Implementation of Dynamic Routing as a Web Service for Emergency Routing Decision Planning

Choosumrong, S.,[1] Raghavan, V.,[1] Delucchi, L.,[2] Yoshida, D.[1] and Vinayaraj, P.[1]
[1]Osaka City University, Graduate School for Creative Cities, Osaka City University, 3-3-138 Sugimoto, Sumiyoshi-ku, Osaka 558-5858, Japan, E-mail:sittichai.ocu@gmail.com
[2]Fondazione Edmund Mach, Research and Innovation Centre, Department of Biodiversity and Molecular Ecology, GIS & Remote Sensing Unit, S., Michele all'Adige (TN), Italy

Abstract
The main objective of this study is to implement a system for Emergency Routing Decision Planning (ERDP) based on service oriented architecture. A Web-based application is developed to facilitate ubiquitous dynamic routing services on up-to-date road network data. The system pays special attention to enhancing the ERDP system in order to facilitate timely and smooth data transfer between client and the server. The ERDP is built based on an integration of the pgRouting Dijkstra's algorithm and Analytic Hierarchy Process (AHP). The system allows the clients to query the shortest path and travel time on the road network that is kept updated using data inputs from mobile client devices. Combined with AHP, this affords the calculation of minimum travel-time to the destination while considering many dynamic factors such as road condition and situations at destination. Static and dynamic road network data are maintained as separate tables in the ERDP system for anticipated improvements in processing speed. The routing algorithm is deployed as Web Processing Service (WPS) is using the ZOO kernel. Another highlight of the work is a jQuery-based mobile application which is used to update the ERDP database from field sites. This study demonstrates two scenarios for application of the ERDP Web services. In the first scenario of medical emergency, the ERDP calculates the route with minimum travel time and proximity of hospital to treat the patient. In the second case of routing service for rescue and evacuation, the ERDP provides dynamic route based on updates received from disaster forecast models. The system is developed entirely based on Open Source Software and its open architecture makes to easily amenable to customization and integration in a variety of application scenarios.

1. Introduction

Computing best and often the fastest achievable routes in road networks from a given source to a desired target location is a typical problem in everyday life. People frequently deal with this question when planning their trips. Further, routing services play an important role in emergency responses and decision-making. Traffic congestion is perhaps the most conspicuous problem in urban transportation network and has become a crucial issue that needs immediate attention. The transportation network is inherently dynamic because the state and measures of traffic change over the time. Thus, the correctness of the shortest route depends upon the efficacy of the cost model (Panahi and Delava, 2008). In previous work, Choosumrong et al., (2012a) have prototyped dynamic routing for Emergency Route Decision Planning (ERDP). This prototype is based on a combination of the pgRouting Dijkstra's algorithm with Analytical Hierarchy Process (AHP) for ERDP. The above mentioned study demonstrated the working of a non-standard dynamic routing system. In deploying a practical application of ERDP functionality, one is challenged to facilitate routing on road network that correctly reflect present conditions. This study addresses routing under rapidly changing road conditions arising from increased traffic, accidents or disasters. In order to achieve this objective, a method to implement dynamic routing using Web Processing Service (WPS) standard and updated road network data is proposed. The WPS introduces a standard method, which affords access geoprocessing services accessed from different client platforms. Such standardized geospatial Web services provide interoperability in decentralized distributed environments such as the Web. In the present research, ZOO kernel[1], pgRouting[2] and AHP is used to facilitate computation of optimal routes for emergency response scenarios. This new routing solution can be deployed on a variety of computer platforms, operating systems and Web applications.

[1]http://www.zoo-project.org/ [2]http://www.pgrouting.org

ISSN 1686-6576 / © Geoinformatics International

The routing services can be also used for multi-purpose applications such as routing of emergency vehicles and in planning routes during forecasted disaster such as floods. Further, a jQuery[3]-based mobile application has been developed to facilitate easily update road conditions as well as query the updated road network from mobile devices. Finally, the use of this dynamic routing service for medical emergency and disaster evacuation planning is demonstrated

