A Collaborative GIS Solution for Public Transport

Liviu Adrian COTFAS*, Mihai Cătălin CROICU**, Dumitru COTFAS***
*Academy of Economic Studies, Bucharest, Romania, liviu.cotfas@ase.ro
**Softscape Data Solutions, Bucharest, Romania, mihaicatalin.croicu.5817@student.uu.se
***Fraser Stream Software, Vancouver, Canada, dumitru.cotfas@gmail.com

The recent years brought forward a large number of solutions for automating route finding given the increased availability of geographical data. However, such solutions rarely focus on mass transit or involve the user in submitting information in a collaborative manner to further improve the available dataset and provide additional services. The system presented here intends to fully address these issues by providing a modular, extensible collaborative one-stop-shop for public transport needs based on multi-source collaborative data inputs from both official and user-submitted sources with the usage of a flexible, genetic-algorithms based route-finding application. Implementation wise, the solution is based on an open-ended system of collaborative web-services with front-ends available on mobile, desktop and web platforms. The proposed solution will not only provide users with a powerful technical solution, but will address the theoretical concern by which the increase of available GIS data is solely used for last-mile, map-like solutions.
Keywords: *collaborative solution, public transport, genetic algorithms, geographic information systems (GIS).*

1 Introduction

A frequent problem for residents of big cities and for tourists who visit these cities alike is the difficulty of identifying an optimal route between two or more places. The difficulty consists in both the need to first determine the location of the objective (park, museum, cinematograph, university), and of choosing a convenient route to the objective. Currently, there is a multitude of solutions that allow the generation of routes for transport with private cars, but not the same applies in the case of travelling in more environmentally friendly ways like public transportation, traveling by bicycle or walking on foot. Almost all implementations are relatively superficial, neither offering support for multi-modal or multi-operator transit networks or sufficient choices or options for the passenger, not offering a clear, easy to understand route output nor offering the possibility of passenger-driven input. Furthermore, most are desktop-based applications that offer limited appeal to passengers, especially to those wishing to change options "on-the-go", while travelling. In most cases, users are forced to use maps and diagrams to compose paths – a state of affairs that is especially complex in the case of large cities where multi-modal and multi-company transit is used – leading to passengers typically selecting less than optimum routes decreasing individual satisfaction, as well as increasing congestion and ridership cost.

In addition, most applications only offer information regarding a relatively small number of objectives. Even if information is provided for a certain objective, it is usually poor lacking pictures or a detailed description.

The model proposed in this paper intends to eliminate the existing limitations by implementing a collaborative structure that allows users to easily add or modify information. Furthermore, as opposed to most existing systems, the proposed model is open, allowing data input through multiple channels and from multiple sources, through mechanisms such as web services and by implementing open standards (such as XML or RSS) allowing easy intercommunication to other solutions and data sources. This allows the system to incorporate weather information, geo-referenced information from sources such as Flickr or Wikipedia, air quality data as well as suggestions relating to attraction points or available city services (coming from either the municipality servers or advertisers' servers).

The system will allow users to generate routes for multi modal transport networks, cycling and walking. It will be accessible from a wide range of devices ranging from web to mobile solutions. In order to generate routes that better reflect the users' preferences a flexible restriction mechanism allowing multiple options will be described. Upon generating each route, cost information will be displayed, as well as fuel savings as opposed

to using a personal car, allowing users to calculate statistics for daily, weekly, monthly or yearly savings.
Implementing such a system will also have a positive effect on the environment as car traffic generates great amounts of polluting gases which accelerate the global warming process and irremediably affects both men's health and the environment.

