Context-Aware Route Planning: a Personalized and Situation-Aware Multi-Modal Transport Routing Approach

Pieter Bonte[0000-0002-8931-8343], Mathias De Brouwer[0000-0001-8769-6861], Dörthe Arndt[0000-0002-7401-8487], Miel Vander Sande, Pieter Heyvaert, Anastasia Dimou[0000-0003-2138-7972], Pieter Colpaert[0000-0001-6917-2167], Ruben Verborgh[0000-0002-8596-222X], Filip De Turck[0000-0003-4824-1199], and Femke Ongenae[0000-0003-2529-5477]

IDLab, Ghent University – imec
Pieter.Bonte@UGent.be

Abstract. People increasingly rely on multi-modal (car, train, bus, e-bike, etc.) transport. However, current route planning does not allow the flexibility to seamlessly integrate users' preferences and goals, different transport types, or adapt to disruptive traffic events. In this demonstrator, we showcase a context-aware semantic route planner that combines users' individualized goals with geographical route planning using multi-modal transport modes. This means that the route planning can incorporate various goals of the user, e.g. visiting a friend or picking up an item, while the planner figures out the best order of visiting and the most suited transport modes. As transport is very dynamic, Stream Reasoning techniques are employed for real-time traffic monitoring. This allows the planner to automatically adapts the route when interruptions or obstructions, resulting in delays, are detected.

Keywords: Route planning · Transport · Context-aware · Stream Reasoning.

1 Introduction

As the pressure on traffic infrastructure increases, route planning is currently missing the flexibility to keep up with the diversity of its users' preferences [1]. Current route planning can be seen as rather static, only limited transport modes are supported, and not all modes can be combined. Most importantly, they lack the flexibility to incorporate personalization, i.e. user preferences and goals, and adapt to changing traffic situations. This demonstration paper proposes a context-aware route planner, that exploits knowledge-driven techniques to integrate profile information and traffic situational awareness. The designed semantic route planner finds the best connections, based on the profile information and preferences of its users. It takes into account the preferred transport modes of its users in different situations, e.g. a user might like to walk up to 2 km when it is not raining; bike up to 5km; take the bus when it's not rush hour; etc. Furthermore, one can add various goals, i.e. tasks one wants to accomplish when reaching the destination. This can range from visiting certain locations to picking up personal transport modes, such as picking up a bike and using it for further transport. As traffic can be very turbulent, our route planner can monitor real-time traffic streams. We exploit Stream Reasoning (SR) techniques [6] to infer disruptions and obstructions from the traffic streams in real-time. The SR infers which events will have negative influence, e.g. delays, on the suggested routes of the users, such that alternatives can be automatically suggested in order to decrease waiting time.

2 State of the art

Specialized software exists to calculate routes on road and transit networks. These can be categorized in: (i) software as a service, (ii) self-hosted server software, as well as (iii) client-side route planners.

Software as a service that can be used for route planning includes Mapbox turn-by-turn[1], Google Maps[2], HERE[3], Navitia.io[4], etc. The drawback of this approach is that it is only customizable to the extent the service provider allows to customize the route planning.

Open-source tools to set up a route planner on your own server include Open Trip Planner[5], OSRM[6], or Itinero[7]. The drawback of this approach is that it is tedious to maintain.

A third approach is a client-side route planner. A Software Development Kit (SKD) called Planner.js[8] is used in this demo. The SDK can integrate with public transport data. Currently, it works for Belgian public transport data [5], and world-wide short road network queries with a republished version of OpenStreetMap as Routable Tiles [4].

None of the existing approaches allows to easily integrate profile information or monitor traffic streams. However, client-side route planners such as Planner.js allow the most flexibility to integrate different types of transport, profile information, and traffic monitoring.

3 Context-aware route planning

Figure 1 depicts the architecture of the proposed approach. Each of the components will now be discussed in more detail.

[1] https://www.mapbox.com/use-cases/turn-by-turn-navigation/
[2] https://developers.google.com/maps/documentation
[3] https://developer.here.com/
[4] https://navitia.io
[5] http://opentripplanner.org
[6] http://project-osrm.org/
[7] https://itinero.tech
[8] https://planner.js.org

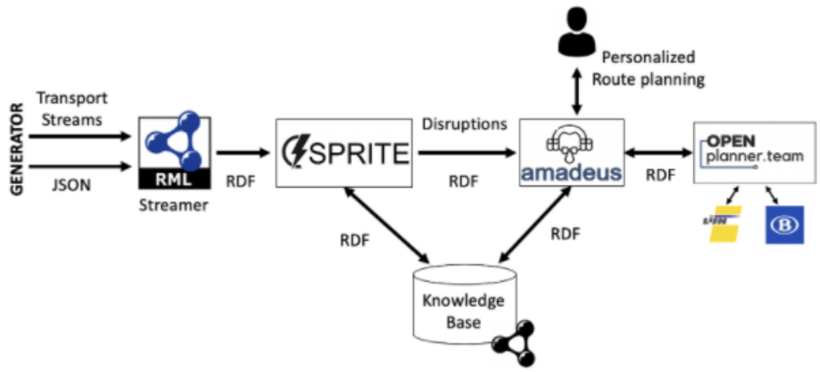

Fig. 1. Overview of the different components in our context-aware route planner.

