The Linked Connections framework for reliable historical and real-time data

David Chaves-Fraga [a], Julian Rojas [b], Pieter Colpaert [b] Oscar Corcho [a] and Ruben Verborgh [b]
[a] *Ontology Engineering Group, Universidad Politécnica de Madrid, Spain*
[b] *IDLab, Department of Electronics and Information Systems, Ghent University-imec, Belgium*
E-mails: dchaves@fi.upm.es, julianandres.rojasmelendez@ugent.be, pieter.colpaert@ugent.be, ocorcho@fi.upm.es, ruben.verborgh@ugent.be

Abstract. Using Linked Data based approaches a wide amount of companies and institutions share their data in an affordable way while allowing third parties to perform standard and federated queries. In transport domain, where a complex environment emerge, static, real-time and historical data have to coexist to provide reliable data to the information systems. Currently, however, public transport data is published in a way in which the processing is too expensive. In previous work, the Linked Connections (LC) framework was introduced as a cost-efficient publishing alternative to the *de-facto* standard GTFS and route planning APIs. LC provides a light HTTP interface that allows smart clients to create their own route planners. We notice that at the moment in which of historical and real-time are taken into account in the LC framework, some improvements are needed. Specifically in the case of live updates about the schedules, where it is important to maintain stable identifiers that remain valid over time.

This paper continues the previous work we have made for developing an affordable framework to publish reliable real-time and historical transport data. Our main contributions are: (i) a Linked Connections Real Time server that is able to process GTFS-RT feeds providing consistent identifiers, (ii) an efficient management of historical data taking into account the size of each fragments exposed on the Web and (iii) an implementation of multiple route planning algorithms to test if our approach provides reliable access to real-time and historical data. We evaluate and compare the contributions with our previous approaches where the distribution of the fragments were based on other variables, like the time. We discover that taking into account the size of the fragments has a relevant impact in the performance of query evaluation, as originally expected. In future work we would like to identify the optimal size of the fragments automatically, taking into account multiple variables like the geographical location or the type of transport.

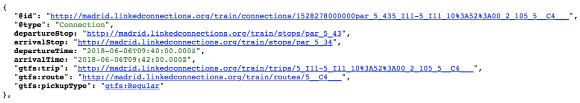
Keywords: Linked Connections, Historical Data, Real time data, Reliable data

1. Introduction

In the current state of the Web of Data, a wide amount of that data are exposed following the principles of Linked Data[1]. Giving unique identifiers to each resource, representing the data using a shared and common vocabulary of the domain or the possibility of dereferencing each URI are some of the relevant aspects that made Linked Data as one the most common approaches to organize and expose the data on the Web[2] This features allow third parties to query the data in a standard way, using for example, the corresponding query language for RDF, SPARQL[3], and the possibility of doing federation across multiple datasets[4]. However, today, a lot of domains need to deal with real-time and historical data where is important to maintain stable identifiers that remain valid over time. Transport is one of these domain, where a complex environment with multiple types of data sources have to be managed to provide reliable data to information systems.

Since May 2017, one of the main motivations for developing solutions about multimodal and integrated travel information services is the publication of the new directive by the EU Commission about discoverability and access to public transport data across Europe. This document proposes the making of public transport data from providers available on national or common access points saved on databases, data warehouse or repositories. All the states will provide access

1570-0844/0-1900/$35.00 © 0 – IOS Press and the authors. All rights reserved



```
{
  "@id": "http://madrid.linkedconnections.org/train/connections/1528278000000par_5_435_I11-5_I11_10%3A52%3A00_2_105_5__C4___",
  "@type": "Connection",
  departureStop: "http://madrid.linkedconnections.org/train/stops/par_5_43",
  arrivalStop: "http://madrid.linkedconnections.org/train/stops/par_5_34",
  departureTime: "2018-06-06T09:40:00.000Z",
  arrivalTime: "2018-06-06T09:42:00.000Z",
  "gtfs:trip": "http://madrid.linkedconnections.org/train/trips/5_I11-5_I11_10%3A52%3A00_2_105_5__C4___",
  "gtfs:route": "http://madrid.linkedconnections.org/train/routes/5__C4___",
  "gtfs:pickupType": "gtfs:Regular"
},
```

Fig. 1. Example of a connection at LC in JSON-LD

to a unique common point following different static standards as Transmodel[1], Datex II[2] or GTFS[3] and real-time standards GTFS-RT[4] or SIRI[5]. So the domain requires solutions able to provide reliable data and to deal with the heterogeneity of access points and the data formats.