2 Theoretical and Background Assumptions

The paradigm of collaborative software rests on two different levels – first, one can speak of collaboration on the level of user-application interfacing and participation, where content generation and content usage is done through the aggregation of what researchers called "collective intelligence" [1] - the ability of multiple unrelated sources of knowledge to generate an integrated data flow, and second, on the level of mechanisms and software that permit many types of data structures coming from multiple sources to be integrated in a meaningful application (see, for example, [1]) as well as providing the tools for the users to communicate and share information [2]. Thus, collaborative software can be understood from the point of view of the user and the way he or she interacts with the software but also from the point of view of the mechanisms that make this collaboration possible [3]. Albeit the two levels are closely inter-related, and are more-often-than-not put together with other paradigms under the vague term of "Web 2.0" [1], both levels can function independently.
On both levels, there is the problem of collaboration between different systems across both horizontal (different systems having to share process and understand data) and vertical boundaries (different levels and classes of users interacting in different ways across different cultural rifts) [4]. Such a problem (abstracting data as to allow many entry points as well as different interface methods) was debated in the pre-web service days of developing collaborative efforts, with varying success, with authors assuming that there is always a potential trade-off between complexity, modularity and usability (see [2] for an implementation model, or [5], for a theoretical and philosophical approach on collaboration in software mediated environments). However, the high scalability potential of collaborative software, as well as the flexible logical organization patterns are clearly recognized by literature [7]. Recently, however, with the advent of web-services as an integrated platform, this issue was mitigated somewhat, due to the flexible mechanisms that this technology allows in designing modules and collaboration processes. Two main collaborative implementation models have surfaced – choreography, a collaborative linkage model that allows each service to describe itself, serving mainly in inter-task and inter-party communication and orchestration, where web-services describe a directed process in a task-based organization [6]. In this, web-service oriented architecture can mold itself flexibly on providing for both horizontal and vertical usage, with abstraction becoming less of a problem. Thus, both more robust interfaces can be constructed, that would facilitate user interactions as well as an increase in the usable entry and exit points into the system.
Collaborative software on user-generated GIS already exists and is fairly well developed – with services such as Wikimapia and Panoramio providing the collaborative front-ends, and with APIs such as Google Maps, Yahoo Maps and Windows Live providing the geolocation data. However, scholars analyzing existing solutions argue that currently existing models are very specifically focused on a single task (such as geolocating pictures or geotagging) and serve solely as end-points from a data perspective (goals in itself, without making existing data further available to other services) [8]. A more open-ended approach – where different classes of objects can be attributed to real-world spaces is further considered as lacking and necessary. Thus, existing collaboratively provided spatial data, only function currently as "last-mile"/"first-mile" implementations, using other web-services for data amalgamation, but providing limited further output. Moreover, there are significant arguments that collaboratively provided geographical information can be of significant value, especially for local-oriented data gathering [9].
The proposed implementation intends to focus on achieving these goals – on one hand create a collaborative infrastructure at the level of the application business logic and mechanics, allowing both for the collection of data from multiple services and for extensibility through the usage of web-services oriented architecture and on the other hand code submitted information and use it into further processes (as a data-source for other services). Such an implementation would eliminate the last-node/first-node approach from existing collaborative public GIS solutions, by creating usable solutions with user data. Furthermore, the same implementation would also provide for the modularity and extensibility that are required

in order to permit other web-services to use and transact with it at both ends (inserting new data and mining for existing data).
The software presented below intends to create a web-service oriented model that would permit both the design of extensions for achieving different data processing goals but also the input of spatial data for providing additional data into the software. The goal of the software is to be a one-stop-shop mass transit gateway, providing route-finding facilities, street identification and spatial mapping of objectives. Both architecturally wise and from the point of view of the user, the application is constructed around the collaborative paradigm. From the point of view of infrastructure, the application is divided into "containers" of web-services working collaboratively to create a package. One such container is composed of sets of different web-services provide the input data from a variety of sources (such as weather channels through RSS feeds, Google Maps or Windows Live! Earth for GIS data and satellite imagery, public transportation route databases for route generation, user submitted spatial data for objectives and points of interest as well as pictures, news feeds for geo-referenced information, blogs for personal information aggregation etc.), working in orchestration to provide to the application logic a mash-up of information while abstracting the original sources. Further, services themselves gather data in different levels of detail – as they are built flexibly as to permit submission of data of different complexities (for example, the user-submission module is robust enough to accept data at a city-wide level, but also at a building wide level, and treat the input accordingly). Additionally, the container is extensible, so that modules can be taken out or put into place easily as additional web-services. At this level, programmers are able to collaborate in information creation.
The second level (container) of the application – the user interface logic also uses collaborative implementations designed over a web-services paradigm. Users are able to use multiple models of input that would suit them best (mobile, web-based, smart-phone based, desktop-based), with different levels of detail and complexity (including zoom functions showing information on different levels, tagging-based folksonomy for generating popular locations based on both submissions and votes, layers for showing/hiding different data, the possibility to submit geo-located data, etc.). At this level, users will be able to collaborate and participate directly in information creation. Each service at the interface level works independently from the others, yet communicates trough an identical mechanism with both the first and the third-level containers.
The third-level container – business logic, has three purposes, resolving conflicts between different information fluxes and making sense of the existing information on the basis of requests from users, grabbing and storing data through a collaborative transaction model with the data storage container and generating information (such as route-finding, aggregation of preferences, folksonomy and social information generation, creation of hierarchical structures and so on). These collaborate with each-other as well as with the other containers in order to provide a unitary set of information. Further, since the container is designed as a network of web services orchestrated by a common mechanism, it allows for extensibility, allowing programmer based collaboration at this level.
The fourth level – data storage, uses a transaction-based architecture over web-services in order to facilitate yet another type of collaborative effort, this time in terms of physical interfacing: cloud computing.
Extensions and outside hooks are possible at each level of the implementation – other, external, web services can use data existing at all levels of the application for their own purposes, making further sense and usage of pre-existing material.

