An automated framework to derive model variables from open transport data using R, PostgreSQL and OpenTripPlanner

Marcus Young*[1]

[1]Transportation Research Group, University of Southampton

March 11, 2016

Summary

This paper outlines a framework that utilises open source software tools to automatically generate the explanatory variables needed for certain transport models, such as mode choice or station choice, from disparate sources of open transport data. A key component of the framework is OpenTripPlanner, an open source multi-modal route planner that builds transport networks using OpenStreetMap, GTFS transit feeds, and digital elevation model data. A case study, applying the framework to generate data for a multinomial logit station choice model, is also described.

KEYWORDS: open data, public transport, open source, OpenTripPlanner

1 Introduction

When developing certain types of transport model, such as mode choice, data are needed for a range of attributes relating to the journey between each origin and destination pair for each transport mode, such as in-vehicle time, walk time, wait time, cost, and number of interchanges. Some models also require non-journey related attributes, for example a railway station choice model will need measures of station services and facilities. In the UK a large amount of public transport data is available under various open data initiatives, but it is supplied by different organisations in different formats and requires specialist tools to derive useful information from it. A key requirement is a multi-modal network, that can generate routes involving both motorised and non-motorised modes. A possible solution is to utilise commercial web services, such as Google Maps (Google, 2016b) or TransportAPI (Placr Limited, 2016), but these limit free API calls and may impose usage conditions[1]. They also only use current published timetables, meaning that the impact of service changes, such as new routes, changed frequencies or new stops, cannot be evaluated. Commercial software, such as Visography TRACC, can import and analyse UK public transport data, but this

*m.a.young@soton.ac.uk

[1] Google typically limits the number of API calls from an IP address to 2,500 per day (Google, 2016a)

is of little benefit to a researcher without access to such software, and an open source alternative is therefore preferable. This paper will outline a framework that utilises open source tools to bring together transport data from disparate sources and generate the variables required as model inputs. A case study, applying the framework to generate data for a station choice model, is then described.

2 The framework

In its basic form, the framework consists of a PostgreSQL database, the OpenTripPlanner (OTP) route planner (OpenTripPlanner, 2015), and the R software environment. The framework has the potential to automatically populate an origin:destination table with attributes obtained from internal and external data sources. The components and how they interact are illustrated in Figure 1, and described below.