2. Background

The previous work[4] by Choosumrong et al., (2012b) focused on integration of AHP and pgRouting algorithms to implementing ERDP. pgRouting was combined with AHP to calculate routes and assign priority weights for each impedance factor for the ERDP. In general, pgRouting computes a route from the minimum cost which can be considered as distance or travel time; as the cost can be influenced by a variety of factors, other parameters such as road condition need to be introduced in order to compute more realistic routes in emergency scenarios. The normal relational database schema used in pgRouting consists of three tables. The *geometry_columns* table stores information required to keep track of any table within a database that has the special geometry data type. The *spatial_ref_sys* table stores data that define thousands of different combinations of projections and datums required for accurate georeferencing of geometry data. Lastly, the road network data itself is stored in a separate table. Details of the pgrouting schema are available in the project documentation[5]. Once the database structure is established, the road network data can be imported using set of pgrouting utilities that allow import of ESRI Shape file or Open Street Map (OSM data). Subsequently, topology is added and the road network table is indexed. In order to assign temporary costs and allow for the dynamic routing, the road network was modified by adding new columns to store dynamic cost values.

3. Dynamic Shortest Path Algorithm for ERDP

This section describes the enhancements to our previous ERDP implementation mentioned in the previous section. These new improvements facilitate better data management and could also offer speed improvements in computing routes in complicated road networks. As mentioned earlier, in implementation of ERDP by Choosumrong et al., (2012b) both static and dynamic costs were stored in the same road network table. In order to optimize the system, static and dynamic data were stored in separate tables in the pgRouting database. Table 1 shows the modified schema where dynamic cost data is stored in two separate tables (Table 1). The new tables were joined to the static road_network table using *edge_id* (road segment ID) as the primary key. Managing dynamic data in separate tables enables faster indexing, join and vacuum functions (Momjian, 2001). ERDP implementation of Choosumrong et al., (2012b) was based on a non-standard approach and interoperability with Web applications was limited. To overcome some of the limitations of the non-standard approach and facilitate seamless integration with Web applications, the new ERDP algorithm was implemented based on a service-oriented architecture. Further, our earlier ERDP was less elegant and more cumbersome as it required several candidate routes to be calculated and subsequently prioritized based on distance, speed limit, and junction delay. In the new implementation, a more robust method has been devised to calculate the minimum travel-time by setting "*cost values*" to "*traverse values*" and calculating cost as a product of road length and average speed for the entire road segment *(avg_speed)*. The *avg_speed* is used instead of junction delay, since junction cost is the average time to traverse two edges plus the junction delay. The cost, thus, includes the junction delay by adjusting the average speed and using join avoidance between static and dynamic database tables to compute the dynamic routing result.

Table 1: Database structure for dynamic routing

Schema	Name	Type	Owner
ERDP_schema	geometry_columns	Table	Postgres
ERDP_schema	spatial_ref_sys	Table	Postgres
ERDP_schema	road_network	Table	Postgres
ERDP_schema	Dynamic_cost	Table	Postgres
ERDP_schema	AHP_weight_priority	Table	Postgres

[3]http://www.jquerymobile.com/
[4]https://www.jstage.jst.go.jp/article/geoinformatics/23/4/23_159/_pdf
[5]http://docs.pgrouting.org

4. Routing as Web Processing Service

The OGC®[6] Web Processing Service (WPS) protocol provides a standard way for clients to access dynamic computation models that operate on spatially referenced data on a remote server. The WPS is intended as a solution for developing Web-based geoprocessing to afford easy sharing of algorithms and analytical GIS functionality. In this study, ZOO Kernel is available as a robust WPS implementation that can be used to create, manage and chain WPS-compliant services by loading geospatial libraries dynamically and calling them on-demand (Fenoy et al., 2013). Further, Weiser and Zipf (2007) have also discussed about deployment of orchestrated service chain for Emergency Route Service (ERS) in disaster management. Implementation of dynamic routing as WPS and development of prototype system to provide near real-time ERDP for emergency scenarios has been carried out. At this time, Python has been used to implement the WPS services because it supports a variety of modules, classes, exceptions, very high-level dynamic data types, and dynamic typing. It also provides interfaces to many system calls and libraries. The prototype platform uses WPS extensively for data sharing as well as delivery of outputs. In order to run these WPS services, users only need to specify the location of the start and target Point of Interest on http request to the WPS server and calculate the minimum travel-time to the nearest suitable destination point. The new routing Web service was implemented by calling pgRouting library functions from a ZOO service. Figure 1 illustrates an overview of the ZOO WPS client/server workflow. In the Figure 1, step (1) to step (5) exhibit the processing of ERDP system between the normal user and Server. In step (1) and (2), the normal client accesses the system through the Web browser by different devices. In step (3) and step (4) Client sends dynamic routing query request to the server by Web map application (OpenLayers). On the server side, pgRouting with AHP analysis were integrated to create a dynamic shortest-time model. Not only has the algorithm to compute of a minimum travel-time been enhanced but also the dynamic situation at candidate destination points is kept up-to-date to provide optimal route result. After receiving the request from users, ZOO-Service connects to the PostgreSQL/PostGIS database and executes pgRouting function in step (5).

