Multi-Modal Transportation Advisor System

João C. Ferreira, Porfírio Filipe and Alberto Silva *Member, IEEE*

***Abstract*— The current work deals with the problem of growing population in the cities and the associated mobility problems. For this problem it is proposed a Multi-Modal Transportation Advisor system, based on the integration from different data sources such as public transportation systems, car and bike sharing or car pooling. Taking into account real time traffic information the system shows best option for a user goes from point A to a point B and calculates the associated trip cost. This is part of the European Project START, where we show the contribution of a public transportation ontology for different DB integration and a best path advisor based on an appropriate Dijkstra's algorithm implementation.**

I. Introduction

THE availability of the Internet and the current development of Information and Communication Technologies (ICT) became the best way to disseminate information, inspiring the development of strategies to support tourism and culture. Additionally, the mobile guides are increasingly seen as an asset to offer an experience more appealing of visitation and interpretation to natural parks or historic sights. Technological advances allow higher processing in smaller devices, making possible the use of technologies such as GPS and Wi-Fi. In addition, the popularity of social networks, like Facebook, showed the willingness of users to share their experiences and be part of communities with similar interests.

In the European Union, over 60% of the population lives in urban areas (information from Eurostat). Air and noise pollution is getting worse each year. Urban traffic is responsible for 40% of CO_2 emissions and 70% of emissions of other pollutants arising from road transport [1]. Increasing traffic in town and city centers is responsible for chronic congestion, with the many adverse consequences that this entails in terms of delays and pollution. Every year nearly 100 billion euros, or 1% of the EU's GDP [1], are spent by the European economy to deal with this phenomenon. Several solutions have been proposed to these problems, such as a diversity of intelligent transportation systems and solutions.

Our proposed approach is a central system with a goal of creating conditions/incentives for drivers use less their own car, by giving guidance and suggestions for others transportation systems, like Public transportation, bike sharing, car sharing or even car pooling [2]. The Mobility Advisor Application is also able based on real traffic information give decision for the best option taken in account pre-defined criteria (like fastest option, cheapest and also guidance), as Figure 1, suggests:

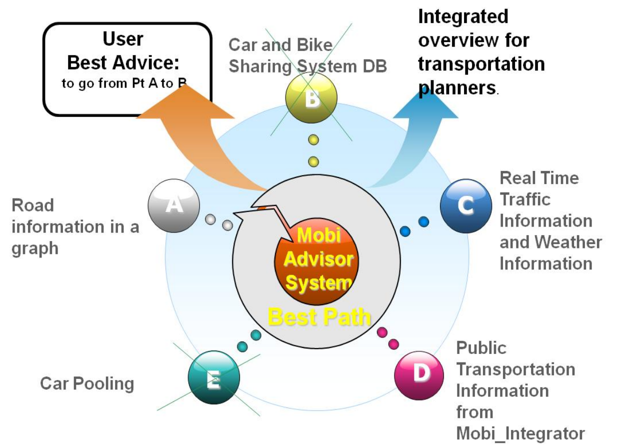

Fig. 1. Mobility Advisor System

This research paper attempts to describe ISEL participation on the Seamless Travel across the START (Atlantic Regions using sustainable Transport) project, START Project is a European Commission's Transnational Territorial Cooperation Programme with 14 partners from the UK, France, Spain and Portugal [3]. Its main mission is the establishing a transnational network of regional & local authorities to promote enhanced accessibility, giving tools to make easy to travel to, from, and around the Atlantic regions using environmentally friendly, collective modes of transport greater interconnectivity between transport systems clearer information within regional gateways, airport hubs ports & rail interchanges. Our focus on current work is the Integra project [4], whose aim is to provide a single brand that links together and provides information on the many different public transport operations across the Atlantic Area. So the system should allow the interrogation of multiple sources of information through a single interface. The questions and answers to them should reflect a single data model. The existence of a common data model takes the software applications with the difficult task of dealing with various technologies and their different relational schemas. Different public transportation system can be added with total transparency to the end user. In Integra project different public transportation data and data base schemas are being tested. Also this integration allows the creation of mobile systems oriented for tourisms purposes, other main goal of Integra project, where "low budget

Manuscript received May 26, 2011.

