The UGent Institutional Repository is the electronic archiving and dissemination platform for all UGent research publications. Ghent University has implemented a mandate stipulating that all academic publications of UGent researchers should be deposited and archived in this repository. Except for items where current copyright restrictions apply, these papers are available in Open Access.

This item is the archived peer-reviewed author-version of:

What Public Transit API Logs Tell Us about Travel Flows

Pieter Colpaert, Alvin Chua, Ruben Verborgh, Erik Mannens, Rik Van de Walle, and Andrew Vande Moere


To refer to or to cite this work, please use the citation to the published version:

What Public Transit API Logs Tell Us about Travel Flows

Pieter Colpaert*  
Erik Mannens*  
Alvin Chua†  
Rik Van de Walle*  
Andrew Vande Moere†  
iMinds – Ghent University – Data Science Lab  
firstname.lastname@ugent.be  
Research[‡]Design – KU Leuven  
firstname.lastname@kuleuven.be

ABSTRACT

In the field of smart cities, researchers need an indication of how people move in and between cities. Yet, getting statistics of travel flows within public transit systems has proven to be troublesome. In order to get an indication of public transit travel flows in Belgium, we analyzed the query logs of the iRail API, a highly expressive route planning API for the Belgian railways. We were able to study ~100k to 500k requests for each month between October 2012 and November 2015, which is between 0.56% and 1.66% of the amount of monthly passengers. Using data visualizations, we illustrate the commuting patterns in Belgium and confirm that Brussels, the capital, acts as a central hub. The Flemish region appears to be polycentric, while in the Walloon region, everything converges on Brussels. The findings correspond to the real travel demand, according to experts of the passenger federation Trein Tram Bus. We conclude that query logs of route planners are of high importance in getting an indication of travel flows. However, better travel intentions would be acquirable using dedicated HTTP POST requests.

1. INTRODUCTION

Today, the Belgian railway company (SNCB) takes manual samples of people getting on and off trains in stations. To date, this is still the SNCB’s only source to determine people flows through their network. Also within other transit systems the data is tedious to get, certainly when passengers do not have to check-in and check-out. Nonetheless, indications of how people move within a transit network is of uttermost importance for, amongst others, making demand-driven policy decisions concerning mobility.

The iRail project started in 2008 to make route planning data of the Belgian railway accessible for developers. The project offers developers both a data dump for third party apps (since ~ August 2015) and a route planning API (since ~ October 2010). The query logs of the latter have been stored since November 2012 (requests before that period got lost due to server migrations).

Getting an indication of how people move within a transit network is of high importance within the field of smart cities, urban planning as well as demand driven mobility. Furthermore, it may benefit travelers to know in real-time whether the upcoming trip is in popular demand, as it may trigger the traveler to take the next, less congested, trip. In this paper, we study the query logs of route planning web-services as an indication of people flows within transit networks.

We give a small overview of related work to gather data for flow analysis. Next, we introduce the iRail query logs, which we study by the means of visualizations. Finally, we look at how public transit data is published and how we would be able to get a better indication of travel flows.

2. RELATED WORK

Flow analysis is a topic of theoretical interest and practical importance conventionally conducted to study spatial dynamics and identify routine patterns in the movement of people. For instance, interest in modelling traffic flows emerged from the strain placed on urban transportation systems during peak hours [18, 9]. Likewise, insight into routine travel patterns is crucial for the conceptualization of functional urban areas [19], urban hierarchies [6] and other territorial structures.

Over the past decade, large datasets have become increasingly commonplace thanks to the proliferation of sensor networks and portable devices like smartphones. Termed “Big Data” due to the large volume of data records that emerge from real-time sensing [13], such datasets typically contain information of activities or processes linked to the space and time where they occur. In the domain of “Smart City” research, much has been accomplished with the use of “Big Data” to monitor human movement. Smart card data from public transport systems [18, 5], taxi journeys [9] as well as cellular call data [20] have provided planners with new opportunities to develop greater understanding of mobility patterns in urban environments [4]. Yet query logs were, to our knowledge, never used for flow analysis so far.

2.1 Studying Query Logs

On highly expressive servers, such as SPARQL-endpoints, query logs can play an important role in fine-tuning the underlying datastores [3]. E.g., they determined that in 2011 in the context of the graph query language SPARQL, most of the queries are simple and include few triple patterns and joins. Furthermore, the graph patterns are usually star-shaped and despite triple pattern chains exist, they are generally short. Other, more cacheable interfaces, as introduced by Linked Data Fragments [25], also benefit from query logs in order to e.g., know who is using what clients to query the Web, yet the user intention is lost [24]. This was called “a blessing.
for privacy”, yet adding the user query as an HTTP header was suggested.