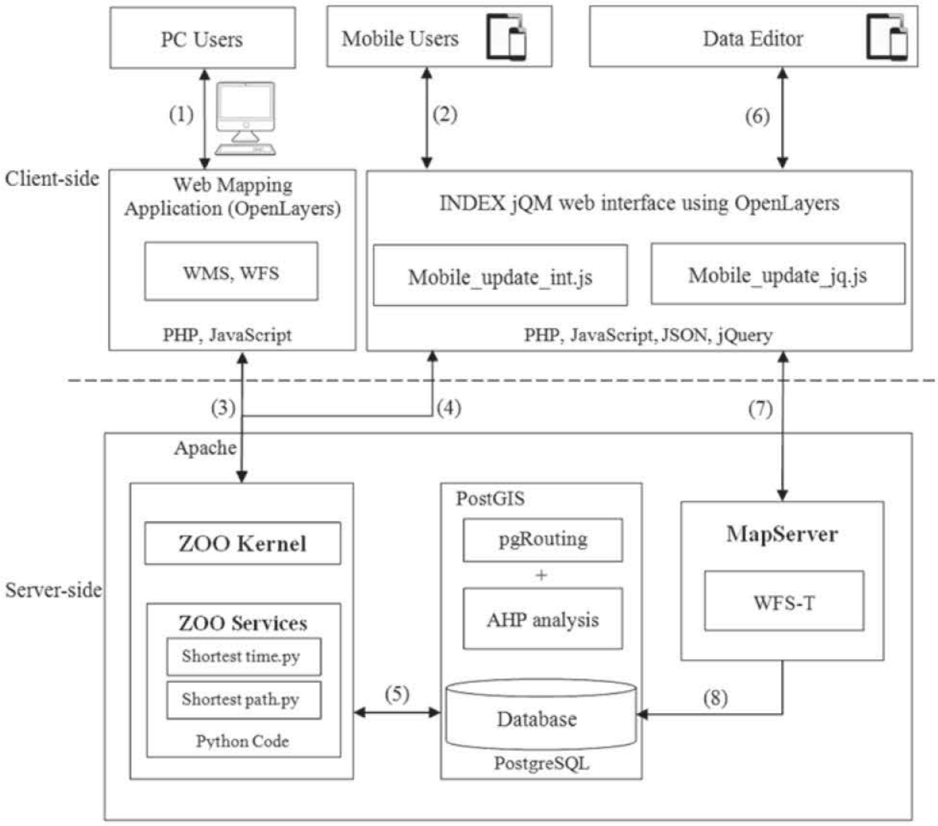


Figure 1: Architecture of routing as web services

[6]http://www.opengeospatial.org/

Then, the database is queried for the routing result which is sent back to the client. The client application also includes a WFS-T (Web Feature Service-Transactional) editing interface based on jQuery that which allows users with Data Editor to modify/update features and attribute in the road network database. In the Figure 1 step (6) to step (8) show the processing of ERDP system only between the Data Editor and the Server. At the step (6), the data editor accesses the system via mobile devices and sends the update on road condition to the server in step (7). In step (8), WFS-T modify/update the road network data in the database system after receiving the update from the data editor. The ERDP algorithms were implemented and deployed on prototype server platform using OpenSUSE 12.3 Linux distribution. The hardware specification included 3.20GHz Intel® Core™ i5 CPU and 8GB RAM. The server software tools used Apache 2.2.22, PostgreSQL 9.1, PostGIS 2.0, pgRouting 2.0, Python 2.7.2, MapServer 6.2 and ZOO-Kernel 1.3. The client side applications were built using OpenLayers 2.0, jQuery 1.3.2 and scripted using PHP 5.5 and JavaScript.

