Busfinder: a personalized multimodal transportation guide with dynamic routing

Dimitris Tsolkas[1], Nikos Passas[1], Christos Xenakis[2], Vassilis Papataxiarhis[1], Vassileios Tsetsos[3]

[1]Department of Informatics and Telecommunications
University of Athens, Greece
{dtsolkas, passas, vpap}@di.uoa.gr

[2]Department of Digital Systems
University of Piraeus, Greece
xenakis@unipi.gr

[3]Mobics Ltd
Athens, Greece
btsetsos@mobics.gr

Abstract — **Multimodal routing services with public transportation are already present in many large cities across the globe. In this paper we describe a mobile guide that goes beyond existing solutions, mainly due to its service personalization and its dynamic routing algorithm. Personalization is based on knowledge engineering principles and Semantic Web technologies. The dynamic routing algorithm is based on short term history and estimators and supports cities with pre-installed fleet management systems in the public transport means.**

Keywords - multimodal transportation, dynamic routing, personalized location services.

I. INTRODUCTION

Multimodal transportation with public means of transport is seen by civil engineers as a key solution to traffic problems in large and medium sized cities [1]. Moreover, it is a way to significantly reduce pollutant emissions and improve the environmental conditions in densely populated areas. Technology can play an important role towards the wider use of public transport, by providing accurate and real-time information in a user-friendly way. An IT system providing such services to public commuters faces several challenges, such as user-friendliness, accurate and comprehensive instructions, calculation of optimal routes, personalization of results, etc. Busfinder is a system initially designed for the city of Athens, Greece, but it is applicable to any other city. Its goal is to provide a useful tool to moving commuters, by calculating optimal routes and guiding them through a mobile application. It can be interfaced to any fleet management system that monitors the mobility of buses and other public transport media. The specific services provided by the system are:

- Stop Guide: maps and lists showing all stops around the user's location.
- Line Search: the user can search for public transport lines and see their trajectories and bus stops.
- Next Bus: updates the user on the estimated time of arrival of the buses, based on current traffic conditions, bus location and speed, historical data and incidents.
- Nearest POIs: the nearest Points of Interest (POIs) are depicted on a map, based on user's location (either for moving or stationary users) and preferences.
- Route Guide: the most advanced service that performs end-to-end guidance of a mobile user based on several criteria (route duration, walking time, number of transits). The innovation of the service is that it continuously monitors the route validity and, if required, re-calculates the optimal route. Moreover, the results are fully customized for the specific user based on her profile.

Users are able to register their profile information so that the results from all services are as targeted as possible to their needs. Such information contains preferred means of transport, physical capabilities, Points of Interest (POIs) etc. and is correlated with the respective semantics of the public transport lines through knowledge technologies. A major differentiator from other similar systems is the algorithmic substrate that ensures accuracy of the results. A combination of graph search algorithms, rules and estimators are fed with the metadata collected for the user profile and the status of the public transport network. The outcome of such dynamic process is always-up-to-date guidelines for the most efficient and convenient transportation of the user.

II. RELATED WORK

There have been implemented many transportation guides so far. Some of them inform the user on the time of arrival of specific buses (e.g., Nextbus [2], Bongo [3], OneBusAway [4]). Others exploit advanced routing/planning algorithms for multi-criteria multimodal route calculation (e.g., OPTI-TRANS [5]). Busfinder differs in several aspects from the aforementioned systems. Firstly, its functionality is a superset of the features implemented in each individual system. Specifically, it calculates optimal routes based on multiple means of transport and multiple criteria, provides real-time status of the transport network, shows points of interest (e.g., bus stops) on maps, and provides routing instructions.

Another main asset of Busfinder is the adopted two-layer route estimator. Generally, a public transportation network is depicted by a graph with time-dependent weights on its edges. In that way, the problem of finding the required routes is transformed to a time-dependent shortest path problem. Noted that this problem is a high complex problem and it is classified to N-P complete (NPc) problems [6]. Dreyfus [7] has developed many solutions with no consideration of users' preferences. Users' preferences are involved in the generic algorithm provided [8], however, the solution is highly complex. In contrast to these approaches, Busfinder is based on a two step scheme. In the first step, a conventional route selection algorithm takes place based on user's personalization without involving to the public transport network time-depended or stochastic parameters. In the second step, the set of solutions provided by the first step is evaluated taking into account real-time information and time-depended parameters. In that way, the complexity of solving an NPc problem is avoided and also the time-depending parameters are included in the final solution. An additional difference is that Busfinder can dynamically re-calculate routes that have been already proposed to the commuter, thus being a very helpful "trip advisor". This is described in Section III. Another point of difference is that it takes into account several aspects of the user profile and can support even more user (or common sense) criteria during route calculation and selection. Such functionality is enabled by modern knowledge engineering tools, such as ontologies and rules, as described in Section V.