2.2 Visualizing Travel Flows

Visualizations are crucial for making data analysis tangible [22], so that results can be communicated and debated [17]. This is key for our research, since the questions presented are exploratory in nature [14] and can be addressed in many ways. Movement data has received substantial attention from the visualization community and a range of techniques has been developed to support flow analysis [2]. Visualizations of flows can be broadly categorized into two groups. Representations that display the complete trajectory and Origin-Destination (OD) type representations that take only the start and end locations into account. Since iRail’s query log data solely consists of OD information, we narrow our description to the latter group of visualization techniques.

OD type representations are employed when the origin and destination preside over intermediate locations on the path of movement. The OD Matrix is the most common visualization technique based on this type of representation. Rows and columns correspond to locations while cells are coloured to express the volume of flow. OD matrices can be reordered to emphasize connectivity between locations. While this feature is particularly useful for identifying trends and outliers in the data, the tabular display is challenging for lay users to interpret and less expressive of connections between locations than representations based on the node-link metaphor [11]. In recent years, such representations have become increasingly prominent for displaying movement in science as well as popular culture. Locations are depicted as nodes and arrows are drawn between them to indicate movement in a certain direction, allowing for exact and individual relationships to be traced. This representation is detailed and straightforward to understand but becomes cluttered when multiple lines intersect.

A number of solutions have been proposed for clutter reduction. Filtering is the most commonly practiced. The same goal can be achieved by clustering lines that have similar properties like common start, end or intermediate locations [1]. Both solutions are effective at reducing clutter but it should be noted that excessive aggregation or filtering removes a significant amount of information from the visualization.

Another solution is to transform the visual representation. This can be achieved by bundling lines in close proximity to minimize intersections [12]. Flow maps [16] follow the same principle to reduce clutter but yield layouts that align with key geographic features like roads, rivers or coastlines. However, a certain degree of ambiguity is introduced to the visualization by both techniques since the arrows do not depict actual paths of movement and may be misleading to interpret. Transition diagrams are a type of visualization that summarizes large numbers of trajectories while minimizing arbitrary transformations to the layout [2]. The idea is to divide each trajectory into segments, so that movement on a path is approximated by a sequence of transitions between adjacent discrete locations. Locations are either predefined or obtained by various space partitioning algorithms. Arrows depict flows and are scaled to the aggregate number of transitions between pairs of adjacent locations. An advantage of this representation is that the topology of flows is captured as a directed graph, allowing for quantitative metrics like centrality to be computed and visually compared. Nonetheless, the solution requires substantial data processing prior to visualization and analysis. A computationally cost-effective alternative is to draw links as arcs instead of arrows [7]. Yet, this visual encoding may intensify the visual complexity of the representation.

Non-geographic representation is a viable alternative to this challenge in this specific context of visualizing flows on the rail network since the locations of stations and the exact routes between them are known and have been pre-determined. Thus the analytical focus and goal of visualization is to depict differences in the volume of flows between different stations. While many visualization techniques fulfill this purpose, we find chord diagrams [15] particularly appropriate for delivering information in a compact form, allowing for comparison across multiple plots in a single view. A further description of chord diagrams is provided in Section 4.2.

3. THE QUERY LOGS

The iRail API is an XML/JSON HTTP API: each request on http://api.irail.be/ returns a response that can be used directly in an end-user app. It exposes four features:

1. Route Planning2 plans a route from one station to another, taking into account a preferred start time.
2. Next Departures3 shows the next train departures in a certain station, useful for creating quick overviews.
3. Trip Status4 returns the stations this train passes by on the day of the request.
4. List of all stations5 gives an overview of all stations Belgian trains can arrive at.

This interface is open source and we were able to suggest changes to the system in which the query logs become open data6. As of the 17th of December 2015, iRail publishes the most recent 1,000 requests done on the API as open data, each second, at http://api.irail.be/logs/. In order to publish this data, we chose a newline-based format: each new line contains a new query object encoded in JSON. The JSON objects can be read one per one, so UNIX commands such as grep work well with this format. We chose this over CSV as we have more flexibility towards changing the object properties in the future. Next, we chose to add an HTTP header with a link to the query log context according to the JSON-LD specification7 to turn it into Linked Data.