One of the main challenges when the Linked Data approaches deal with transport domain, where real-time and historical data has to be taken into account for providing consistent data to the information services, is how to ensure that the identifiers of each resource is stable and valid over the time. For example, if we define the connection entity as a departure-arrival pair, in previous works on Linked Connections[5], the URI of each connection was defined getting information from an static GTFS datasets. At the moment that real-time is involved, that URIs are not consistents because the real-time information has to be taken into account. An example of the conceptualization of a connection is shown in the Figure 1. Other relevant challenge is how to manage the exposed historical data on the Web to allow an optimal query performance. One of the requirements of most relevant route planning algorithms is that the data have to be sorted by the time. In previous works of Linked Connections[6], we developed a server that paginates the list of connections in departure time intervals (10 minutes) and publishes these pages over HTTP. We have noticed that the clients were able to analyze the historical data but the performance was too low.

Our work is focused on providing an improvement of the current state of Linked Connections framework by allowing an efficient management of historical and real-time data with a standard vocabulary for the transport domain. This is relevant because two main reasons: (i) currently, a lot of transport companies are starting to provide access to their real-time services, but they the most common way to do it, is to develop an ad-hoc solutions like APIs or web services that only works locally [7] and (ii) the heterogeneity of the domain in terms of data formats will be a very relevant problem next years, especially in Europe, based on the proposal of the EU Commission so a standard framework will be needed.

In this paper we present the Linked Connections framework to provide reliable access to real-time and historical data. Our main contribution is the extension of previous version of the LC server, by providing an efficient management of real-time and historical data. First, we develop a library that get information from static GTFS datasets and GTFS-RT data streams and exploit and integrate them following the LC vocabulary. Second, we modify the approach of Linked Connections for splitting the fragments of the data using a specific size of each fragment instead of the time. Third, we program a set of route planning algorithms in the top of the LC server to test if our approach is able to provide reliable data. Finally, we evaluate the improvements comparing the new versión of the LC framework with our previous approaches, as base line.

The paper is organized as follows: Section 2 presents some of the related work on benefits of Linked Data, solutions to query data on the web, the EU regulation about the availability of transport data on the Web and current (semantic) route planners. Section 3 describes our proposal about the improvements of the Linked Connections framework. Section 4 presents the design of our experiments. Section 5 describes the results we obtained evaluating our main contributions. Section 6 provides a brief discussion about the relevance of our contributions, and Section 7 presents conclusions and areas for future work.

2. Related Work

On the current state of the Web, huge amount of data are exposed following the principles of Linked Data. We describe the main contributions on this topic, and also the relevance of the ones made in querying data on the web and the current solutions on planning transport routes using semantics.

One of the most well-known alternatives to publish data on the Web is Linked Data [1]. Linked Data allows to identify in an unique way resources on the Web using identifiers, or HTTP URIs. It is a method to distribute and scale data over large organizations such as the Web. When looking up this identifier by using the HTTP protocol or using a Web browser, a definition must be returned, including links towards poten-

[1] http://www.transmodel-cen.eu
[2] http://www.datex2.eu
[3] https://developers.google.com/transit/gtfs
[4] https://developers.google.com/transit/gtfs-realtime/
[5] http://www.transmodel-cen.eu/standards/siri/

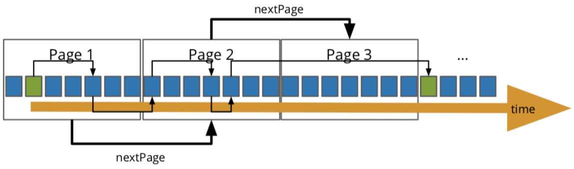


Fig. 2. Linked Connections implementation

tial other interesting resources, a practice called *dereferencing*. The triple format to be used in combination with URIs is standardized within RDF. The URIs used for these triples already existed in other data sources, and we thus favoured using the same identifiers. It is up to a data publisher to make a choice on which data sources can provide the identifiers for a certain type of entities.

A common problem in Linked Data is the availability of the triple stores. They provide a way to getting data using the SPARQL query language but at the moment of queries involved long periods of time, these approaches are not efficient[8]. The Linked Data Fragments[9, 10] solve this issue fragmenting the data in several HTTP documents. Following this approach the store moves the load from server side to client side improving its availability. Linked Connections[5] applies this approach to develop an cost-efficient solution based on a HTTP interface for transport data. The main assumption of LC is that the relevant data for route planners can be based on the connection concept. Basically, as the LC vocabulary[6] defines it, a connection describes a departure at a certain stop and an arrival at a different stop with their corresponding departure and arrival times. A basic implementation of LC is shown in Figure 2, where the route planning algorithms have to analyse the connections (small rectangles) through the pages (big rectangles) and across the time to find the expected route. The join between the connections is possible because same resources have same identifiers, based on one of the principles of Linked Data.