5. jQuery-Based Interface for Mobile Devices

Nowadays mobile devices are widely available and they are used in gathering field data in near real time. The jQuery Mobile (jQM) provides a rich set of jQuery plug-ins, widgets and a cross-platform API for creating mobile web applications. In terms of code implementation, it is very similar to the jQuery User Interface (jUI). However, while jUI is focused on desktop applications, jQM is built with mobile devices in mind. jQM is a freely available as an Open Source code base and provides a rich user experience on Web browsers running on mobile devices (Ableson, 2011). jQM uses HTML5, JavaScript and CSS3 features to enhance basic HTML markup in order to create a consistent mobile experience across supported platforms. jQM based applications work on mobile devices without JavaScript, even though a lot of redundant HTML is transferred over the network. For users who have a browser that supports JavaScript, the server only generates HTML on the first request and then subsequent requests use JSON and client-side templates to dynamically render the page. JSON is syntax for storing and exchanging text information. JSON documents are typically smaller than XML, and faster and easier to parse. In this study, a jQM Web application was implemented to interact with the road network database on a mobile device. The user-friendly application for mobile device is developed using HTML, PHP and JavaScript. Since the Web application is executed on the server side users need not have jQM installed on client devices and the application can on all Web browsers for mobile devices. The *index* file performs the database connection and calls the *Mobile_update_int.js* script which initializes map display by request to MapServer and JSON file retrieved is rendered on the Web browser using OpenLayers (Figure 2a). Finally, *Mobile_update_int.js* script used to file a report about conditions at the road segment that need to be temporarily avoided (Figure 2b, Figure 2c).

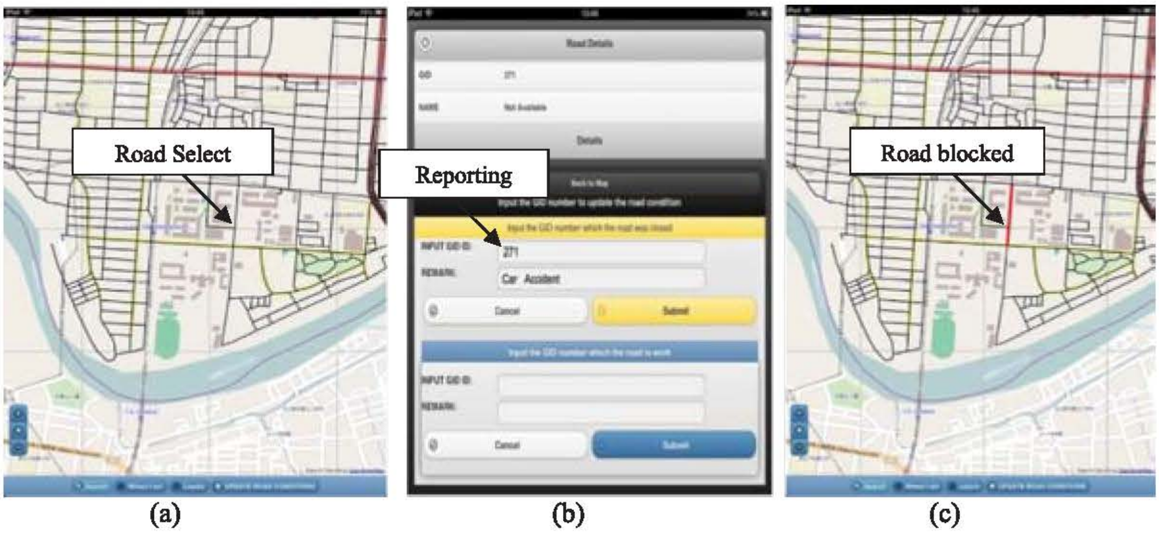
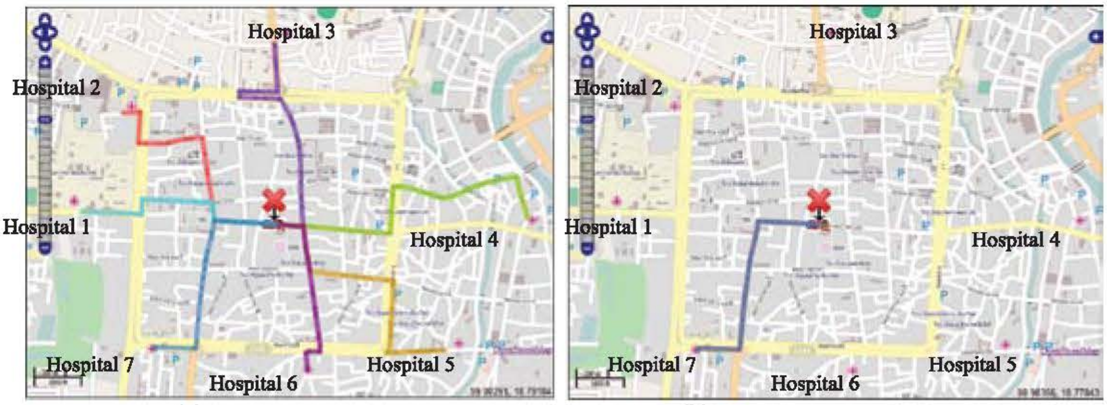