Using this data, everyone can build their own historic query log dataset. The server costs for iRail to host this dataset stay limited. Furthermore, this way of publishing data enables research for real-time predictions of travel intentions. Privacy preservation is paramount [21] when dealing with query logs and, therefore, IP addresses have been removed to protect the privacy of iRail’s users. Additionally, the integrity of this dataset, without IP addresses, has been vetted by the Belgian Privacy Commission and permission has been granted for these query logs to be published as open data.

We are particularly interested in studying the potential application of route planning queries as an information resource for flow analysis due to the input it captures from human users. The other features are less relevant for this purpose as they are generally polled by digital signage providers or status boards, and thus do not reflect direct human demands. The dataset described in this paper is compiled from the Apache access.log files generated between October 2012 and December 2015, filtered by “connections” entries. We made a copy of this dataset available in CSV format can be downloaded at https://github.com/pietercolpaert/irail-route-planning-query-logs. Each row within this dataset contains:

1. A timestamp of when the request was executed, 2http://api.irail.be/connections/{?from, to, date, time} 3http://api.irail.be/liveboard/{?station, date, time} 4http://api.irail.be/vehicle/{?id, station, date} 5http://api.irail.be/stations/ 6https://github.com/IRail/IRail/pull/138 7http://www.w3.org/TR/json-ld/
were excluded since they do not explicitly represent human travel with an intention to travel by rail, iRail’s query log data captures which explains the sudden rise of iRail API queries.

Accordingly, such queries made by automated user-agents like search engine bots and data harvesters can be considered as valid traffic but may over represent beyond the time range of 06:00 hours and 10:00 hours. Queries this regard, the dataset was filtered to exclude data records created resulting in widespread adoption of alternative software applications built on top of the iRail API. Calculated with indicators provided by the Flemish regional government of Belgium, we infer a monthly average of 19.3, 19.4, 19.6 and 19.7 million passengers in 2012, 2013, 2014 and 2015 respectively. Correspondingly, the average number of iRail queries per month amounts to 0.11, 0.17, 0.15 and 0.33 million respectively. Assuming that each request reciprocates with an intention to travel by rail, iRail’s query log data captures a respective 0.56%, 0.88%, 0.77% and 1.66% of the actual journeys that have occurred.

2. The request path between two stations and their associated geographic coordinates,
3. The user-agent string cfr. RFC2616,
4. The operating system of the user-agent and
5. A flag stating if the user-agent is a bot.

There has been an increase in the number of queries made between 2012 and 2015. Figure 1 provides a breakdown of the numbers on a monthly basis. A distinctive increase in the number of queries can be observed in April 2015 when one of the official route planning apps provided by the Belgian rail company (RailTime) was discontinued, resulting in widespread adoption of alternative software applications built on top of the iRail API. Calculated with indicators provided by the Flemish regional government of Belgium, we infer a monthly average of 19.3, 19.4, 19.6 and 19.7 million passengers in 2012, 2013, 2014 and 2015 respectively. Correspondingly, the average number of iRail queries per month amounts to 0.11, 0.17, 0.15 and 0.33 million respectively. Assuming that each request reciprocates with an intention to travel by rail, iRail’s query log data captures a respective 0.56%, 0.88%, 0.77% and 1.66% of the actual journeys that have occurred.

4. METHOD

Flow analysis involves the comparison of aggregated movement between distinct locations within a specific time frame. In most cases, the analytical objective is to identify trends that occur in journeys made between pairs of locations and allow for their differences to be explored. Data visualization is typically employed to facilitate in this process thus the input data must be transformed into a format that allows for the quantity and directional flow of movement to be visually encoded. The data must undergo several stages of data processing to arrive at this outcome.

4.1 Data Processing

We were specifically interested in understanding how travel during the morning rush hour on weekdays differ from weekends. In this regard, the dataset was filtered to exclude data records created beyond the time range of 06:00 hours and 10:00 hours. Queries made by automated user-agents like search engine bots and data harvesters can be considered as valid traffic but may over represent connections between certain stations. Accordingly, such queries were excluded since they do not explicitly represent human travel behavior. Station names tend to contain be highly inconsistent since any text information can be parsed to the API. This may include invalid character encoding, spelling mistakes and the use of unstandardized abbreviations. We employed an open-source reconciliator (https://github.com/irail/stations) to reduce spelling variations and remove data records with missing information. Through this process, station names were also linked to their corresponding geographical coordinates.