3.1 Personalization

To integrate user preferences and profile information, we exploit the Amadeus workflow engine[9], built on top of eye, i.e. an N3 reasoner[10]. Amadeus allows the composition of dynamic workflows, here the workflow composition is the composition of traject segments. Amadeus accepts requests from users that want to travel to a certain location while performing a set of goals. It will use their profile information to link the route segments that match the preferences of the user. Amadeus will investigate the best order to link the various route segments, such that the least amount of time is spent traveling while complying with the preferences of its users. Thus, Amadeus figures out which route segments can be taken, the mode of transport and the best order to take them in. The exact details of filling in the route segments are left to the route planner.

3.2 Route planning

The route planner, Planner.js, will accept the selected route segments from Amadeus and connect with the available public transport APIs to find the exact schedules. It can also calculate exact routes when a user is walking, biking, or driving.

3.3 Dynamic disruption detection

When a suggested route has been selected, traffic is being monitored to detect traffic disruptions that will influence the suggested route. For example, when a user has to take a bus connection after a train ride, the traffic that influences the bus ride will be monitored once the user nears the bus station. If the traffic data streams indicate a disruption regarding the bus ride, alternatives can be suggested.

To avoid waiting time, the platform will monitor real-time traffic streams and infer which traffic events in the stream will require a recalculation of the suggested routes. As

[9] https://github.com/IDLabResearch/AMADEUS-workflows
[10] https://github.com/josd/eye

depicted in Figure 1, we use the RML Streamer [7] to map GTFS (i.e. a specification of transport updates) related streams to the transport disruption ontology[11]. The stream reasoner C-Sprite [2] is used to reason upon the traffic events, link them to the route connections, and infer which events will cause delays. Streaming MASSIF[12] [3] is used as a prototyping platform to seamlessly integrate the various streaming and reasoning components.

4 Demonstrator

In this demonstrator, we will showcase the following scenario:

A user Rik is currently at work in Brussels and wants to go the Ghent where he lives. We know from his profile that he likes to walk, however not longer than 15 minutes, he likes to bike, but if possible he prefers public transport. His bike is currently at the library in Ghent. While going home, he wants to visit his friend Sabine and wants to have his bike back home. He does not want to manually calculate the best order to perform his goals but expects the route planner to provide him an optimal route and adapt it if disruptions occur.

In the demonstrator, the platform will suggest the routes from Rik his work, to his home in Ghent, while visiting his friend Sabine and stopping to fetch his bike. We will showcase what happens when a disruption in traffic occurs causing to delay Rik's route. The demonstrator will showcase how the real-time streams are incorporated and disruptions are detected. Most importantly, the route planner will suggest alternatives routes, taking the disruptions into account and still complying with the preferences of the user. For the demonstrator, we use generated streams to showcase the functionality. A video[13] summarizes this demonstrator.

Acknowledgments: This research was funded by the FWO SBO grant 150038 (DiS-SeCt).

References

1. Bast, H., Delling, D., Goldberg, A., Müller-Hannemann, M., Pajor, T., Sanders, P., Wagner, D., Werneck, R.F.: Route planning in transportation networks. In: Algorithm engineering, pp. 19–80. Springer (2016)
2. Bonte, P., Tommasini, R., De Turck, F., Ongenae, F., Valle, E.D.: C-sprite: Efficient hierarchical reasoning for rapid rdf stream processing. In: Proceedings of the 13th ACM International Conference on Distributed and Event-based Systems. pp. 103–114 (2019)
3. Bonte, P., Tommasini, R., Della Valle, E., De Turck, F., Ongenae, F.: Streaming massif: cascading reasoning for efficient processing of iot data streams. Sensors **18**(11), 3832 (2018)
4. Colpaert, P., Abelshausen, B., Meléndez, J.A.R., Delva, H., Verborgh, R.: Republishing openstreetmap's roads as linked routable tiles. In: European Semantic Web Conference. pp. 13–17. Springer (2019), http://pieter.pm/demo-paper-routable-tiles/

[11] https://transportdisruption.github.io/
[12] https://github.com/IBCNServices/StreamingMASSIF
[13] https://tinyurl.com/ybddezz9

5. Colpaert, P., Verborgh, R., Mannens, E.: Public transit route planning through lightweight linked data interfaces. In: International Conference on Web Engineering. pp. 403–411. Springer (2017), https://pietercolpaert.be/papers/icwe2017-lc/
6. Dell'Aglio, D., Della Valle, E., van Harmelen, F., Bernstein, A.: Stream reasoning: A survey and outlook. Data Science **1**(1-2), 59–83 (2017)
7. Haesendonck, G., Maroy, W., Heyvaert, P., Verborgh, R., Dimou, A.: Parallel rdf generation from heterogeneous big data. In: Proceedings of the International Workshop on Semantic Big Data. pp. 1–6 (2019)