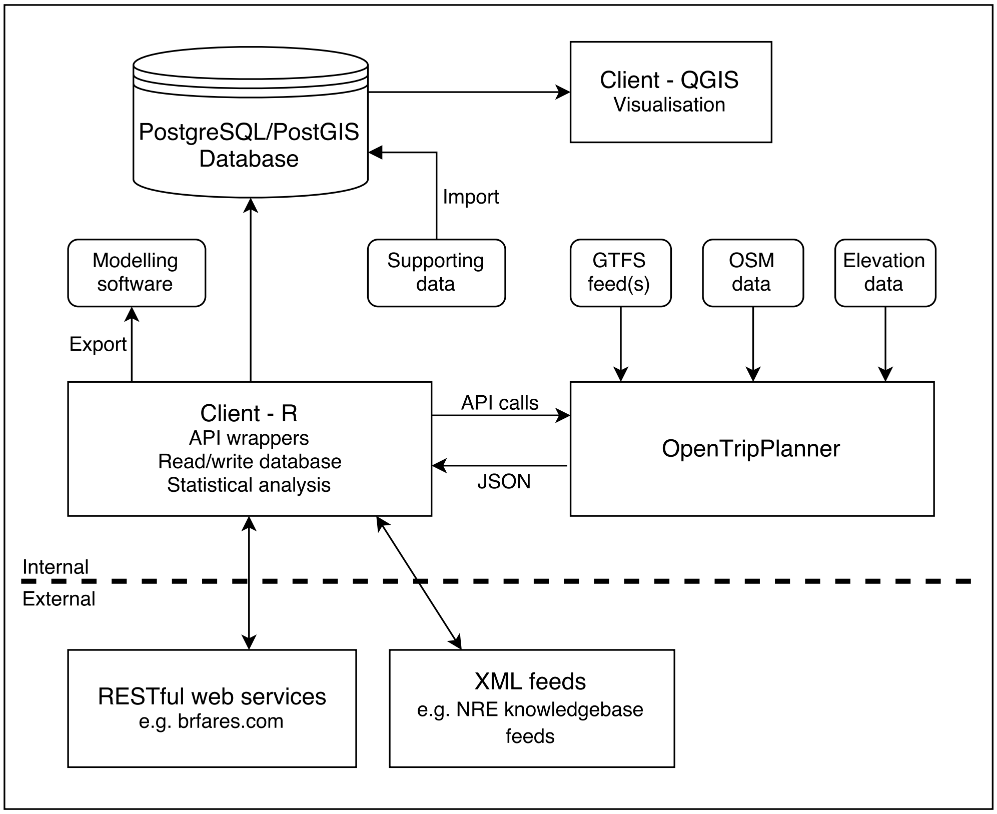


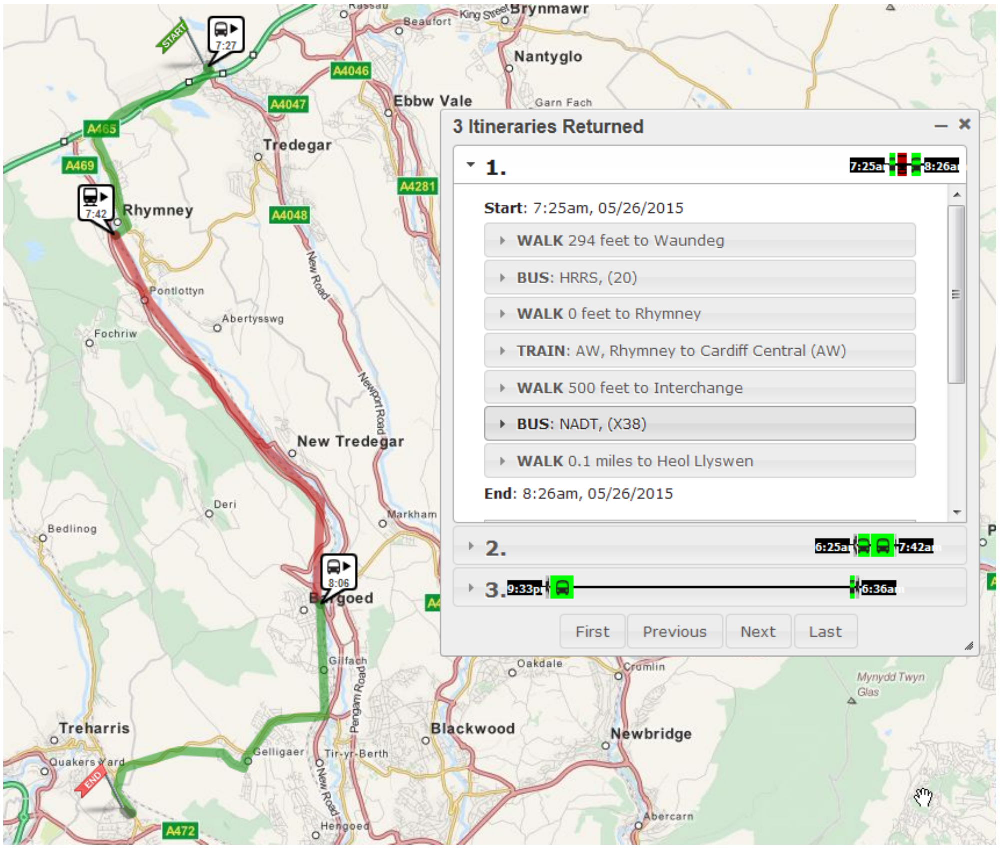
Figure 1: Framework to derive explanatory variables from disparate open transport data sources.