João Carlos Ferreira is with the ADEETC – ISEL, Lisbon Portugal (corresponding author) phone: +351 218317296; fax: +351 218317114, (e-mail: jferreira@deetc.isel.ipl.pt).

Porfírio Filipe is with the ADEETC – ISEL, Lisbon Portugal, (e-mail: pfilipe@deetc.isel.ipl.pt).

Alberto Rodrigues da Silva is with the Inesc-ID, Lisbon Portugal, (e-mail: alberto.silva@acm.org)

tourism" can be guided, to reach POI by public transportation.

This integration task is performed, as suggested in Figure 2, through a domain ontology definition for public transportation, where local public transportation operator data base are mapped.

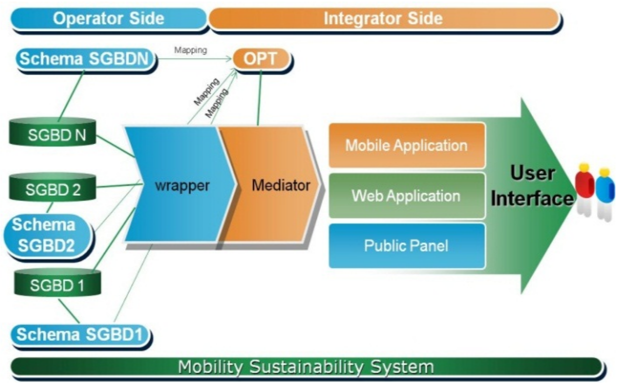

Fig. 2. Mobility Advisor System Public Transportation Data Integration

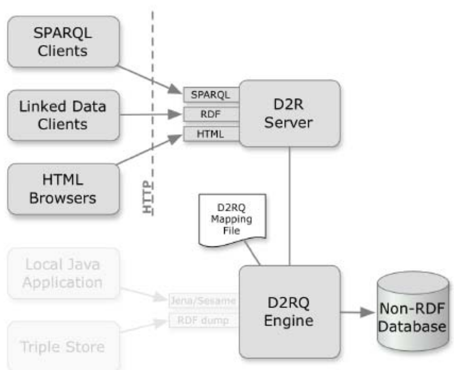

Fig. 3. Platform D2RQ used at wrapper components [5].

II. Motivation and Expted Impact

Mobile tourist guides have been in the spot light for the past decade and are becoming increasingly available in various forms to tourists visiting places. The majority of these mobile tourist guides are to be used via a constant network connection and some as proprietary standalone mobile applications installed on-device. Some are solely navigational assistants using positioning technologies for large cities offering exploratory services and others are used indoors, for example as museum guides.

There is difficulty in obtaining information about traveling to, from and within a region, even in the same city due to the diversity of transportation operators. Most of these operators have their own system and plan the routes and schedules independent of nearby operators. Still, public transport systems differ from region to region. It is therefore understandable that when reaching at a destination, even for the most traveled person, it becomes difficult to use local public transport due to poor organization of information and especially due to language barriers. In this context, denotes the scarcity of appropriate information systems to assist travelers in these regions, including providing practical information essential to understanding the operation of means of transport.

Transportation planning requires substantial amounts of data and cooperation among transportation planning agencies. This data integration with increasing availability of geographic information systems (GIS) are giving transportation planners the ability to develop and use data with a much higher degree of efficiency. Experiences among different countries can be used for local continuous improvements.

III. Data Exchange and Interoperation Model

Figure 1 illustrates the Mobi_integrator main modules, where is possible the data information integration from different operators of Public Transportation. This application output can be available through a user Mobile Device, a Public Panel, or even a Web Application. The data integration is based on a domain ontology and mappings between DB models. For these we need a ontology definition, a mediator, a local public transportation operator wrapper and the definition of mappings between local operator DB and PT domain ontology (OPT). If the operator DB is constructed under OPT the wrapper and mapping definition is not needed. Wrapper is developed at each operator side based on operator information source and is a common interface for data access. Figure 3 shows the wrapper solution based on a solution proposed by Bizer [5]. D2RQ is a declarative language to describe mappings between relational database schemata and OWL/RDFS ontologies. The D2RQ Platform uses these mapping to enables applications to access a RDF-view on a non-RDF database through the Jena and Sesame APIs, as well as over the Web via the SPARQL Protocol and as Linked Data.