III. Overall System Architecture

A. Requirements Analysis

In order to build a system that satisfies the needs of the real users, we performed an extensive requirements elicitation process. Specifically, we surveyed various types of potential users: students (who are not usually car owners), tourists, people with disabilities and random users (mostly visitors of Ploigos.gr, a web site with geo-information services owned by Mobics Ltd). In total, a number of 313 subjects answered to the respective questionnaires, and another 21 subjects commented on the use cases of the services provided by the system. Some major conclusions drawn from the collected responses are:

- The route selection criteria for people with disabilities are the number of transits, avoidance of congested transport media, and trip duration.
- Students and younger commuters request instructions mainly at the beginning of a trip. Afterwards, they can easily follow the instructions and do not need further assistance. What is most important is reducing their waiting time at bus stops.
- City visitors and tourists find it difficult to use public transport in Athens. Most of them try to avoid buses and other similar means and are, in general, willing to pay for a taxi.
- Line Search and Route Guide are the most useful services, with Stop Guide following.

B. System Architecture

The overall system architecture is depicted in Fig. 1. The user through her mobile device, accesses the services provided by the platform. The system front-end (end user application) is a native mobile application running in all modern smart phone platforms. The back-end is composed of the core service execution engine, where all algorithms are executed, the database, where all service data and metadata are stored, and an open interface to fleet management systems (FMS). The system is agnostic to the actual FMS used as long as it implements an Application Programming Interface (API) defined in the context of the project. The system is also open to data input from other third parties such as drivers, users or traffic and road status monitoring systems operated by public authorities.

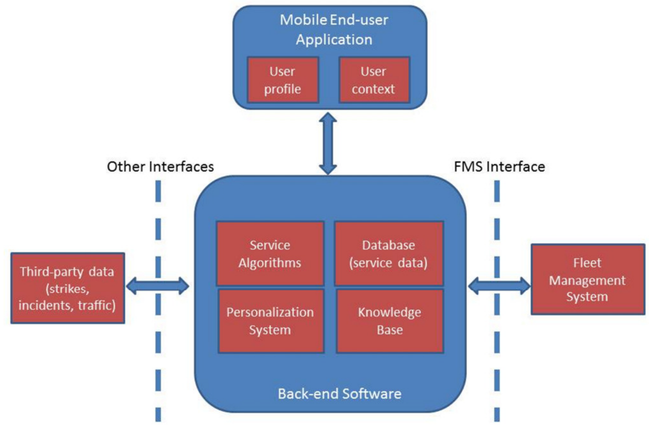

Figure 1. Busfinder High-level Architecture

IV. Dynamic Routing Algorithm

In the core of the proposed mobile application for guided traveling by public transports is a dynamic route estimator. Taking into account the high complexity of routing problem, especially in the case that real-time parameters and personal preferences are considered, a two-step solution is provided. The first step takes advantage of past measurements and statistical methods to estimate the duration of a transition providing a list of possible routes that can be selected. This list is optimized by a dynamic step incorporating real-time measurements taken by a Fleet Management System (FMS).

A. OpenTripPlanner (OTP) platform

For the first step the OpenTripPlanner (OTP) platform [11] is adopted. OTP is an open-source multi-modal planner with the following abilities:

- Plans multi-modal walking, biking, and transit trips
- Takes road type, bike lane, and elevation data into account
- Shows elevation maps for bike trips
- Imports data from GTFS (General Transit Feed Specification) [12]
- Plans trips in about 100ms in a moderate sized city
- Proposes a list of routes based on user's preferences.

OTP is a reliable and efficient planner in the case that the effect of real-time parameters such as traffic/weather conditions, demonstrations, and car-accidents is assumed negligible. However, noticeable impact of real-time parameters can be observed in the case that the selected route includes transportations by bus (or trolley) where there is no rail to ensure a normal course from starting to destination point. To cover this gap, a dynamic step is proposed for cities with pre-installed FMSs in public transport.

B. *Dynamic Route Optimization (DRO) algorithm*

The dynamic part of the solution (Fig. 2) is called *dynamic route optimization (DRO)* algorithm and combines routes proposed by OTP with real-time data provided by an FMS. The target is to provide reliable estimations to route queries made by users waiting at a bus stop or being on their way to take the bus.