2.1 PostgreSQL database schema

The database schema adopted will depend on the nature of the research study, but as a minimum it will include a table containing a row for each unique origin:destination pair along with their respective latitude and longitude coordinates. A more efficient solution would be to store these coordinates in a separate table, for example by importing the Ordnance Survey (OS) Code-Point dataset.

2.2 OTP instance

OTP is an open-source and cross-platform multi-modal route planner written in JAVA. It uses imported Open Street Map (OSM) data for routing on the street and path network and supports multi-agency public transport routing through imported GTFS feeds. It can also apply a digital elevation model to the OSM street network, allowing, for example, cycle-friendly routes to be requested. OTP has a web front-end (see Figure 2) that can be used by end-users, and a sophisticated routing API.



3 Itineraries Returned

1.

Start: 7:25am, 05/26/2015

- **WALK** 294 feet to Waundeg
- **BUS**: HRRS, (20)
- **WALK** 0 feet to Rhymney
- **TRAIN**: AW, Rhymney to Cardiff Central (AW)
- **WALK** 500 feet to Interchange
- **BUS**: NADT, (X38)
- **WALK** 0.1 miles to Heol Llyswen

End: 8:26am, 05/26/2015

2.

3.

Figure 2: The OTP web interface, with example walk, bus and train trip itinerary.

2.3 R software environment

The R software environment is the hub of the framework. A set of functions have been developed to query the OTP API and process the JSON response. These are the beginnings of an API wrapper which could be released as an R package in the future. R is able to read from and write to the database by sending queries using the RPostgreSQL package (Conway et al., 2013). The steps required in a typical R script to populate an origin-destination table are illustrated in Figure 3.