SGBD Schema Publication. The mapping process between local data base and the vocabulary of the ontology (OPT) using R2RQ language. The Process is divided in the follow steps: (1) Entity definition; (2) Add proprietaries to the entities; (3) Connected entities; and (4) when necessary define conditions and aggregations. Mediator is based on MediaSpaces Mapping Framework, where we can perform SPARQL queries based on OPT.

IV. Ontology for Public Transportation (OPT)

OPT is a semantic markup language for publishing and sharing ontologies in the automobile business and its development as a vocabulary extension of RDF and is derived from DAML+OIL [6]. OPT aggregates other concepts (e.g. classes, proprieties, etc). It groups instances of other concepts that represent similar or related knowledge. On Figure 3 we show the hierarchy of OPT concepts: (1) resource is one of the bases of RDF, it represents all things described and is the root construction. It is an instance of MOF classes; (2) property, defines the relation between subjects and object resources and is used to represent relationships between concepts. Ontology class attributes or associations are represented through proprieties; (3) ontology is a concept that aggregates other concepts (classes, properties, etc). It groups instances of other concepts that represent similar or related knowledge; (4) classifier is the base class of concepts that are used for classification and is divided in: (i) datatype, a mechanism for grouping primitive data; (ii) abstract class; and (5) instance that is the base class and is divided in individuals and data values.

Property is a relation between a subject resource and an object resource. There are two types of properties: (1) object property may additionally be: transitive, symmetric, functional and inverse functional; (2) datatype. Users can relate properties by using two types of axioms: (1) property subsumptions, which specifies the extension of a property is a subset of the related property; (2) property equivalence defines extensional equivalence.

Classifier is a Class that represents a concept for grouping resources with similar characteristics. It is similar to the concept of classes defined in UML but with an orientation to object programming languages and they are set theoretic. Class description is defined as a subclass of class to solve the problem of dynamic classifier. Class can be defined in the following four ways:(1) AllDifferent states that all of its instances have different identity; (2) restriction; (3) enumeration of individuals that form the instance of a class; (4) logical class of all individuals that could have the following attributes: (i) union; (ii) complement; (iii) intersection.

From several Public Transportation data base of consortium partners and standard specification, such as SIRI (Service Interface for Real Time Information) <www.siri.org.uk> and IFOPT (Identification of Fixed Objects in Public Transport) <www.ifopt.org.uk> a common schema data was created. Figure 5 shows the main entities of the domain model. OPT is built on top of RDF using OWL (Ontology Web Language).

Also in Figure 5 represents the relationship between the Operator, Service and Contact entities. Public transport operators are represented by the entity Operator. This entity is associated with the entity that represents the Service(s) provided by a particular operator. For example in the Portuguese case, indicates the case that the operator *Carris* offers two services: bus and tram. The relationship with the entity Contact is due to the need that exists to represent the contacts provided by operators of public transport. The Contracting Service, represents the services provided by an operator of public transport that can be, e.g. rail, road, air or water (sea, river or lake). Each service will be linked to a type, represented by the entity Type, which represents the categories within each mode, e.g. rail, bus, plane or ship. Within these types may still exist variants, Variant entity. The Contact entity, represents the contact types, such as: electronic mail (email), telephone, address and website. This authority is shared by all entities Operator and POI (points of interest), because both the transport operator and points of interest have similar types of contact. POI entity are divided into categories which highlight, for example, Academic (Academic), Rest (Restoration), Monument (Museums) and others.