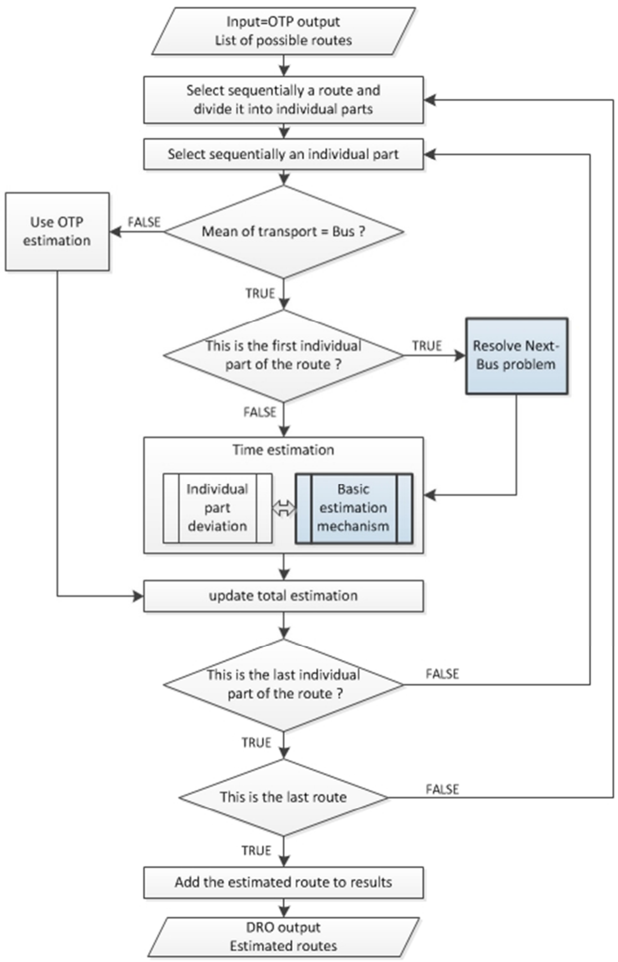

Figure 2. Flowchart of DRO algorithm

Each route produced by OTP is divided into independent segments referred to individual transitions from a boarding to a landing stop using a specific mean of transport. In the case that the mean of transport is bus or trolley the time duration from boarding to landing estimated by OTP is not reliable and, thus, the DRO algorithm is used. In an individual bus segment the impact of external factors (e.g., a car-accident) to the route duration is not uniformly distributed to the total route. For example, a car-accident has stronger effect to the time needed traveling from boarding stop to a bus stop near the accident point, than traveling away the accident point to a landing stop. Incorporating these variations in the proposed algorithm, the estimation is based on a core estimator provided in the next subsection estimating separately the durations between sequential bus stops. Results are merged providing the estimated duration from boarding to landing stop. As side effect of this approach, extra information for arrival times to interim stops can be offered to users giving a better understanding of the time needed for the trip. This information can be essential for a user that decides to reschedule its trip being on the way to destination.

1) *Core estimator*

The core estimator estimates the duration between two sequential bus stops by observing the two previous buses of the same service line passed from the same bus stops. The estimation is based on the idea that the duration is expected to be proportional to the time measured for the last bus of the same service line that passed from these stops and, also, follows the trend resulted by the comparison of the last two sequential passes made by buses of the same service line.

TABLE I. DYNAMIC ROUTING OPTIMIZATION NOTATION

Symbol	Description
s	Bus stop identifier
m	Bus identifier
T_s^m	Arrival time of bus m to bus stop s
$\Delta_{s,s+1}^m$	Inter-arrival time between bus-stops s and s + 1 for bus m

Let the notation shown in Table I, and also, denote by $\widetilde{\Delta_{s,s+1}^{m+1}}$ the inter-arrival time between bus stop s and $s+1$ for the expected bus $m+1$. Measurements: T_s^m, $\Delta_{s,s+1}^m$, T_s^{m-1}, $\Delta_{s,s+1}^{m-1}$ are extracted by the FMS, and a linear function $f(X) = A \cdot X + B$ is used to estimate $\widetilde{\Delta_{s,s+1}^{m+1}}$ as follows:

$$\widetilde{\Delta_{s,s+1}^{m+1}} = f\left(\widetilde{T_s^{m+1}}\right) = A \cdot T_s^{m+1} + B \quad (1)$$