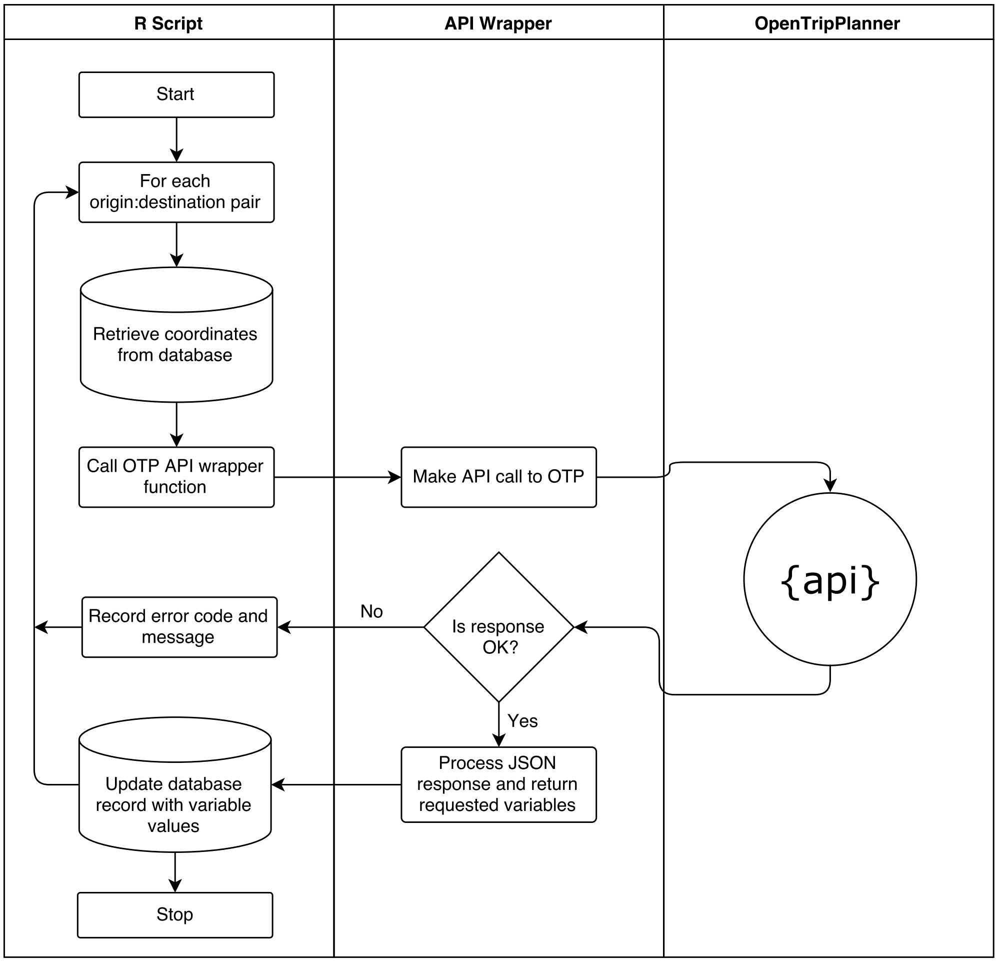

R Script
API Wrapper
OpenTripPlanner
Start
For each origin:destination pair
Retrieve coordinates from database
Call OTP API wrapper function
Make API call to OTP
{api}
Record error code and message
No
Is response OK?
Yes
Process JSON response and return requested variables
Update database record with variable values
Stop

Figure 3: The steps in a typical R script to populate an origin-destination table using data from OTP.

Data from external web services can be accommodated through appropriate API wrappers or feed parsers. Examples that have been incorporated include the BR Fares website (BR Fares Ltd, 2016), and the National Rail Enquiries Knowledgebase XML feeds (National Rail Enquiries, 2016).

3 Case study

The framework was used to derive a range of explanatory variables for input into a multinomial logit station choice model estimation, and to derive variables at the unit postcode level when the model was applied to generate example probabilistic station catchments. The variables measured aspects of the access journey, facilities at stations and the train leg. The revealed preference data for the model was obtained from an on-train survey carried out in South Wales which recorded for each individual the trip origin unit postcode, the access station, and the egress station (Blainey, 2009)[2]. The background to this work, calibration results, and example application of the model are reported in Young and Blainey (2016). This case study will consider the database schema, the OTP build process, and some issues with obtaining trip itineraries from OTP[3].