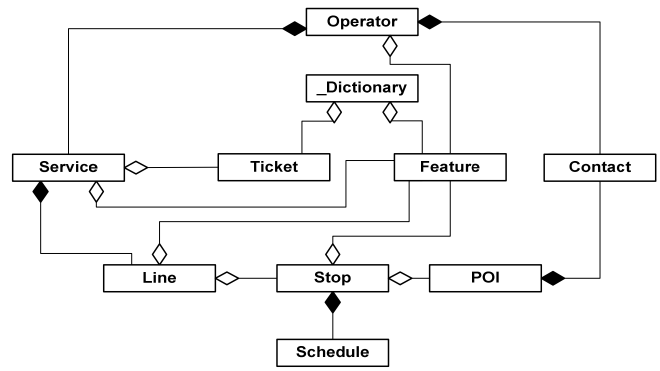

Fig. 5. Entities in the domain model

Also Figure 6 illustrates the relationship between the entities Schedule, Stop, POI, Interface and Contact. Stop entity include the attributes: name of the stop, geographical coordinates (Latitude, Longitude) zone (taking the subway as an example verifies the existence of two areas, one inside the city limits and another in the vicinity). The association of the entity with the entity Stop with POI should be relevant to the fact that giving information about the area around the station so that users from other countries have tourist information. The relationship between Stop and Schedule justified with the representation of a schedule.

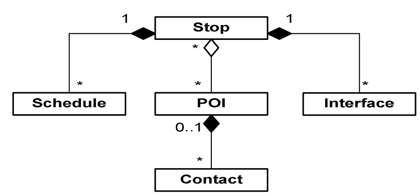

Fig. 6. Stop hierarchy

V. ISPT: Integrated System for Public Transportation

ISPT or SITP (Portuguese system name) is the prototype developed for STAR project. Figure 7 illustrates typical client operation from the client side: search for stops, search information of PT for a certain path, get price, schedules and best itineraries path options. From operator is the registration and the registration of DB schema. Two web applications are developed: one to support queries from users and the other for the management of information by operators. These applications were implemented in ASP.NET MVC platform. For more details see [7]. The Web client application was implemented to allow the user to search for routes, stops, query times and fares. Figure 8 illustrates the application menu to support these functionalities. The developed application has the basic structure of a Master Page with an application menu. The menu implemented allows navigation of the site as intended. As can be seen in Figure 9, we have Multi-Language support and a link to access the mobile devices (in Portuguese “Telemóvel”).

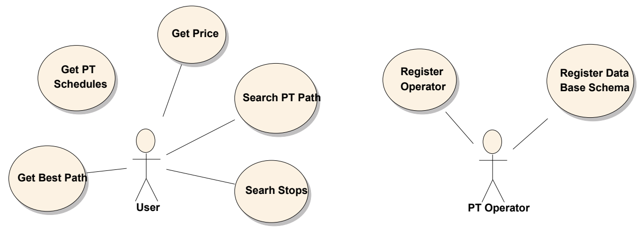

Fig. 7. Application use case

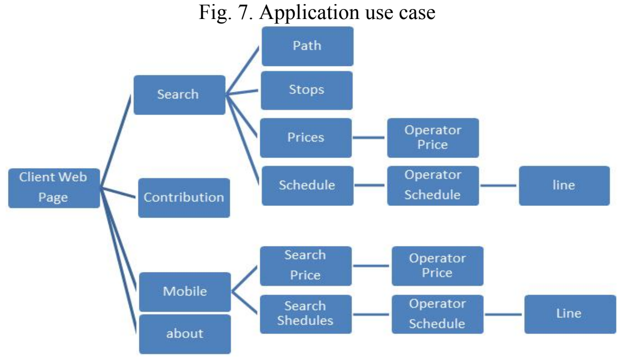

Fig. 8. Web Client Application Menu

Fig. 9. Web Client Menu

Search PT routes: To search for PT routes were considered several criteria's: The first is based on the selection of operators connected with the research. The second considers the number of transfers and price of the desired trip; and the last based on the trip duration using best path algorithm.