where,

$$\widetilde{T_s^{m+1}} = \widetilde{T_{s-1}^{m+1}} + \widetilde{\Delta_{s-1,s}^{m+1}} \quad (2)$$

$$A = \frac{\Delta_{s,s+1}^m - \Delta_{s,s+1}^{m-1}}{T_s^m - T_s^{m-1}} \quad (3)$$

$$B = \Delta_{s,s+1}^m - A \cdot T_s^m \quad (4)$$

and, thus,

$$\widetilde{\Delta_{s,s+1}^{m+1}} = \left(\frac{\Delta_{s,s+1}^m - \Delta_{s,s+1}^{m-1}}{T_s^m - T_s^{m-1}}\right) \cdot \widetilde{T_s^{m+1}} + \left(\Delta_{s,s+1}^m - \left(\frac{\Delta_{s,s+1}^m - \Delta_{s,s+1}^{m-1}}{T_s^m - T_s^{m-1}}\right) \cdot T_s^m\right) \quad (5)$$

The initialization of the recursive procedure shown in equation (2) requires the arrival time to the bus stop where the expected bus $m+1$ is currently located. This extra measurement is denoted by $T_{s_c}^{m+1}$, where s_c denotes the current bus stop, and it is also extracted by the FMS. The procedure adopted by the core estimator is summarized as follows:

- Ask FMS for $T_{s_c}^{m+1}$, T_s^m, $\Delta_{s,s+1}^m$, T_s^{m-1}, and $\Delta_{s,s+1}^{m-1}$ measurements
- Estimate $\widetilde{T_s^{m+1}}$ value either by using $\widetilde{T_{s-1}^{m+1}}$ and $\widetilde{\Delta_{s-1,s}^{m+1}}$ values stored by previous estimations or by using recursively equation (2) and $T_{s_c}^{m+1}$ measurement
- Use equation (5) to estimate $\widetilde{\Delta_{s,s+1}^{m+1}}$ value

This estimator is used for any route segment that the mean of transport is bus or trolley. However, if the first route segment is served by a bus or a trolley the estimator utilizes an extra component dealing with the next-bus problem as described in the following subsection.

2) Incorporation of the next-bus problem

The next-bus problem is defined in the case that a user is on the way to take the bus; there is the constrain that it must be at the bus stop before a bus of the preferable service line passes introducing an uncertainty on which bus will serve the user. DRO algorithm deals with this problem by adding an extra component before the core estimator (Fig. 2). The arrival time of the user is compared with the arrival time of the first bus expected to arrive to the boarding stop. In the case that the user is expected to miss this bus, DRO asks FMS for the measurements of the next bus while the waiting time at the bus stop is added to the total estimation.

V. SERVICE PERSONALIZATION PROCESS

A. Overall Process

The personalization process relies on a semantics-based architecture that a) captures and represents all related metadata and b) correlates them in order to return valid and accurate results to the user. The components of such architecture (shown in Fig. 3) are:

1. User Ontology, which captures all user profile elements related to an individual's transportation.
2. Transport Ontology, which describes the elements of the transport network (stops, lines, etc).
3. Content Ontology, which describes all other data served to the user (POIs).
4. Reasoner, which performs classification of the data.
5. Rule engine and rules that perform the filtering and ordering of the results, returned by the service algorithms.

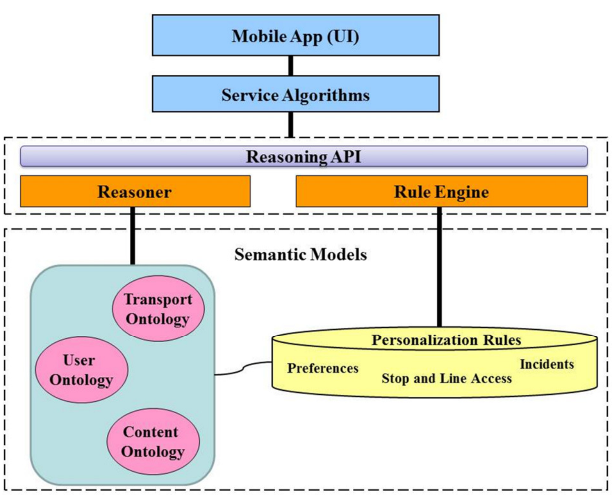

Figure 3. Personalization architecture

The personalization process takes two forms: a) filters out certain results (e.g., routes) due to their inappropriateness, and, b) sorts the results according to user preferences or common sense rules. Fig. 4 shows this process applied to the routing service. The routing service algorithm (DRO based on OTP engine) takes into consideration some hard restrictions of the user (e.g., handicaps, acceptable means of transport, maximum number of transits), and its output is a set of "compatible" routes. In case this set has more than one elements, they are further sorted based on other rules defined by the user or the system administrators (e.g., bus delay due to an accident).