3 An overview of existing implementations

3.1 Transport for London

The Transport for London application [10] is one of the most complex applications developed in order to generate routes in public transport networks. It offers a wide range of options lacking in other applications of this type such as the options for persons with disabilities (the impossibility of using staircases, elevators, moving stairs). Among the disadvantages, there can be mentioned the relatively small number of objectives and the impossibility of using the application on mobile devices. TfL's solution does not offer the possibility for any collaborative input from the user level.

3.2 Google Transit

The Google Transit application [11] aims at offering the possibility of generating routes by using public transport means in several cities from the United States of America as well as some cities in Europe and Japan. Among the applications under analysis, this offers the least options for the

user – for many cities being nothing more than a mash-up between a transit map and a city-wide street map using the Google Maps architecture. Multi-operator solutions are possible, but require the participation of all involved companies in a certain geographical space. The software is designed around a collaborative paradigm, and is offered in similar ways as with other Google Maps services - however, data input is limited to authorize mass transit companies (it is seen as a business-to-business collaborative system rather than a community driven solution).

3 Design and Implementation Architecture

The proposed solution is based on a multi-tier architecture integrated with web-services (Fig. 1). A web service is defined by [12] as a software system designed to support interoperable machine-to-machine interaction over a network, described using WSDL (Web Services Description Language). Such a solution has the advantages of permitting both a solution to rationalization issues but also a compartmentalization of the application based on the internal logic and in the same time on the external requirements, while eliminating the pitfalls of a "spaghetti architecture" [13].

Choosing this type of architecture combines the advantages of multi-level architectures with those stemming from web services such as interoperability, RESTful (Representational State Transfer) and loosely coupled models. Since we expect a large demand exerted on the system due to a large potential user base generating a substantial number of independent hits, a RESTful operational framework is considered suitable, since it eliminates the need for a permanent user-server connection, thus reducing the server load. Web services offer both scalability and extensibility for the system – as components can be installed on different servers, as well as individually upgraded and/or replaced without altering other parts of the system.

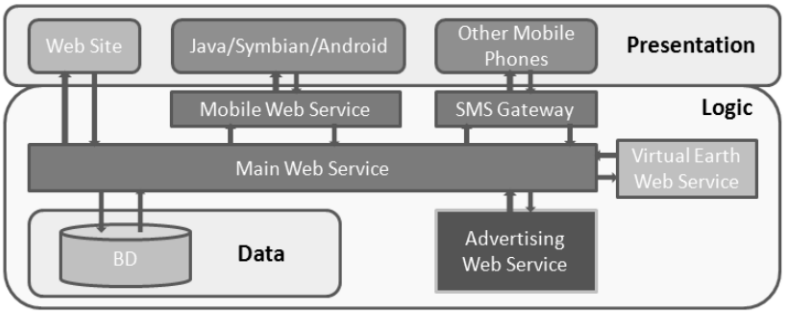

Fig. 1. System Architecture

The workhorse of the proposed architecture is the "Main Web Service", providing the core functions of the system, including the route generation algorithm as well as the links to external data sources (either provided through web services, or through RSS or XML formatted data). As such, the current implementation uses web services to take maps for route display from Microsoft Virtual Earth, while weather and air quality information are taken via XML and RSS.