It also relevant to describe the previous works that has been carried out using the specification of Linked Connections. For example, [11] describes a analyse the behaviour of a basic transport API and the Linked Connections framework for public transit route planning, comparing the CPU and query execution time. The authors found that, at the expense of a higher bandwidth consumption, more queries can be answered using LC than the origin-destination API. In [12] studies the impact of taking into account user preferences in a public transit route planning adding that features both on server and client and comparing the two solution on query execution time, cache performance and CPU usage on both sides. A first step for providing reliable access to historical and real-time data using Linked Connections is described in [6], where a mechanism is introduced to tackle the problem of the management of data modifications when real-time is involved in route planning queries. Tripscore[7], a Linked Data client that consume several Linked Connections servers with real-time and historical data is also described in [13], where the Connection Scan Algorithm (CSA) is implemented as the route planning algorithm in top of the client[14]. Tripscore add multiple user preferences at the client side to provide an score for each possible route.

All the solutions that we aforementioned are rely on the *de-facto* standard for representing public transport data, the General Transit Feed Specification or GTFS. This model, and its extension for real-time (GTFS-RT) is used by Google Maps[8] since 2005 but also by other route planners like Open Trip Planner [9] or Navita.io[10]. It is also the most common model used by the transport companies to expose their data on open data portals, like for example the Consorcio General de Transportes de Madrid[11], the TRAM in Barcelona[12] or the Belgium National Train System (SNCB). GTFS defines the headers of 13 types of CSV files and a set of rules the must be take into account when the dataset is created. Each file, as well as their headers, can be mandatory or optional and the have relations among them as show in Figure 3. Linked Connections is getting the necessary information from a subset of the full dataset:

- stops: Individual locations where vehicles pick up or drop off passengers.
- calendar: 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: Exceptions for the service IDs defined in the calendar file.
- stop_times: Times that a vehicle arrives at and departs from individual stops for each trip.

[6] http://semweb.mmlab.be/ns/linkedconnections

[7] www.tripscore.eu
[8] http://maps.google.es
[9] http://www.opentripplanner.org
[10] https://www.navitia.io
[11] http://datos.crtm.es
[12] https://opendata.tram.cat/

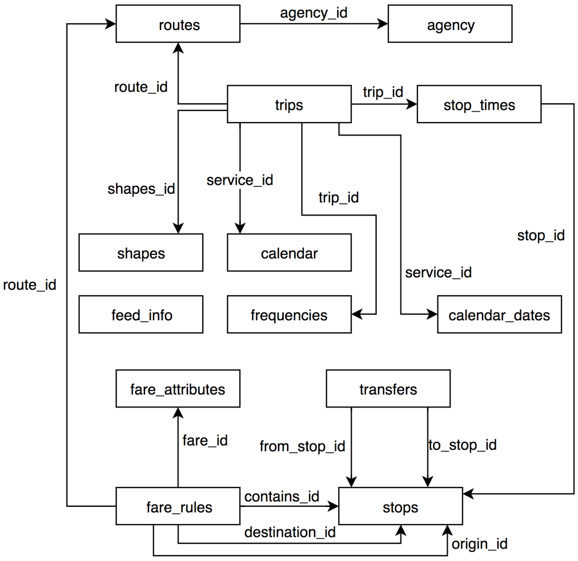


Fig. 3. The GTFS model and its primary relations

- trips: Trips for each route. A trip is a sequence of two or more stops that occurs at specific time.
- routes: Transit routes. A route is a group of trips that are displayed to riders as a single service.
- transfers: Rules for making connections at transfer points between routes.

In order to link the terms and identifiers define in these files with the Linked Open Data cloud, we have created the Linked GTFS[13] vocabulary. We create mappings able to transform GTFS files to Linked GTFS following the CSV2RDF[15] W3C recommendation[14] but also using other standard OBDA mapping languages that are able to deal with CSV files[15], like RML[16] or R2RML[17].

The extension of GTFS for real-time, GTFS-RT[16] is a feed specificaftion that allows public transport agencies to provide realtime updates about their fleet. The specification supports three types of information: (i) trips updates like delays, cancellations or change routes, (ii) service alerts like stop moved, unforeseen events affecting stations, routes, etc and (iii) information of vehicle positions including location and congestion level. The data exchange format is based on Protocol Buffers[17].

It is also important to mention the works about route planning algorithms, that can be developed at the top of the Linked Connections framework. The problem that this algorithms has to solve using the data is the Earliest Arrival Time (EAT). An EAT query consists of a departure stop, a departure time and a destination stop.

3. The Linked Connections Framework

The basic properties used to define an `lc:connection` are:

- `lc:departureTime` Date-Time (including delay) at which the vehicle will leave for the lc:arrivalStop.
- `lc:departureStop` The departure stop.
- `lc:departureDelay` When the lc:departureTime is not the planned departureTime.
- `lc:arrivalTime` Date-Time (including delay) at which the vehicle will arrives at lc:arrivalStop.
- `lc:arrivalStop` The departure stop.
- `lc:arrivalDelay` When the lc:arrivalTime is not the planned arrivalTime.