Next, the data is aggregated so that province level flow patterns are emphasized. Figure 3 provides a diagrammatic representation of spatial aggregation on the basis of provincial administrative boundaries. To compare the volume of flow between rural provincial areas and urbanized zones, train stations in major cities were aggregated into separate groups. Similarly, stations located outside of Belgium were aggregated into distinct group. A diagrammatic breakdown is depicted in Figure 2 where color and shape is used to demarcate each group. The following types of flows are observable from the outcome of aggregation:

- Travel from any provincial station to a major city.
- Travel between any two major cities.
- Travel between any two provincial stations.
- Travel between any international station to a major city.
- Travel between any international station to a provincial station.

4.2 Visualization

The chord diagram is a visualization technique based on the node-link metaphor that arranges nodes along the circumference of a circle [15]. Each node is represented by an arc where its length is proportional to the total volume of incoming and outgoing flows. Chords or curved line segments are drawn to connect nodes. The width at the head or tail of each chord indicates the amount of movement relative frequency of movement from a certain location to another. In our implementation of the chord diagram, nodes are colour coded to indicate individual provinces and the major cities located within their administrative boundaries. Interactive filtering is introduced to simplify the visualization and details are provided in pop-up dialogue when on mouse-over.

5. RESULTS

In this chapter, we report on the results of studying visualizations with experts of Trein Tram Bus[10], a not for profit passenger federation.

5.1 Time Based Analysis

Results from the analysis of this dataset reveal trends in route planning queries that may provide insights into rail travel demand. First, the daily and hourly distribution of queries appear to correspond with known commuting patterns. Figure 4 depicts a break down of the queries that occur, on average, on an hourly basis for each day of the week. Several trends can be observed in this chart. Morning and evening peak periods in particular, are clearly distinguishable. At noon, a small dent is noticeable, which may be attributed to part time workers and meetings on location after or before noon. Peak hours on weekdays are also distinctively different from those of weekends. Similarly, the frequency of queries during the evening peak period on Fridays appear to be substantially higher than those that occur between Monday and Thursday. The discrepancy maybe explained by students that travel home from their student homes over the weekends. The absence of a clear peak on Saturdays can be attributed to lack of journeys to work queries while the evening

---

Figure 1: Amount of queries on the route planning part of the iRail API between October 2012 and December 2015. In April 2015 an official app of the Belgian railways was discontinued, which explains the sudden rise of iRail API queries.

\[\text{https://hello.irail.be/2015/04/22/april-updates/}\]

\[\text{http://treintrambus.be}\]
5.2 Structure of Flows in Belgium

An interactive chord diagram can be viewed at [http://rxd.architectuur.kuleuven.be/irail/weekdays.html](http://rxd.architectuur.kuleuven.be/irail/weekdays.html) and [http://rxd.architectuur.kuleuven.be/irail/weekends.html](http://rxd.architectuur.kuleuven.be/irail/weekends.html). The chord diagrams depict the aggregated number of queries made between any two stations on weekdays, as well as on weekends. Visualizing our dataset in this manner reveals the complexity of flows on the Belgian rail system and the significance of cities in daily travel. Brussels City stands out as the most distinctive visual element in both figures, indicating its function as the principal centre of rail activity. On a regional level, its function as a centre appears to be more distinctive for the Walloon region than Flanders. With exception to Liège city, queries from the Walloon region are generally made from provincial stations towards Brussels instead of major cities within their administrative boundaries. Queries from Flanders, on the other hand, tend to be distributed among four major cities otherwise known as the Flemish Diamond, a network of metropolitan areas in Belgium comprised of Brussels, Antwerp, Ghent and Leuven. These insights indicate that the structure of flows in Flanders appear to follow a polycentric pattern while that of the Walloon region is relatively monocentric, with Brussels city as a major centre. The difference between both regions appear to correspond with existing measures of population density and degree of urbanisation, providing valuable insight as well as alternative perspectives into the function of cities in rural and urban settings.

5.3 Counter-Intuitive Results

There are several inconsistent patterns in the data that do not correspond with an existing understanding of flows in Belgium. In particular, the large number of trips made between Antwerp city and other stations within the province of Antwerp in the morning, is not observable in other major cities in Flanders. Further investigation led to the identification of an isolated connection originating from Antwerp city towards a suburb, suggesting anomalous usage of the peak on Sundays may emerge from students returning to college accommodations.