Figure 2: jQM web interface for update the road condition

6. Application Examples

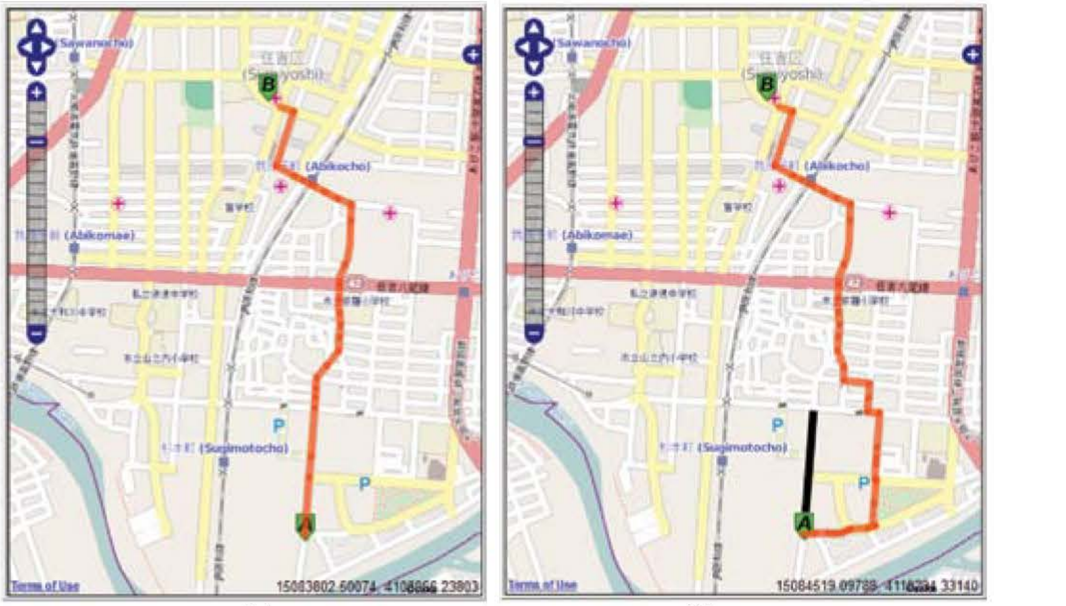
6.1 Routing Service for Medical Emergencies

Routing services for medical emergencies require that dynamic data not only for road conditions but also for the destination point and for the situation of the patient. In the case of road networks, the arcs in a directed graph correspond to street segments (edge E) whereas the nodes correspond to street segment intersections (node vertex V). Each arc has a weight associated with it, representing the impedance (cost) of traversing it. In this study, an arc's impedance is assumed to be the travel time for traversing that arc. The integration between AHP analysis and pgRouting Dijkstra's algorithm in order to create a new ERDP function (Choosumrong et al., 2012b), from a graph $G = (V, E)$, is calculated as the minimum travel-time after priority weight values are updated with a graph $G = (V, E_U \cup E_S \cup E_J)$ Users locate the road segment accident site (the X mark in the Figure 3(a)), all the hospitals located within a distance of 5 km is shown in Figure 3(a). In this example, the state of the patient is not "Code Red" condition ("Code Red" means that the patient needs to either be resuscitated or needs intensive care), and hospitals 1, 2, 3, 4, 5 have no available beds and the condition of the road from the accident point to hospital 6 is congested. With the ERDP function, all the parameters, such as patient state and available beds are recalculated and the routing result is returned to the user, as shown in Figure 3(b). However, other elements, such as road width are ignored in the calculation.



