)			2.1.2	Data Byte 8	
FMS_ADD_INFO_1	VARCHAR(32)			2.1.2	Additional informations	
FMS_ADD_INFO_2	VARCHAR(32)			2.1.2	Additional informations	
FMS_VERSION	VARCHAR(32)			2.1.2	Versionsinformation FMS	