VI. User Guidance

A. *Identify Public Transportation Routes*

To calculate the paths we considered two phases. At first it is checked whether the stations of departure and arrival belong to the same line. So when the departure and arrival stations belong to the same line, the field IsSameLine is marked as true, otherwise it will be marked as false. Where IsSameLine be true is executed the first phase of the research journey thus obtaining the data required to design the course. These data highlight the coordinates of the stations, the price of the trip, points of interest that are near the station of arrival and other information such as schedules and features. If IsSameLine is false then runs the second phase. In the second phase, the stations of departure and arrival are not on the same line. It is therefore necessary to find a path or paths considering the available interfaces on the lines corresponding to the stations of departure and arrival. This research takes advantage of the method getAllInterfacesBetween. This method was implemented in a recursive manner to allow the taking of all possible paths between two stations. To obtain the pathways analysis was carried out taking into account the interfaces of the line of the station that is being verified. In each interface is checked if the line matches the line of the arrival station guarding the path and if so, otherwise, it uses recursion repeating the analysis to find a route. A sample calculation (s) route (s) is illustrated in Figure 10, for Portuguese Metro (Underground). In this figure we can see the paths generated between the starting point Odivelas (metro line) and the arrival point for Oriente (metro line). The first route uses only one interface (Saldanha (interface metro line)) to reach the destination while the second route uses two interfaces (Campo Grande (interface metro line) and Alameda (interface metro line)) to reach the same destination. In the case of route X as there is no interface that the route reaches the destination is not valid. The method getAllInterfacesBetween to be implemented recursively have to take care to avoid cycles. To prevent cycling was implemented VerifyCicleInPath method that checks if the route has passed or not by a given interface. If so this path is excluded. The route will be shown to the user by taking advantage of Google Maps API.

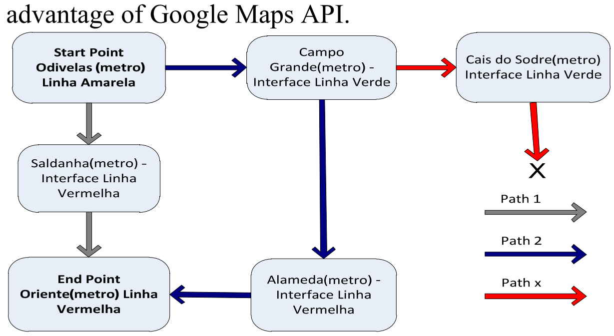

Fig. 10. Path Search using interfaces, a case from Portuguese Metro

B. *Calculate the price of the trip*

To calculate the price of the trip was defined InfoServiceZone the class containing the service identifier and zone. This class is used in the method ObtainArrayTickets to get a list of services and areas used on a trip, or a list in which each position corresponds to an instance of InfoServiceZone. This represents an instance of InfoServiceZone stage of the journey. After obtaining the list of stages of the method is invoked CalcPrice. This method evaluates the list of steps in order to return the price of the journey. Analysis of the list of steps highlight the fact that the next step is to compare with the previous order to ascertain whether there was an exchange zone or service. This comparison can then calculate the price given the exchange area or service over the route.

As mentioned before a route is divided into stages. Figure 11 shows that the route is only done by a service (IdService = 1), so the price calculation takes into account the zones (Zone) of the steps. Once the trail starts in Area 2 the price to be added is € 1.10. When you perform the initial exchange of steps, from Step 1 to Step 2, it is verified that the area is below the Step 2 from Step 1. The fact that this happens will not affect the current price. If the area of Phase 2 was superior to that of Step 1 then the price would be the sum of the pricing zone subtracted from Step 2 to the pricing of the zone in Step 1. In return for the second stage, from Step 2 to Step 3 is to be noted that the pricing is the same zone is not changed. Please note that these calculations are made for services where the services allow transfers in your network. If the services do not allow transfers in calculating the price is the sum of the prices of each step. Assuming that the service (IdService = 1) does not provide the transfer on your network with a single ticket is the price of 2.70 € (1.10 € + 0.80 € + 0.80 €). The search page provides the user stops the ability to search stops in a certain radius. To implement this research used the Google Maps API, where the user can select a point on the map to determine if there are stops and a certain radius around that point. The radius will be chosen by the user through a DropDown available for this purpose. If there are stations within the radius desired markers will be placed with the operator logo on the coordinates where they are. The user can view station information by interacting with the markers. The information provided is related to service characteristics with the characteristics of the station, as well as information on upcoming departures of service. It is also available to the user the possibility to observe the line where the station is located. To make this information available we used the technique also taking advantage of AJAX and ActionResult Controllers. Thus, whenever a request is made, the Controller responsible for the care of returns information about the form of a StringJson which is treated in the Client (JavaScript). Client-side (JavaScript) are methods that deal with the response and put the information available to the user.To determine the stations that lie within the perimeter defined by the user, the method GetAllStopsInArea. To find the two stations are set intervals by adding and subtracting the radius of latitude and longitude of the point chosen. Once these intervals a search is performed to determine the geographic coordinates of the stations that lie within these. For timetables and fares have been released two pages that allow the user to select the times and rates depending on the desired operator.