(a) (b)

Figure 3: Multi-routing calculation



(a) (b)

Figure 4: Routing result from start point A to end point B, (a) shows the routing result before the road is blocked. (b) shows the new routing result after updating data on the blocked road

Therefore, this routing scheme will not be adopted for ERDP as an ambulance might be delayed due to a narrow road width or a long junction delay. The system assumes that areas with a high traffic density would also influence ambulance speed. As discussed in the previous section, the ERDP system provides the client GIS with current road data via mobile devices. The route is queried from the road network and the database receives the real-time road conditions from the client application. This Web client interface allows the user to edit the cost value of each street segment, using simple touch events and mobile optimized HTML forms. Figure 2 shows the system interface rendered in a browser of mobile devices. The arrow shows in Figure 2a an example road segment where an accident occurred and the road is blocked. The road condition thus needs to be updated into the database, in order for the user to get the current real-time road conditions. This can be achieved by the user selecting the road segment which needs to be updated in order to get its road segment number. The cost value is then automatically updated after input and the road segment number is provided, as shown in Figure 2(b). The dotted line in Figure 2(c) then highlights the road segment that was just edited. Figure 4(a) shows the routing result from start point A to end point B. In this example, the state of the patient is not considered to be in a "Code Red" condition, and hospitals 1, 2, and 3 have no available beds, therefore, the minimum travel-time route from the start node (i.e., accident point) to hospital 4 is computed. Figure 4(b) presents the new routing result after the road condition has been updated.

6.2 Routing Service for Evacuation and Rescue in Forecasted Disaster

Natural disasters often result in a huge number of fatalities and extensive property damage and ERDP is necessary for quick rescue and relief operations. A road may be temporarily blocked because of a natural disaster such as a flood or a landslide (Hsueh et al., 2008). Dynamic information such as travel times may change after initial routes have been computed and the routing result must be recalculated based on the position of the destination and the travel-time. In other words, the routing algorithm must have the ability to re-estimate the optimal path at any given time and provide an acceptable solution. A flow chart for the dynamic routing system, involving a forecasted flood scenario, is shown in Figure 5. The processing sequence is; first, users select the start time, and input the start and end point.

Second, the server locates the nearest node and executes the shortest path and export as WMS. Third, the system determines whether the route result is intersecting with flooding area or not.