3.1. *Real-time Linked Connections library*

3.2. *Real-time and historical Linked Connections server*

3.3. *Route planning algorithms*

4. Evaluation Design

4.1. *An example*

5. Results

6. Discussion

7. Conclusions and Future work

Acknowledgements

References

[1] C. Bizer, T. Heath and T. Berners-Lee, Linked data-the story so far, *International journal on semantic web and information systems* **5**(3) (2009), 1–22.

[2] T. Heath and C. Bizer, Linked data: Evolving the web into a global data space, *Synthesis lectures on the semantic web: theory and technology* **1**(1) (2011), 1–136.

[13] http://vocab.gtfs.org/terms

[14] https://github.com/OpenTransport/gtfs-csv2rdf

[15] https://github.com/dachafra/gtfsmappings

[16] https://developers.google.com/transit/gtfs-realtime/

[17] https://developers.google.com/protocol-buffers/

[3] E. Prud, A. Seaborne et al., SPARQL query language for RDF (2006).

[4] C. Buil-Aranda, M. Arenas, O. Corcho and A. Polleres, Federating queries in SPARQL 1.1: Syntax, semantics and evaluation, *Web Semantics: Science, Services and Agents on the World Wide Web* **18**(1) (2013), 1–17.

[5] P. Colpaert, A. Llaves, R. Verborgh, O. Corcho, E. Mannens and R. Van de Walle, Intermodal public transit routing using Liked Connections, in: *International Semantic Web Conference: Posters and Demos*, 2015, pp. 1–5.

[6] J.A. Rojas Melendez, D. Chaves, P. Colpaert, R. Verborgh and E. Mannens, Providing reliable access to real-time and historic public transport data using linked v-connections, in: *ISWC2017, the 16e International Semantic Web Conference*, Vol. 1931, 2017, pp. 1–4.

[7] P. Colpaert, A. Chua, R. Verborgh, E. Mannens, R. Van de Walle and A. Vande Moere, What public transit API logs tell us about travel flows, in: *Proceedings of the 25th International Conference Companion on World Wide Web*, International World Wide Web Conferences Steering Committee, 2016, pp. 873–878.

[8] R. Verborgh, O. Hartig, B. De Meester, G. Haesendonck, L. De Vocht, M. Vander Sande, R. Cyganiak, P. Colpaert, E. Mannens and R. Van de Walle, Querying datasets on the web with high availability, in: *International Semantic Web Conference*, Springer, 2014, pp. 180–196.

[9] R. Verborgh, M. Vander Sande, O. Hartig, J. Van Herwegen, L. De Vocht, B. De Meester, G. Haesendonck and P. Colpaert, Triple Pattern Fragments: a low-cost knowledge graph interface for the Web, *Web Semantics: Science, Services and Agents on the World Wide Web* **37** (2016), 184–206.

[10] R. Verborgh, M. Vander Sande, P. Colpaert, S. Coppens, E. Mannens and R. Van de Walle, Web-Scale Querying through Linked Data Fragments, in: *LDOW*, Citeseer, 2014.

[11] P. Colpaert, R. Verborgh and E. Mannens, Public Transit Route Planning Through Lightweight Linked Data Interfaces, in: *International Conference on Web Engineering*, Springer, 2017, pp. 403–411.

[12] P. Colpaert, S. Ballieu, R. Verborgh and E. Mannens, The Impact of an Extra Feature on the Scalability of Linked Connections., in: *COLD@ ISWC*, 2016.

[13] D. Chaves-Fraga, J. Rojas, P.-J. Vandenberghe, P. Colpaert and O. Corcho, The tripscore Linked Data client: calculating specific summaries over large time series, in: *Proceedings of the Workshop on Decentralizing the Semantic Web (DeSemWeb)*, 2017.

[14] J. Dibbelt, T. Pajor, B. Strasser and D. Wagner, Intriguingly simple and fast transit routing, in: *International Symposium on Experimental Algorithms*, Springer, 2013, pp. 43–54.

[15] J. Tennison, G. Kellogg and I. Herman, Model for tabular data and metadata on the web. W3C recommendation, *World Wide Web Consortium (W3C)* (2015).

[16] A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens and R. Van de Walle, RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data., in: *LDOW*, 2014.

[17] S. Das, S. Sundara and R. Cyganiak, R2RML: RDB to RDF Mapping Language, W3C Recommendation 27 September 2012, *Cambridge, MA: World Wide Web Consortium (W3C)(www. w3. org/TR/r2rml)* (2012).