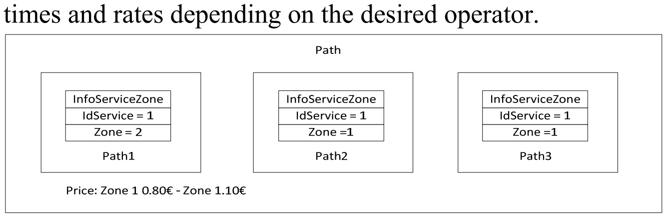

Fig. 11. Price calculation in a route

Regarding the Rates page is provided information on the tickets of a particular operator such as area, price and description. This information is represented whereas the seasons, although the same line, may belong to different zones with different price lists. Thus, the user gets the price depending on the area from where or where it goes. As an example, the journey from Campo Grande station (metro line) station Odivelas (metro line). In this case, the patient away from a station belonging to a zone must purchase a ticket for zone 2 since it will move to a station that belongs to zone 2.

C. *Real Time Information and Best Path*

Real time information is available from road concessionaries but external information access is most of times denied. To solve this problem in a project at ISEL [8] we create a web crawler to pick traffic information from specialized sites and from a pre-defined heuristics an XML file with traffic information is created. The XML file is an approach or future data integration from different source providers. In Figure 12 is illustrated the process of XML file with road information oriented to a geo-reference graph.

Arc	Coordenates (lat,long)	Name
1	38.927410, -9.011100	Alhandra
2	38.894050, -9.048801	Alverca
3	38.848341, -9.089346	Santa Iria
4	38.790796, -9.114538	Sacavem
5	38.787406, -9.105337	Portela/Moscavide
6	38.717140, -8.948650	Montijo
7	38.740385, -8.931608	Alcochete
8	38.642651, -9.049044	Lavradio
9	38.607083, -9.019177	Nó IC21-IC32
10	38.583166, -9.017658	Coina
11	38.589418, -8.864434	Lau
...	...	...
50	38.689940, -9.177090	Pte 25 Abril
51	38.700951, -9.424789	Cascais
52	38.717284, -9.385188	Golf do Estoril

Arch	Road	Lenght (m)	Node Max. Speed (Km/h)
{1,2}	A1	5000	120
{2,3}	A1	6200	120
{3,4}	A1	7000	120
{3,5}	IC2	8500	80
{5,49}	IC2-A12	6600	120
{49,6}	A12	9900	120
{6,7}	IC3	4500	120
...	...	...	
{37,39}	A8-A9	11500	120
{39,40}	A9-A10	5800	120
{39,2}	A9-A1	3300	120

```
<graph>
        <node id="1" name="Alhandra" lat="38.927410" long="-9.011100"/>
         <node id="2" name="Alverca" lat="38.894050" long="-9.048801"/>
        <node id="3" name="Santa Iria" lat="38.848341" long="-9.089346"/>
        <node id="4" name="Sacavem" lat="38.790796" long="-9.114538"/>
        <node id="5" name="Portela" lat="38.787406" long="-9.105337"/>
        ...
         <node id="51" name="Cascais" lat="38.700951" long="-9.424789"/>
        <node id="52" name="Golf do Estoril" lat="38.717284" long="-9.385188"/>
        <node id="53" name="Pontinha" lat="38.768703" long="-9.209198"/>
         <arc id="1" name="A1" node1="1" node2="2" length="5000" maxSpeed="120"/>
        <arc id="2" name="A1" node1="2" node2="3" length="6200" maxSpeed="120"/>
         <arc id="3" name="A1" node1="3" node2="4" length="7000" maxSpeed="120"/>
         <arc id="4" name="IC2" node1="3" node2="5" length="8500" maxSpeed="80"/>
        <arc id="5" name="A12" node1="5" node2="49" length="6600" maxSpeed="120"/>
        ...
        <arc id="68" name="A10" node1="39" node2="40" length="5800" maxSpeed="120"/>
        <arc id="69" name="A9" node1="39" node2="2" length="3300" maxSpeed="120"/>
        <arc id="70" name="Eixo Norte-Sul" node1="18" node2="17" lenght="5000" maxSpeed="90"/>
</graph>
```