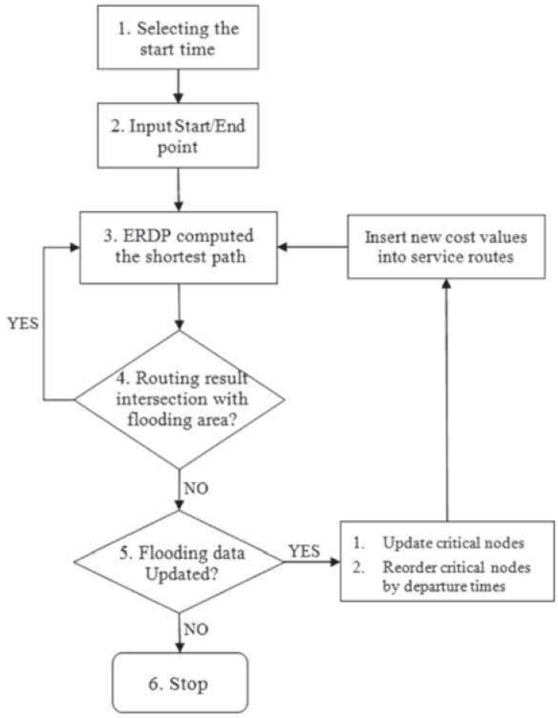
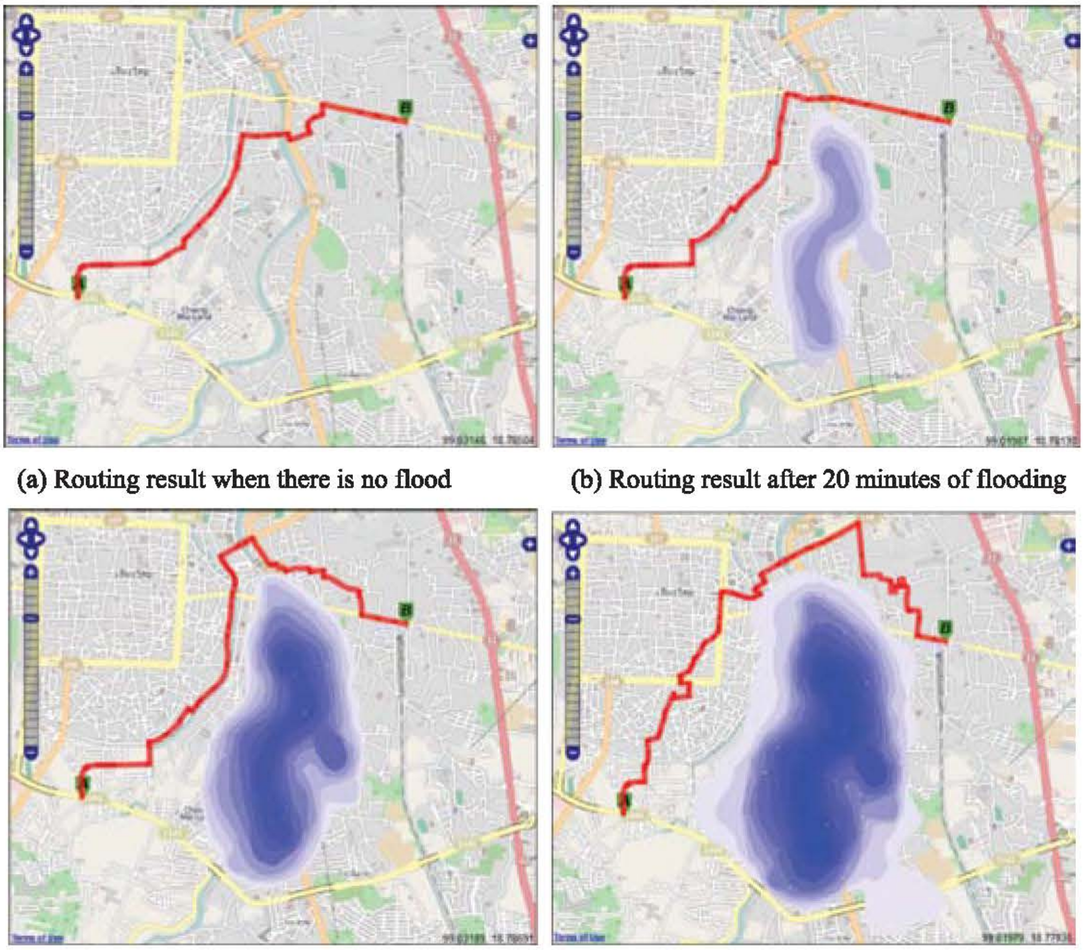


Figure 5: Framework of flood solution procedure

At the same time, the flood data are checked to determine update status. If the forecast map is not up-to-date, step 3 is recomputed using update forecast results. Figure 6(a, b, c, and d) depicts the newly computed route with respect to updated inundation areas caused by incessant rainfall over a period of time. The road network can be dynamically updated based of flood forecast model using rainfall, digital elevation

7. Discussion and Conclusion

In this study, the development and implementation of a multi-purpose dynamic routing system using a services-oriented architecture is described. Geospatial Web service technologies offer interoperability between geospatial and non-geospatial processing systems. Further, a new approach for dynamic routing is proposed by integrating pgRouting with AHP in order to assign a priority weight for alternative routes. The algorithm facilitates routing results to reflect present road network conditions and also take into account changing situation during transit and at alternative destinations.