The Main Web Service is called directly by the web component, by the web service dedicated to mobile devices (SMD) and by the module that manages SMS requests/replies.

The SMD is tasked to prepare data for transmission and display to devices with limited display and processing power (such as mobile phones, smart phones, PDAs, Blackberry/iPhone devices, etc.). The SMS Gateway is tasked to translate text messages with route generation requests into data that can be used by the Main Web Service and transform the replies received into text messages that can be sent through the mobile network. From a user-centric point of view, access points to the system are either through the dedicated website (for desktop use and web-enhanced smart phones), through applications developed for mobile platforms and through SMS messages.

The extensive use of web services allows an easy integration of new web services in the system. For example an Advertising Web Service can be easily integrated to provide location based advertising.

4 Database Structure

In order to generate routes, data will be stored regarding stations (such as GPS coordinates and station/stop name) as well as available transport modes (mode name/no., mode type, interval, etc.). Available connections between two consec-

utive stations, as well as distance and time needed for traveling the distance between the two are stored through the Links table. Further, considering that between two stops there is the possibility for more than one mode of transport to exist, the association between links and transportation modes is done through the Links2Modes table (Fig. 2).

The proposed system allows users to add and modify objectives in a collaborative manner. Objectives are organized in categories in order to enable the users to select what types of objectives they want to be displayed on the map. The Objectives table stores the description for every objective as well as its position and its shape. The shape is used to display the objective as one or more colored polygons on the map.

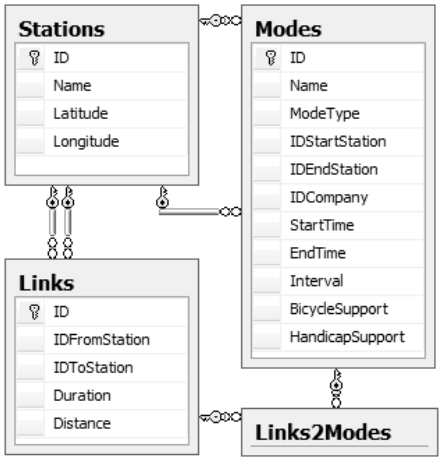

Fig. 2. Database – Route generation tables

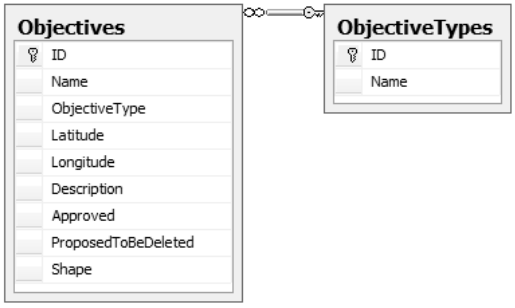

Fig. 3. Database - Objectives tables

Providing for a mechanism to handle transfers between stops located at very close distances are of high importance for designing such a system. Unlike other systems [14] choosing to statically store possible transfer stations, the proposed solution comes with a dynamic approach – identifying such stations by comparing the maximum walking distance selected by the user to the actual distances between available station pairs. Such distances are obtained through the Virtual Earth web service that offers facilities for calculating walking distances between two GPS coordinates. The web service is called directly from the database server using CLR (Common Language Runtime) which enables running .net code directly in SQL Server.

Besides data used in the route generation process, additional user supplied information will be stored such as the type of car owned by individual users. Data in these tables is extracted from publicly available databases containing data for various cars. Based on this, the system will be able to compute more accurately the savings in terms of fuel and money as determined by using mass transit as opposed to driving.