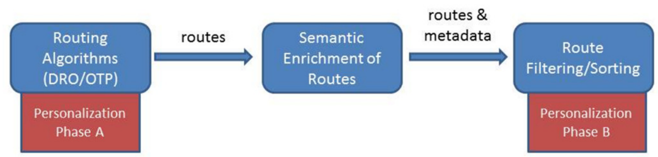

Figure 4. Personalization proces

It should be noted that such approach for separation of the routing algorithm and personalization has already been applied to other applications [9] and has some very interesting advantages over specialized algorithms with embedded personalization logic [10]:

- No need to design a very complex (and hardly extensible) graph search algorithm for supporting such multi-criteria optimization task.
- Model-driven design of personalized applications, which eases the system implementation and further adaptation to real-world conditions.
- Efficient reasoning with subsets of First Order Logic and scalability of the solution.

B. System Models

A set of data models have designed in order to capture the semantics of the entities involved in service provision. The first of them is the Transport Ontology, the main hierarchy of which is depicted in Fig. 5. Its main classes represent the stops of various types of means of transport along with their operational and acessibility status [13].

The second model is the User Ontology [13] that captures all aspects of a user profile in a declarative way (through classes and binary relationships). The instances of this ontology for a specific user represent her abilities (e.g., physical), her demographic data and her preferences. The initial instances are asserted by the user through her mobile device and then the reasoner infers the user profile and classifies her to a more specific class of the ontology.

In order to better demonstrate the usage of the aforementioned ontologies during the service personalization process a route selection example follows:

Let us assume that Bob is a user with mobility impairments and is located in Omonoia Square (Athens) and wants to reach Ampelokipoi. There are three possible (loopless) paths returned by the routing algorithm:

Route 1: Use trolley #8, arrival time: 09:13
Route 2: Use bus #22, arrival time: 09:15
Route 3: Use the Red and then the Blue Line of metro, arrival time: 09:18

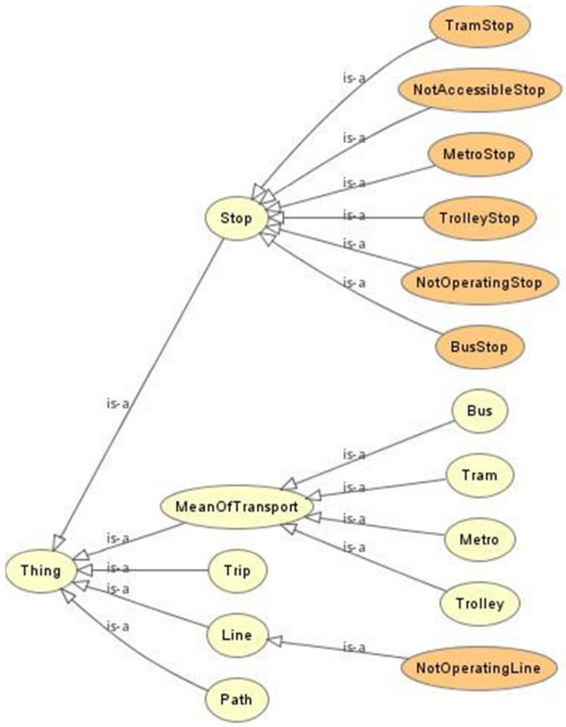

Figure 5. Taxonomy of Transport Ontology

Let us also assume that the system administrator has declared the following information in the knowledge base of the system (part of the ontological user model):

Ontology definictions (a.k.a. TBox)

WheelchairedUser ≡ hasAbilityToWalk.BadAbility ⊓ hasAbilityToUseWheelchair.GoodAbility

BadAbility ≡ Ability ⊓ hasQuality.bad

GoodAbility ≡ Ability ⊓ hasQuality.good

Moreover, the following instances are asserted through the registration of the user to the service. Specifically, the user fills in a simple form with his personal data and the respective ontological instances are automatically created by the system.