3.1 Database schema

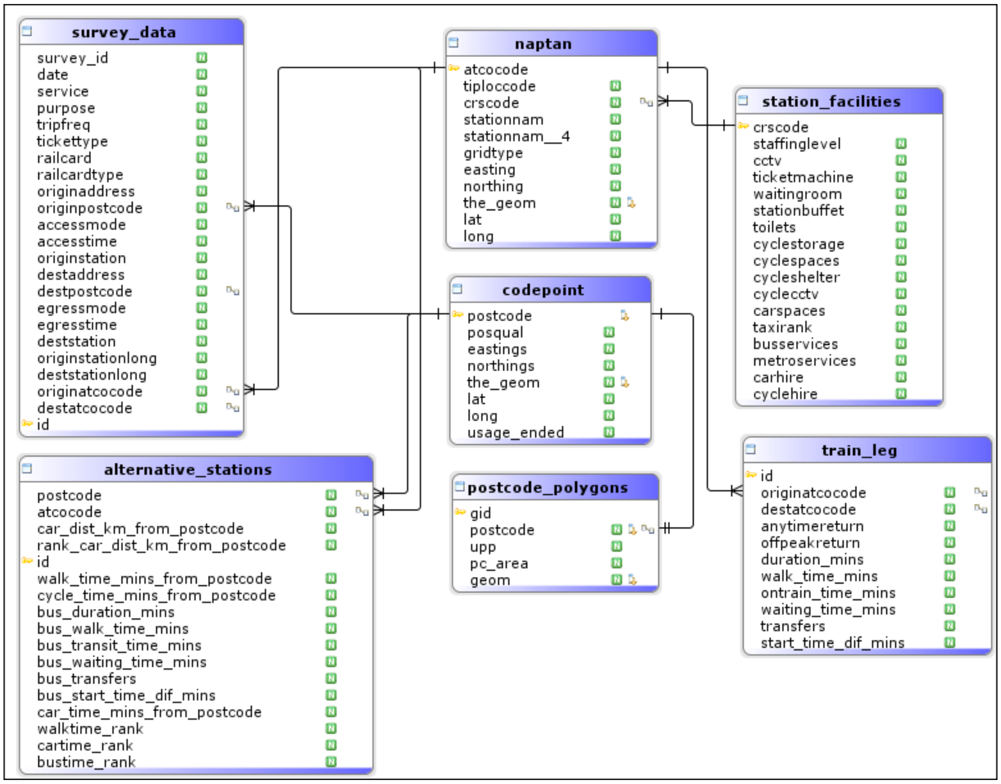


Figure 4: Key tables in the PostgreSQL database schema for the case study.

[2] This survey was carried out for a different purpose and Blainey (2009) did not use the data to estimate station choice models.

[3] This paper complements Young and Blainey (2016). In that paper the technical details of the framework and its application were only briefly described.

The schema adopted for this study, which is illustrated in Figure 4, included the following key tables:

- survey_data - imported survey data
- naptan - station locations from the NaPTAN database
- codepoint and postcode_polygons - postcode centroids and polygons
- alternative_stations - access journey attributes for each mode obtained from OTP. For each origin postcode in survey_data this table contains multiple rows, each a potential alternative access station
- train_leg - stores attributes for the access station:egress station train leg obtained from OTP and BR Fares, such as time and fare
- station_facilities - stores station facilities attributes obtained from the NRE Stations Knowledgebase XML feed.

3.2 Building the OTP graph

Due to the high random access memory (RAM) requirement when building a graph[4] using large datasets, the build was carried out on a Microsoft Azure Cloud Server with 56GB of RAM[5]. The graph was then transferred to a local server with 16GB RAM for operation of the trip planner. The initial graph build used OSM data for GB in PBF format obtained from the Geofabrik website (Geofabrik, 2015) and GTFS data for GB rail services[6]. Although this resulted in a fully-functioning trip planner, a couple of deficiencies were identified:

- OTP allows roads tagged with highway=trunk to be traversed only by cars. In the UK there is no real distinction between trunk and primary roads, other than the body responsible for them, and the source code was amended to give traversal permission to all modes.
- OTP suggested unlikely driving routes via narrow unclassified roads. The UK OSM tagging guidelines note that tertiary roads are busy unclassified roads wide enough to allow two cars to pass safely (OpenStreetMap, 2015). However, OTP specifies the same average speed (25 mph) for tertiary, unclassified and residential roads. The source code was amended to increase the average speed of tertiary roads to 35 mph. Other adjustments included raising the average speed of secondary roads from 35 mph to 40 mph, and adjusting the speed of primary roads and motorways to 47 mph and 67 mph respectively, based on published free-flow road speeds (Department of Transport, 2015). These changes resulted in more realistic driving routes.

[4] The graph specifies every location in the region covered and how to travel between them, and is compiled from the OSM and GTFS data.

[5] It is only necessary to build a new graph when the underlying public transit data requires updating, or if an updated street network from OSM is required. Graph build is therefore likely to be an irregular occurrence.

[6] Converted from the ATOC CIF format and provided by GB Rail (GB Rail, 2015).

The next stage was to incorporate bus timetable data for Wales from the Traveline National Dataset (TNDS). This is in TransXChange format, a UK XML schema, and attempts were made to find an open source tool to convert this to GTFS format. The TransXChange2GTFS Converter (GoogleTransitDataFeed, 2016) was tested, but it aborted when processing the majority of TNDS XML files, despite the files passing validation, and it was rejected as a plausible solution. The only available alternative was Visography TRACC, which can import TransXChange files and export a GTFS feed. A number of error checking, correction and clean-up processes were performed on the exported GTFS feed before it was used for an OTP graph build, either to prevent fatal build errors or to improve performance.