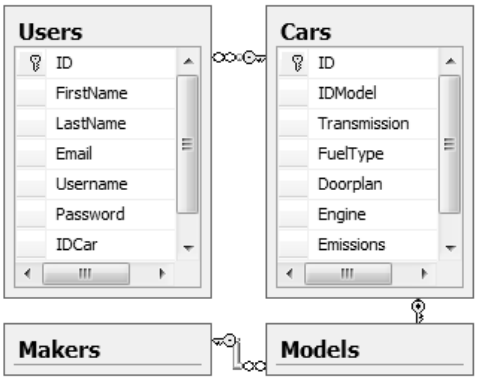

Fig. 4. Database - User information tables

5 Applying Genetic Algorithms for route generation

Besides the common fixed restrictions, the proposed system model also uses flexible restrictions as a key feature for route generation. As such, users can set individual weights to different criteria such as total commuting time, number of transfers, maximum walking distance to stops/transfers, and total cost of ridership in order to suit their interest and preferences better. Examples of fixed restrictions include generating routes using only selected types of transport means or only transport means that offer support for persons with disabilities.

In order to allow the use of both fixed and flexible restrictions, the route finding system is designed around genetic algorithms that allow unlimited restriction sets through the construction of the fitness function.

Genetic algorithms [14], [15] are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover. They are being applied successfully to find acceptable solutions to problems in business, engineering, and science [16]. The two most impor-

tant aspects of any genetic algorithm implementation are the fitness evaluation function and the reproduction scheme that must be mutually compatible [17]. Genetic algorithms try to eliminate the bad traits from the population using the process of selection. The good traits survive and are mixed by crossover to form potentially better individuals. Mutation also exists in GAs, but it is considered a secondary operator. Its function is to ensure that diversity is not lost in the population, so the GA can continue to explore.

In the case of public transport systems, the data required to generate the route includes the name of the origin and the name of the destination objective or station or the GPS positions if the user selected the start point and the end point for the journey directly on the map, the fixed restrictions and the selected importance for the flexible restrictions.

Using genetic algorithms the route determination is a multi-staged process involving:

- **Step 1 - Initialization** supposes generating an initial number of random routes, each route composed by a chain of connections between consecutive stations. Two stations can be considered consecutive if they are consecutive stations for a transport line or if the total walking distance for the generated route does not exceed the maximum value selected by the user. Two or more stations that are closer than the selected walking distance form a transfer area. In order to obtain feasible solutions and to reduce the total number of iterations needed, the generated chromosomes should not include a certain station more than once. The generated routes form the initial population which will be improved over several iterations. The individual routes are called chromosomes and the routes between consecutive stations are called genes. Therefore the optimality of a chromosome is characterized by the composing genes and their order.
- **Step 2 - Evaluation** includes calculating a fitness value to evaluate the optimum nature of each route (chromosome). The optimality of each chromosome is calculated using a fitness function. The function receives as parameters the evaluated route and the user selectable restrictions. It penalizes transfers, transport mean changes and other unwanted characteristics. In order to calculate the function's value, the minimum number of transport mode changes has to be determined first. The resulting value represents the overlap between the preferences of the user and the features presented by each calculated route – the more a route satisfies the restrictions put forward by the user, the higher the fitness value;
- **Step 3 - Selection** plays the role of a preparatory process needed for updating the current population. The routes having the best fitness values are selected for breeding a new generation of routes in a sequence similar to biological natural selection.
- **Step 5 - Mutation and Crossover** are the genetic operations that help create the new generation of routes starting with the routes selected in Step 3. First, in order to prevent the best chromosomes from being modified they are directly reproduced in the next generation. Second, new chromosomes are created using the two genetic operations.

Mutation creates new routes by replacing segments in the existing routes. It offers a way of adding random variation to the evolving population. Mutation arbitrarily alters the positions of one or more consecutive genes in the chromosome. The first gene and the number of genes to take part in the mutation process are randomly selected. In the public network transportation problem, modifying one or several consecutive genes can generate a new route having disconnected stations. Thus, we have to develop a specific mutation operation to deal with this problem. If a certain gene is selected as the starting point of the mutation, it can be considered the temporary origin of a route. A succession of chromosomes shall be generated using the same technique used in the Initialization step to replace the ones involved in the mutation without breaking the route.

Crossover combines the features of two parent chromosomes to form two similar children by swapping corresponding route segments of the parents. The genetic operation can only be applied if the chromosomes involved have at least two common stations, or two pairs of stations are in the same transfer areas.