Fig 12. Semi-automatic creation of a XML file with road information to be used for graph creation.

We use WebNews Crawler (version 1.0) written by Vladimir Poroshin, configured to pick traffic information from TVI web site <http://www.tvi.iol.pt/transito.php>. Information about node where checked against a heuristic table, where a conversion factor (CVF) will reduce the node speed traffic (if the rode is blocked, with no traffic flow, the CVF is zero). Figure 13 illustrates this process. The result is stored in a adjacency matrix, where the number a_{ij} represents the cost of going from i to j. To determine the best path between two points on a map, it creates a graph representing the map, where the arcs represent roads and nodes represent intersections or traffic areas. It applies an algorithm on the graph to find the path with less weight and faster between the two desired points.

The lead weight is determined based on the distance to us, at full speed and was introduced a weighting factor based on the real traffic information loaded on the system. The weight of an arc is basically the average time in seconds that is needed to be done, for this is the formula used (L / V) * 3600, where L represents the size of the graph in Km and V's top speed. For all arcs there is a speed limit which serves as the basis for the weight of this case there is no traffic information. When there is traffic information for a particular arc, the weight of this is affected because the traffic influences the speed. For example, if an arc with heavy traffic speed limit of 90km/h, given the existing traffic, the reduction factor of the maximum speed will have a value set between 0 and 1 based on current traffic information, causing the increase of the time required for this arc to be traversed as soon as your weight increases. If the transit is cut off, the factor of speed reduction will have the value 0 and the maximum attainable speed for low 0Km/h that causes this arc has infinite weight.

If traffic is proceeding smoothly and without any problems, the factor of speed reduction will have a value and a maximum attainable speed becomes equal to the speed limit, which makes this bow has the lowest weight possible, translating into a reduction in the meantime the arc needs to be traversed. This weight can integrate also public transportation information and a price associated with CO_2 emission of private transportation. User can choose the impact parameters based on their strategy, save money, commodity or even save time. See end year project at ISEL [5] for a complete description of this process.

Example: Path: A1 (Alhandra – Alverca). Maximum speed allowed on this section: 120 km/h (motorway), but traffic information shows average speed is 84 Km/h. Distance is 5Km, so Arch weight = (Arch distance / Average speed) x 3600 s.= (5000/84000) x 3600 = 0,0595238 x 3600 ≈ 214 s.

Let's consider public transportation takes more or less two times more 400 s. Transportation price is 1€ and car transportation is 2€ including the price for CO_2 emissions and motor price. If drivers choose weight factor for time and price the same, then new arch weight are: car 214x2=428 and transportation remains in 400. In this case is approximately the same but if we put parking price of car in the city system will increase again the arch weight for private car and then the system will suggest the public transportation, giving table time, guidance for next stop.