(a) Routing result when there is no flood

(b) Routing result after 20 minutes of flooding

(c) Routing result after 40 minutes of flooding

(d) Routing result after 1 hour of flooding

Figure 6: The dynamic routing result updates in case of flooding

Implementation of mobile applications using jQuery Mobile and OpenLayers not only enables easy update the road condition in a participatory framework but also providing ubiquitous access to routing results. The pgRouting Dijkstra algorithm was enhanced to take into account dynamic changes in road conditions. It is combined with AHP to find the best route amongst different alternative routes and preferences. Such methods for calculating routes are useful especially when the factors under consideration do not have a common scale of measurement, or are intangible with no existing scale of measurement. This efficiency will be most important when an unexpected incident takes place in roads resulting in temporary changes in the condition of the road network. Further, the adoption of Open Standards facilitates an interoperable and scalable ERDP system. WPS is a new international standard, and effective utilization of WPS is still relatively uncommon. As demonstrated in this study, the WPS implementation using ZOO-Kernel allows utilization of results and easy integration into Web applications. With the advantages of Service Oriented Architecture, one is able to use pgRouting and implement the routing as a service for various kinds of needs and facilitate the route selection under diverse conditions at transit and destination. Since the Web service model is on its way to replacing the traditional two-tier client/server model, geospatial data and software services could be decentralized and offer a diversity of choices on data source and service provider. A model of ERDP was discussed based on the successful integration of Web services to provide routing functionality under dynamic conditions and improve emergency responses for disaster evacuation planning needs. The new version pgRouting 2.0 has functions like Turn Restriction and Shortest Path function (TRSP), where cost is a factor could be turn angles, road dividers, temporary changes in traffic flow directions which can be anticipated in a emergency situation. Further, use of elevation data over road segment to estimate the average speed of a heavy supply trucks in hilly terrain is also needed to enhance system functionality for emergency response. In addition to support for new routing functions available in pgRouting 2.0, our present effort in enhancing the system functionality is also focussed on integrating disaster forecast model with the ERDP system. GRASS GIS modelling tools can be easily integrated with ZOO-Kernel and deployed as WPS (Brovelli et al., 2012). Currently, work is in progress to integrate flood and fire hazard modules available GRASS GIS to provide real-time routing using time-series forecast maps. Lastly, all the software used in implementing the ERDP system is available under Open Source licenses.[7] It is planned to make the code and improvements made as a part of this study under Open Source licenses to promote wider use and collaborative development.

Acknowledgement

The paper forms a part of the first author's doctoral research at Osaka City University sponsored by Japanese government (Monbukagakusho: MEXT) Scholarship. The authors are also grateful to Dr. David Hastings, Dr. Go Yonezawa, Dr. Sarawut Ninsawat, Dr. Nicolas Bozon, Dr. Truong Xuan Quang, Mr.Gerald Fenoy, Mr. Daniel Kastl, Mr. Stephen Woodbridge and Mr. Jeff Mckenna for useful discussions in improving the quality of this work and support.

References

Ableson, F., 2011, *jQuery Mobile and JSON, Learn How to Create Mobile Web Applications Powered by jQuery Mobile.* Available from: http://www.ibm.com/developerworks/xml/tutorials/x-jquerymobilejsontut/x-jquerymobilejsontut-pdf.pdf

Brovelli, M. A., Truong, X. Q. and Fenoy, G., 2012, Managing Satellite Precipitation, Data (PERSIANN) Through Web GeoServices: A Case Study in North Vietnam, *Intelligent Systems for Crisis Management*, 159-170 (Editors, Sisi Zlatanova; Rob Peters; Arta Dilo; Hans Scholten) Springer Berlin Heidelberg (ISBN: 978-3-642-33217-3)

Momjian, B., 2001, *PostgreSQL, Introduction and Concepts*. 57-77. (USA: ADDISON-WESLEY)

Choosumrong, S., Raghavan, V. and Bozon, N., 2012a, Development of Web-GIS Application for Emergency Route Decision and Planning using AHP Analysis and pgRouting Algorithm, *Proceedings of GIS-IDEAS 2012*, 414-419.(ISBN: 978-604-73-1498-8)

Choosumrong, S., Raghavan, V. and Bozon, N., 2012b, Multi-Criteria Emergency Route Planning Based on Analytical Hierarchy Process and pgRouting, *Geoinformatics*, 23(4), 159-168.

Fenoy, G., Bozon, N. and Raghavan, V., 2013, ZOO-Project: the open WPS platform. *Applied Geomatics*, 5(1), 19-24. (ISSN:866-9298 (Print) 1866-928X (Online))

Hsueh, C. F., Chen, H. K. and Chou, H. W., 2008, Dynamic Vehicle Routing for Relief Logistics in Natural Disasters. *Vehicle Routing Problem*, Edited by Tonci Caric and Hrvoje Gold (ISBN: 978-953-7619-09-1) Available from: http://www.intechopen.com/books/vehicle_routing_problem/dynamic_vehicle_routing_for_relief_logistics_in_natural_disasters

Panahi, S. and Delavar, M. R., 2008, A GIS-Based Dynamic Shortest Path Determination in Emergency Vehicles. *World Applied Sciences Journal*, 3(1), 88-94.

Weiser, A. and Zipf, A., 2007, Web Service Orchestration of OGC® Web Services for Disaster Management. *Geomatics Solutions for Disaster Management*, edited by Li, J., Zlatanova, S. and Fabbri A.G (New York: Springer Berlin Heidelberg).