The routes constitute themselves in a new generation of routes. Further, the algorithm starts over from Step 2, with this new generation constituting the initial data. The loop ends upon identification of a satisfactory solution or upon reaching a certain predefined number of iterations.

Taking into consideration the large number of potential system users, it is recommended to exploit the inherent parallel nature of the genetic algorithms to improve performance.

6 Interface

In order to maximize the user base of the application, as well as to increase user comfort and satis-

faction through flexibility, the system is implemented as to provide multiple access points with both desktop access for travelers wishing to generate routes before departure as well as mobile access for travelers wishing to modify routes while on the go, or to generate routes while not able to access a conventional computer or a web-enabled system. Implementing the different access points across a wide range of programming languages, platforms and operating systems is possible due to the use of service oriented architecture as described in the Architecture section. Additionally, such an approach would permit ample accessibility [18] by means of efficiency in using the capabilities of each device.

In order to avoid any installation and to more easily implement the collaborative features, the desktop implementation takes the form of a website (Fig. 6). This solution also provides portability across operating systems.

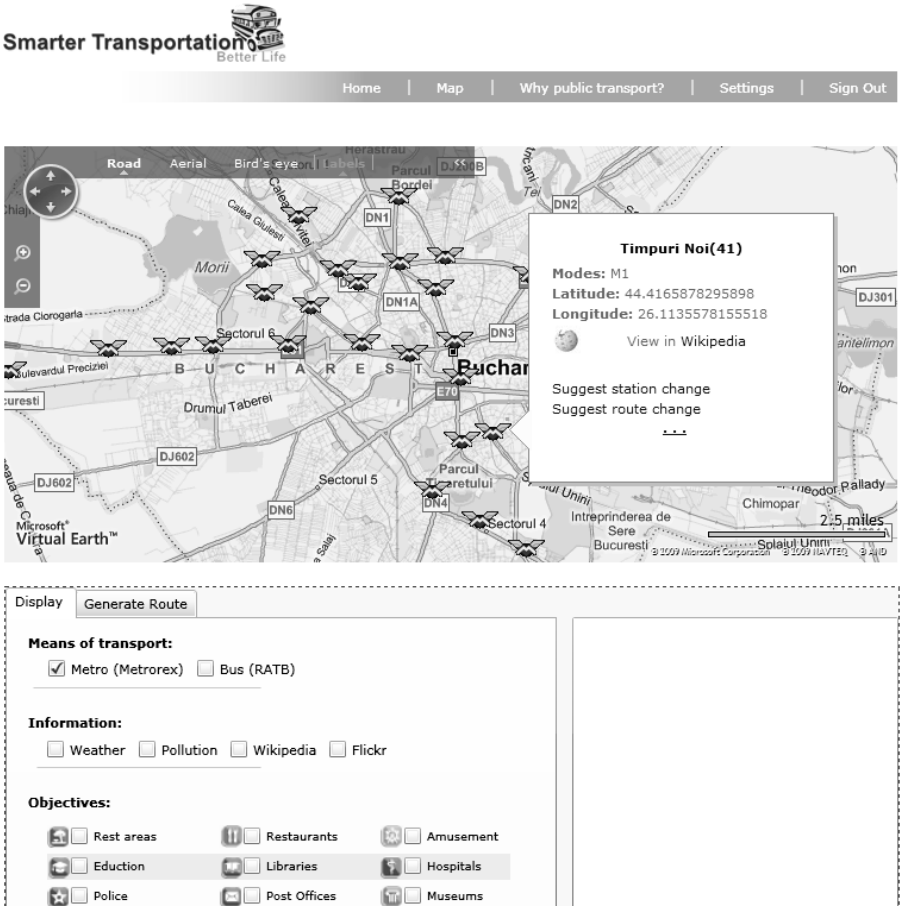

Fig. 5. Web Version

Fig. 6. Mobile Version

The implementation for mobile terminals (Fig. 6) allows users to generate routes whenever necessary, even if they do not have access to a computer. The identification of the client's location in order to be used in the route generating process or for the display of close objectives can be achieved either via the GPS module if this one is incorporated or, if not, via the mobile phone network.