Moreover, the distribution of queries on public holidays is observed to deviate from that of an average day. Figure 5 provides a breakdown of queries per hour on Christmas eve, December 24th 2015, in comparison to an average Thursday of the same year. As illustrated, the evening peak on Christmas eve occurs earlier than that of an average Thursday.
iRail API that were not flagged as bots or a high concentration
people who utilise route planning applications on a regular basis.
Alternatively the large number of trips may have originated from
users who repeatedly request for updates on delayed trains or from
a concentration of people who utilise route planning applications on
a regular basis. Accordingly, the reverse is identified in Liège where
an unusually large connection originating from a suburb towards
Liège city exists. This prompted speculation, as well as discussions,
on how such outliers should be treated or what may have led to their
existence. As with any form of analysis based on appropriation of
data, we acknowledge that more work is required to identify caveats
and understand the validity of the findings presented.

6. PUBLISHING TRANSPORT DATA

The expressiveness of a server affects the way logs can be gath-
ered [24]. We have shown in previous sections that the query logs
of a highly expressive server, such as the iRail API, are interesting:
each request contains an entire query which can be interpreted as
a travel intention. The possibility that two requests are exactly the
same is low, as there are many URLs which can be requested. Nev-
evertheless, we cannot fully guarantee that each HTTP GET request
to the server will trigger a log entry, due to caching mechanisms on
the Web [10], which might lead to a false representation.

Hosting a route planning API as the only way to publish transport
data comes with three identified limitations, as the server will need
to handle the requests from different use cases with different needs:
1. When an application developer would like a new feature, such
as taking wheelchair accessibility information into account,
the feature would have to be implemented on the server of the
data publisher.
2. Keeping the server highly available is costly, as any question
can be asked by anyone for any purpose.
3. Federated querying, which would allow for intermodal route
planning for route planning APIs, is inexisten up to date.

In order to overcome these limitations, the General Transit Feed
Specification (GTFS)

11

11https://developers.google.com/transit/gtfs/reference

supported among all current open-source route planning software
systems and it is used in products/apps such as CityMapper, Ally,
Navitia.io, Google Maps and Bing Maps. It successfully enabled
reuse for intermodal travel, engineers can rely on the data dumps
even if the servers of the transit agency are offline and there is no
limitation to the features that can be implemented. The public transit
agency however has no access any longer to an indication of travel
demand.

In Figure 6 we illustrate these two options as two extremes, with
other options that are yet to be discovered. When a server only
allows to set e.g., a departure station and a departure time, then the
server cannot log the arrival station, yet the client is still able to plan
a route by executing the algorithm on the client-side [8]. We would
be able to fully rely on the query logs if the expressivity would be
maximal (extreme right) and caching would be turned off. Turning
off caching may work in a private setting where you know how
much and what kind of queries you can expect, yet on the open Web,
caching helps decrease load on servers at peak moments.
7. CONCLUSION AND FUTURE WORK

Firstly, we studied query logs to find travel patterns in Belgium. We conclude that this is up to now, the best representation of travel flows over the Belgian railway network. We studied data which represent a maximum of 1.66% of the passengers. By visualizing the average distribution of requests on workdays and Saturday and Sunday, we were able to recognize the patterns we would expect: a.m. a morning and evening peak on workdays and a bigger evening peak on Friday and Sunday. More interestingly, we were able to see that the evening peak on Christmas Eve, Thursday the 24th of December 2015, started earlier than average on Thursdays. Visualizing our dataset using the origin and destination reveals the complexity of movement on the Belgian rail system and the significance of cities in weekday travel. Furthermore, we found evidence that Flanders is polycentric, while Walloon traffic is monocentric.

Secondly, we also concluded that there are obvious caveats associated with the use of such data as proxy for actual statistical counts. We identified a couple of gaps:
1. The data only captures an intention of an end-user: it is unsure whether the person actually took the train.
2. Multiple user queries may be needed to cover a travel intention.
3. Caching may mask peak hours, as many of the same requests in one minute is only stored in the logs once.

Nevertheless, we published the query logs as open data at http://api.traill.be/logs for other scientists to continue to research possibilities for other scientists to continue to research possibilities.

Related work [23] suggested to add additional metadata in the HTTP headers in order to track the user’s intention. While this is an interesting suggestion, it does not overcome the problem of caching. In order to create a better representation of travel intentions, future work can gather logs by POST requests with the only purpose to gather analytics of travel intention. This is a similar approach than what happens on the Web of Documents today with Piwik or Google Analytics.

8. REFERENCES