3.3 Issues obtaining trip itineraries from OTP

The following issues were identified:

- In a very small number of cases OTP reported that the trip was not possible by car[7]. This was due to the nearest road to the origin postcode centroid not being available for car use, such as a pedestrianised street. In these cases the start point was manually adjusted in the OTP web interface until a valid route was returned, and the new coordinates stored in a lookup table.
- The walk reluctance parameter was increased to 5 for bus trips (the default is 2). This was based on experience requesting itineraries using the web interface and ensured a more realistic balance between the walk and bus components of the trip.

4 Conclusions

This paper has shown the successful application of a framework to automatically generate a range of variables for model input by processing open transport data using open source tools. The main limitation is the requirement to use commercial software to convert the TNDS dataset to GTFS format. This appears an exclusively UK problem, with the lack of suitable open source conversion tools most likely a consequence of GTFS becoming the de-facto standard for publishing public transport data around the world. Further work is needed to develop a comprehensive OTP API wrapper which could be released as an R package, and to improve performance by optimising code in the R scripts that link the various components together.

5 Acknowledgements

The authors wish to thank Paul Kelly for permission to use the brfares.com API, and Dan Saunders at Basemap Ltd for providing a TRACC educational license. The work reported here forms part of a PhD funded by EPSRC DTG Grant EP/M50662X/1. Code.Point Crown Copyright and Database

[7] In the initial lookup of drive distance from origin postcode to destination station, this issue only affected 7 postcodes.

Right 2015. Ordnance Survey (Digimap Licence). This work uses public sector information licensed under the Open Government Licence v3.0.

6 Biography

Marcus Young is a PhD student in the Transportation Research Group at the University of Southampton. His research interest is in applying GIS methods and techniques to investigate issues relating to public transport and non-motorised transport. His PhD project is: ”Modelling railway station choice using GIS”.

References

Blainey, Simon Philip (2009) “Forecasting the use of new local railway stations and services using GIS”, Ph.D. dissertation, University of Southampton.

BR Fares Ltd (2016) “BR Fares”, webpage, URL: `http://www.brfares.com`.

Conway, Joe, Dirk Eddelbuettel, Tomoaki Nishiyama, Sameer Kumar Prayaga, and Neil Tiffin (2013) *RPostgreSQL: R interface to the PostgreSQL database system*, URL: `http://CRAN.R-project.org/package=RPostgreSQL`, R package version 0.4.

Department of Transport (2015) *Free Flow Vehicle Speed Statistics: Great Britain 2014*.

GB Rail (2015) “GB Rail GTFS”, webpage, URL: `http://www.gbrail.info/`.

Geofabrik (2015) “Downloads”, webpage, URL: `http://www.geofabrik.de/data/download.html`.

Google (2016a) “The Google Distance Matrix API”, webpage, URL: `https://developers.google.com/maps/documentation/distance-matrix/usage-limits`, accessed on 10 March 2016.

Google (2016b) “Google Maps APIs”, webpage, URL: `https://developers.google.com/maps/`.

GoogleTransitDataFeed (2016) “GoogleTransitDataFeed Wiki”, webpage, URL: `https://code.google.com/archive/p/googletransitdatafeed/wikis/GoogleTransitDataFeed.wiki`, accessed on 10 March 2016.

National Rail Enquiries (2016) “Knowledgebase XMLs”, webpage, URL: `http://www.nationalrail.co.uk/100298.aspx`, accessed on 10 March 2016.

OpenStreetMap (2015) “United Kingdom Tagging Guidelines”, webpage, URL: `http://wiki.openstreetmap.org/wiki/United_Kingdom_Tagging_Guidelines`, accessed on 1 June 2015.

OpenTripPlanner (2015) “An open source multi-modal trip planner”, URL: `https://github.com/opentripplanner/OpenTripPlanner`, version 0.18.

Placr Limited (2016) “TransportAPI”, webpage, URL: `http://www.transportapi.com/`, accessed on 10 March 2016.

Young, Marcus and Simon Blainey (2016) “Defining probability-based rail station catchments for demand modelling”, in *48th Annual UTSG Conference, Bristol, GB*, URL: `http://eprints.soton.ac.uk/384539/`.