Ontology instances (a.k.a. ABox)

hasAbilityToWalk(bob, abilityToWalk_bob)
hasAbilityToUseWheelchair(bob, abilityToUseWheelchair_bob)
hasQuality(abilityToWalk_bob, bad)
hasQuality(abilityToUseWheelchair_bob, good)
AbilityToWalk(abilityToWalk_bob)
AbilityToUseWheelchair(abilityToUseWheelchair_bob)

AbilityToWalk ⊑ Ability

AbilityToUseWheelchair ⊑ Ability

Additionally, the returned routes are represented as a set of stops through the Transport ontology:

hasIntermediatePoint (route1, faros_stop)
…
hasIntermediatePoint (route2, gefira_stop)
…
hasIntermediatePoint (route3, astynomia_stop)
…
containsStop(22, faros_stop)
containsStop(8, gefira_stop)
isAccessibleStop (faros_stop, false)
isAccessibleStop (gefira_stop, false)

Given the accessibility constraints of the specific user, the personalization rules (already declared by the administrators into the knowledge base of the system) will be triggered:

user: WheelchairedUser(u) ∧
transport:NotAccessibleLine(line) →
transport:lineExcludedFor(line, u)

transport:Path(p) ∧ transport:hasIntermediatePoint(p, s) ∧
transport:containsStop(line, s), transport:isAccessibleStop (s, false) → transport:NotAccessibleLine(line)

After execution of the aforementioned rules, only Route 3 will be proposed to the user, since the other two are not accessible. This is due to the fact that both trolley #8 and bus line #22 are not accessible since the returned paths that involve these lines do not contain accessible intermediate stops).

One could claim that the aforementioned filtering process could be performed rather easily without using so formal knowledge technologies. This might hold true for the specific example, but for more complex cases, involving more personalization criteria (e.g., preferred means of transport, preferred bus lines, exclusion of paths that involve stops in the city center due to strikes, inclusion of stops where bicycles are allowed to carry with, prefer stops with many nearby transit options etc.), route selection can become difficult to model, to extend and to program.

VI. Conclusions And Future Work

In this paper a new approach to multi-modal transportation planning and intelligent transportation services is presented. The proposed system exploits real-time information and transportation semantics so that optimal services are delivered to the user. The project implementation is still in progress and a small-scale pilot is planned to take place on Fall 2012. Future work, that is not currently covered by the project but is within the interest of the consortium, is the implementation of a voice interface, usable by people with disabilities, and an integration of a case-based reasoning system for dealing with cities that do not have FMS in their public transport but have historical data about road traffic conditions.

ACKNOWLEDGMENT

This work has been co-funded by the Greek General Secretariat for Research and Technology (GSRT) and the EU (Project Code: 24SMEs2009). The project is jointly implemented by MOBICS Ltd, AMCO Advanced Technologies, SPACE Hellas, and University of Piraeus.

REFERENCES

[1] Todd Litman, Evaluating Public Transit Benefits and Costs, Best Practices Guidebook, Victoria Transport Policy Institute, February 2012

[2] NextBus, http://www.nextbus.com/homepage/, 2012

[3] Bongo, http://www.ebongo.org/, 2012

[4] OneBusAway, http://www.onebusaway.org/, 2012

[5] OPTI-TRANS, http://www.optitrans-fp7.eu/, 2012

[6] H.M. Safer, J.B. Orlin, "Fast approximations schemas for multicriterial combinatorial optimization". *Technical Report No 3756-95* Cambridge, Massachusetts Institute of Technology, Sloan School of Management (1995).

[7] S. E. Dreyfus, "An Appraisal of Some Shortest-path Algorithms", Operations Research, 17, 1969, pp. 395-412.

[8] J. Koszelew, The Theoretical Framework of Optimization of Public Transport Travel, 6th International Conference on Computer Information Systems and Industrial Management Applications IEEE-CISIM'07, pp 65-70, 2007.

[9] V. Papataxiarhis, V. Tsetsos, I. Karali, P. Stamatopoulos, and S. Hadjiefthymiades, "Developing rule-based applications for the Web: Methodologies and Tools", chapter in the "Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches" (Eds. Adrian Giurca, Dragan Gasevic and Kuldar Taveter), Information Science Reference, 2009.

[10] V. Tsetsos, V. Papataxiarhis, and S. Hadjiefthymiades, "Personalization based on Semantic Web Technologies", chapter in "Semantic Web Engineering in the Knowledge Society" (Eds. Jorge Cardoso and Miltiadis Lytras), Information Science Reference, October 2008.

[11] https://github.com/openplans/OpenTripPlanner/wiki, 2012

[12] https://developers.google.com/transit/gtfs/reference,2012

[13] http://www.busfinder.gr/links