No matter the chosen variant, the users can visualize the public transport routes, with the possibility of selecting only a certain type of public transport, can visualize the categorized objectives

as well as information taken from external sources via RSS or XML such as weather or the air pollution level.

The route generation (Fig. 7, Fig. 8) can be achieved by selecting the departure and arrival points directly from the map or by typing the name of a station or objective. The user can flexibly express his preference for one or several restrictions. The implementation of the flexible restrictions mechanism at the interface level was done using sliders which allow the user to give a smaller or a higher importance to a certain preference (Fig. 7).

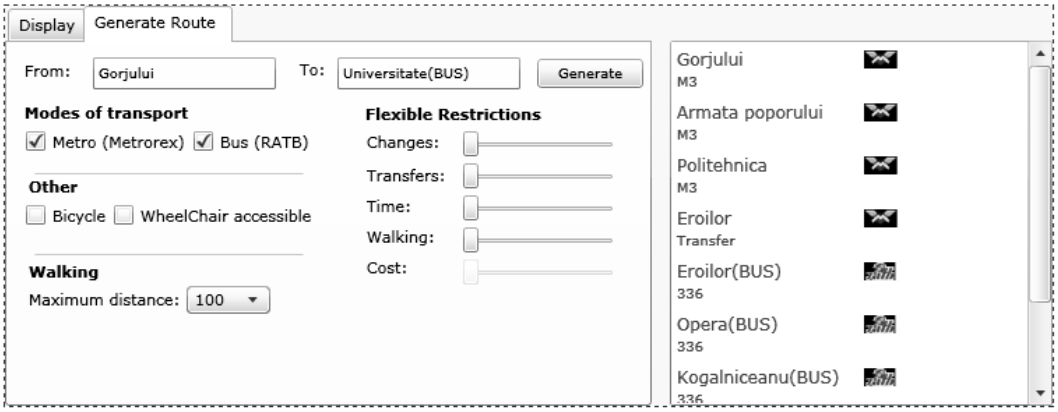

Fig. 7. Route generation (web version)

Alongside the flexible options, there are also available the classical limitations to restrict the generation of routes to a single type of means of transport or to impose the obligation of the existence of facilities for disabled persons on the generated route.

The generated route is displayed both on the map and under the form of a list of stations. Besides the station names, there are also displayed the means of transport that can be used, as well as the transport means type. If the journey implies station transfer, this is also displayed.

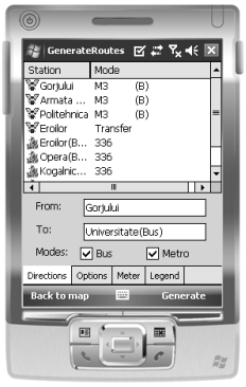

Fig. 8. Route generation (mobile version)

Users can add or modify information in the system in a collaborative manner. For example users can add new objectives or suggest changes to existing ones, they can vote for a certain objective, upload pictures, comment, draw a shape of the objective on the map. If multiple objectives overlap on the map, the decision of which objectives to display is taken based on the zoom level and the objective popularity. The mobile version of the application allows the users to add information and upload pictures even if they are not in front of a computer.

7 Concluding remarks

Implementing a collaborative system like the one described above offers advantages at multiple levels. For individuals it offers a faster and cheaper transit solution without the hassles connected to complicated maps and diagrams. For public managers, the system will offer a reduction in car traffic as well as a reduction in the costs associated with road maintenance, allowing easier implementation and more flexibility in terms of varying public policies in the field of transport. Environmental advantages include reducing air pollution as well as limiting global warming emissions combined with reductions in the consumption of fossil fuels. On the theoretical level, it breaks the ice in utilizing GIS collaborative solutions for providing further than last-mile services to users.

Acknowledgement

This article is a result of the project „Doctoral Program and PhD Students in the education research and innovation triangle". This project is co funded by European Social Fund through The Sectorial Operational Programme for Human Resources Development 2007-2013, coordinated by The Bucharest Academy of Economic Studies.