To determine the best route or the quickest route was implemented Dijkstra's algorithm, where its running time is O (N_2), N being the number of nodes in the graph. The graph to be used by Dijkstra's algorithm, is represented by an xml file. The best way to understand the quickest way, will be determined using the Dijkstra algorithm. In our web application there is a class called Dijkstra, which is where the algorithm is implemented. In our implementation we used the following data structures: (1) An array of nodes (nodes) that will have all nodes in the graph; (2) A list of integers (unsettledNodes), which holds the id of the nodes for which has yet found the best path to the source node; (3) A list of integers (settledNodes), which holds our for whom already found the best path to the source node; (4) A Dictionary (shortestDistances), which is the key to an integer representing the id's of nodes and has a value, other integer that represents the best way to estimate the source node; (5) A Dictionary (predecessors), which is the key to an integer representing the id's of nodes and is another integer value that represents the id's of nodes in the previous best path to the source node. For a complete description of current proposed method sees [8].

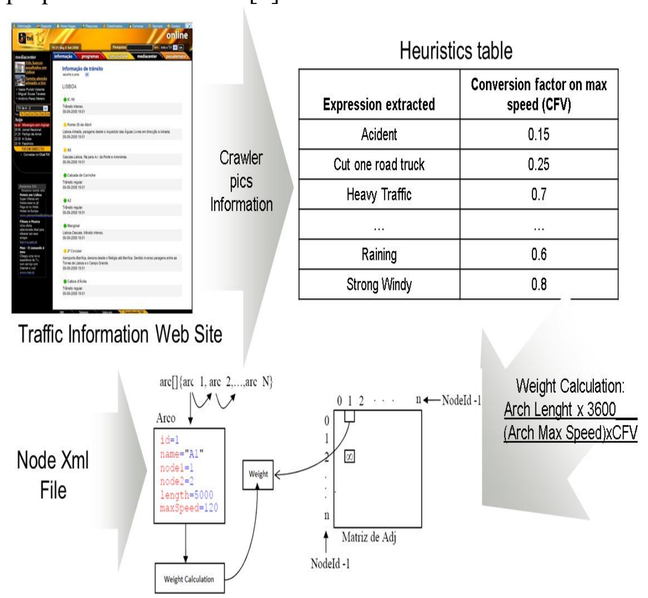

Fig. 13. Graph node weight calculation based on traffic information.

VII. Conclusions

Mobility Advisor system is a platform system integrated on START project, allowing the citizens to obtain information (eg timetables, routes and prices) on the various modes of transport (eg bus, tram, metro, train, ferry, ...) available in a particular region (e.g. Lisbon, Porto, among otjers) focusing on the integrated use of soft transport (eg, electric vehicle, bicycle, ...) and occupation of waiting time (e.g. visiting points of interest).

The main contributions of this paper are: (1) a domain ontology for Public Transportation; (2) a data integration of Public Transportation in European environment; (3) a functional prototype to query and give advice regarding the best path from a point A to B using Public Transportation; (4) the implementation in RDF/RDFS allows the use of the emergent query language SPARQL (Query Language for RDF); and (5) implementation of a best path using Dijkstra algorithm using a diversity of public transportation information sources.

References

[1] Paper, G. (2007). Towards a new culture for urban mobility. http://ec.europa.eu/transport/clean/green_paper_urban_transport/doc/2007_09_25_gp_urban_mobility_working_doc_en.pdf.

[2] Ferreira, J. T., Paulo, Porfírio Filipe (2009). Collaborative Car Pooling System. Proceedings of International Conference on Sustainable Urban Transport and Environment. Paris.

[3] START Project: http://www.start-project.eu/en/Objectives.aspx

[4] Integra Project http://www.start-project.eu/en/Integra.aspx

[5] Bizer, C. P. (2009). The D2RQ Plattform - Treating Non-RDF Databases as Virtual RDF Graphs.

[6] Miller, J., Mukerji, J. (eds.), "MDA Guide Version 1.0", OMG Document:omg/2003-05-01,

[7] D. Domingues, A. F. a. R. A. (2010). SITP – Sistema de Informação sobre Transportes Públicos. ADEETC. Lisbon, ISEL.

[8] Fontes, T. F. a. M. (2008). Determinação do melhor caminho em sistemas de transporte terrestre (BPath – Best Path). ADEETC. Lisbon, ISEL. Final Year Project.