References

[1] T. O'Reilly, "What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software," *Communications and Strategies*, no. 65, pp. 22-23, 2007.
[2] J. Peng and K. H. Law, "A Prototype Software Framework for Internet-Enabled Collaborative Development of a Structural Analysis Program," *Engineering with Computers*, no 18, pp. 38-49, 2002.
[3] Ş. Nițchi, R.A. Nițchi and A. Mihăilă, "Some Remarks on Collaborative Systems Framework," *Informatica Economică*, vol. 11, no. 3 (43), pp. 10-13, 2007.
[4] J. Herbsleb and D. Moitra, "Global software development", *IEEE software*, vol. 18, pp. 16-20, 2001.
[5] E. S. Raymond, *The Cathedral and the Bazaar: musings on Linux and Open Source by an accidental revolutionary*. Sebastopol, California: O'Reilly Media, 1999.
[6] C. Peltz, "Web Services Orchestration and Choreography," *IEEE Computer*, vol. 36, pp. 46-47, 2003.
[7] I. Ivan and C. Ciurea, "Using Very Large Volume Data Sets for Collaborative Systems Study," *Informatica Economică,* vol. 13, no. 1 (49), pp. 29-37, 2009.
[8] M. Baldauf and P. Frolich, "Walking on Web - Combining User - driven Location Mapping and Mobile Visualization," OneSpace2008, *1st International Workshop on Physical and Digital Spaces on the Internet*, Sept. 2008.
[9] M. Goodchild, "Citizens as sensors: web 2.0 and the volunteering of geographic information," *GeoFocus (Editorial)*, no 7, pp. 8-10, 2007.
[10] Transport for London, The Town Hall of London, London [Online]. Available: http://journeyplanner.tfl.gov.uk. [Accessed: April 23, 2009].
[11] Google Transit, Google, Mountain View [Online]. Available: http://www.google.com/transit. [Accessed: April 23, 2009].
[12] Glossary for the OASIS Web Service Interactive Applications, Organization for the Advancement of Structured Information Standards [Online]. Available: http://www.oasis-open.org/committees/wsia/glossary/wsia-draft-glossary-03.htm. [Accessed: May 3, 2009].
[13] I. Lungu, D. Popescu and A. Velicanu, "Multi Channel Architecture Model Based on Service Oriented Integration," *Informatica Economică*, vol. 12, no. 3 (47), pp. 82-87, 2008.
[14] C. Jun, "Route Selection in Public Transport Network Using GA," in *Proc. Esri User Conference*, http://gis.esri.com/library/userconf/proc05/papers/pap1874.pdf.
[15] Chu-Hsing Lin, Jui-Ling Yu, Jung-Chun Liu, Chia-Jen Lee, "Genetic Algorithm for Shortest Driving Time in Intelligent Transportation Systems," in *Proc. of the 2008 International Conference on Multimedia and Ubiquitous Engineering*, pp. 402-406, 2008.
[16] D. Goldberg, "Genetic and evolutionary algorithms come of age," *Communications of the ACM*, vol. 37, pp. 113–119, 1994.
[17] A. Vișoiu, "Structure Refinement for Vulnerability Estimation Models using Genetic Algorithm Based Model Generators," *Informatica Economică,* vol. 13, no. 1 (49), pp. 64-71, 2009.
[18] A. Reveiu and T. Surcel, "Visualization of Accessible Multimedia Content in Web Pages," *Informatica Economică*, vol. 12, no. 2 (46), pp. 130-135, 2008.

Liviu COTFAS is a Ph.D. student and a graduate of the Faculty of Cybernetics, Statistics and Economic Informatics. He is currently conducting research in Economic Informatics at Bucharest Academy of Economic Studies and he is also an Academic Preparator within the Department of Economic Informatics. Amongst his fields of interest are geographic information systems, genetic algorithms and web technologies.

Cătălin CROICU is a consultant in web-based technologies and architectures currently taking a masters' degree in International Relations specializing in Game and Systems Theories at Uppsala University in Sweden. Amongst his fields of interest are web technologies and the software modeling of geographical, political and geopolitical phenomena.

Dumitru COTFAS is an Integration Consultant specialized in the ERP (Enterprise Resource Planning) integration with other applications like web applications, reporting tools, data capture and so on. He is currently working at Fraser Stream Software, Vancouver, Canada.