Student thesis series INES nr 263

Web-based public participation GIS application - a case study on flood emergency management

Peng Wang

2012

Department of

Physical Geography and Ecosystem Science

Lund University

Sölvegatan 12

S-223 62 Lund

Sweden

Peng Wang (2012). Web-based public participation GIS application - a case study on flood emergency management.
Master degree thesis, 30 credits in Physical Geography and Ecosystems Analysis
Department of Physical Geography and Ecosystem Science, Lund University

Web-based public participation GIS application - a case study on flood emergency management

---

Peng Wang

Master degree thesis in

Physical Geography and Ecosystems Analysis

Supervisors

Ali Mansourian and Per Schubert

Department of Physical Geography and Ecosystem Science

Lund University, Sweden, 2012

Table of Contents

List of Figures

Abstract

The increasing prevalence of natural disasters is driving people to pay more and more attention to emergency management. Progress in catastrophe analysis capabilities based on Geographical Information System (GIS) may allow the needs of public participation to be considered. Synchronous data sharing between citizens and emergency workers could effectively promote the process of decision making. This thesis introduces an interactive web-based application which mainly deals with flood risk management in Kamloops in Canada. The application is built for citizens and emergency workers using three layers: (1) the client side is developed in HTML and JavaScript; (2) the web server layer, which connects the users and the database, is implemented in PHP; and (3) the database contains PostgreSQL, GeoServer and OSM. Except the city map, PostgreSQL stores the spatial information with the support of OpenGIS. Generally, the application meets the initial objectives. Citizens can access present shelter information and register their own requirements for shelter, while emergency workers have the power to manage all the shelters and warehouses based on the available flood information and figure out the supply allocation solution based on the response from the public. On the other hand, the application also provides useful routing functions for both citizens and emergency workers, such as searching the available shortest path to a shelter, and computing the optimized allocation routes between all the shelters and warehouses. This practical study proved that Public Participation GIS (PPGIS), combined with IT knowledge, can provide very useful tools for decision making when facing a flood risk.

Keywords:
Emergency management; Flood emergency; Public Participation GIS; WebGIS, standardized web services; Location-Allocation

1. Introduction

Severe natural disasters usually happens when vulnerable areas meet environmental hazards such as floods, earthquakes or droughts (Kahn, 2005). Nowadays, the frequencies of these disasters are much higher than in past years. In the spring of 2012, extreme droughts covered most of US. Some southern parts of China were also affected heavily, such as the Yunnan province. While in the summer of the same year, many areas of China were attacked by horrible floods. Beijing suffered a terrible rainstorm which caused 79 deaths and destroyed at least 8200 homes. When facing such fierce catastrophes, the ability to manage disasters is very important. Emergency management plays a critical role in the whole process. However, in many cases, capabilities were insufficient. One important reason is the low attention paid to "information management" for disaster management. In reality, this information mainly consists of spatial data such as geometric, geographic or topological properties etc. These data provide us with powerful information for disaster response analysis, both on the overall and detailed levels. That's why GIS has thrived so quickly in this context and become a necessary tool for spatial data analysis and management.

Many studies have shown that analysis capabilities based on GIS technology has enhanced decision making. Gas dispersion simulation (Tumay et al., 2002), forest fire management (Akay et al., 2012), earthquake hazards assessment (Sun, 2011) etc. have all been improved. At the same time, it has been gradually understood that up-to-date spatial data for disaster management, especially data during the response phase, is quite important. Besides spatial data describing the disaster, information gathered from the citizens and other organizations are also valuable for emergency management and decision making. In some cases, they become two critical data sources. For example, public intents and emergency supplies' distribution are very important data representing the current status of an emergency. So, the progress of Public Participation Geographic Information System (PPGIS) can be meaningful.

Technically, the implementation and application of PPGIS needs strong support both for data updating and data analysis. The advantages of WebGIS and distributed GIS services can guarantee these needs will be met. And this technology can be used by both citizens and emergency managers. Furthermore, if I take my flood emergency application as an example, citizens can check the current status, update data, create an escape plan and route etc., while emergency managers have the abilities to update data, integrate data, publish information etc.

Based on the above thinking, I developed a flood emergency application which is designed for the river city Kamloops in Canada. I assume the following scenario: With a coming flood, emergency organization should help the public to take refuge where shelters are located. With this case study, I hope to carry out following aims:

1) Achieve information sharing between the public and emergency organization. For emergency workers, I set a deadline to collect the public's refuge-selection intentions through the internet.

2) At the time of the public's decision, the exact registration number of each shelter should be shown to the public synchronously.

3) The public should receive a quick and effective route to the demanded shelter.

4) Based on public feedback, an effective solution for facilities supply should be produced. It should provide transportation's routes and facilities allocation between the warehouses and the shelters.

This application could provide the emergency organizations in vulnerable areas with a more efficient tool when facing catastrophes. At the very least, it provides a good direction for future flood hazards management.

In this paper, I introduce the problem in chapter 1. In chapter 2, I discuss previous disaster management research based on a literature review. In chapter 3, I describe the information technologies for disaster management. In chapter 4, the design and implementation of my system are introduced. In chapter 5, I demonstrate how the system works. In chapter 6, I justify the importance and efficiency of the system, and discuss possible developments and limitations. Finally, the findings are summarized in the main conclusions.

2. Background

When describing emergency management, it's also very common to use the terms risk management, hazard management, disaster management and crisis management instead. Normally a natural disaster will result in various negative outcomes. It's the government's or decision makers' responsibility to carry out mitigation actions to limit the bad effects and reduce the damage. Therefore, emergency management can be considered as the identifying, managing and controlling of disasters (Zlatanova et al., 2009).

The procedures of disaster management can be divided into five phases (Lemmens, 2007) as below.

- Preliminary: analyzing the vulnerability of the region to certain types of catastrophes.
- Prevention: putting long-term measures into place in the region to reduce the vulnerability.
- Preparation: planning and arranging actual provisions to face the possible coming disaster.
- Response: carrying out emergency aid and support when the disaster happens.
- Recovery: reconstructing all the damaged areas to get back to normal life.

When we investigate flood disasters in more detail, much more detailed problems appear. Population density, local topography, historic hydrological records and many other factors have to be considered carefully during flood emergency management (Li et al., 2012).

In terms of actual flood emergency management, there have been GIS-based studies in recent years. A study with a participatory approach describes the implementation of collecting informal settlements data in Cape Town relating to flooding (Musungu, 2011). Relevant questionnaires were produced and surveyors were dispatched to meet local people. They then digitized the data and integrated them into the GIS for further analysis.

Another study introduced some on-line PPGIS tools that can collect information or reports of flooding from users and store them in a central database for further use (White et al., 2010). This effort combined Information and Communication Technologies (ICT) in some

ways. However, it's only a one-way design which means the application can only gather information from the public, but the citizens are not provided with guidance based on their own demands. So it probably cannot fulfill the public's needs under urgent situations.

As I described in the last chapter, the lack of information management has turned to be an important problem during the emergency management. Based on many emergency planning documents produced in different parts of the world, there has been very little done to make the emergency management plans that fit aid requirements and real demands (Alexander, 2002, 2005).

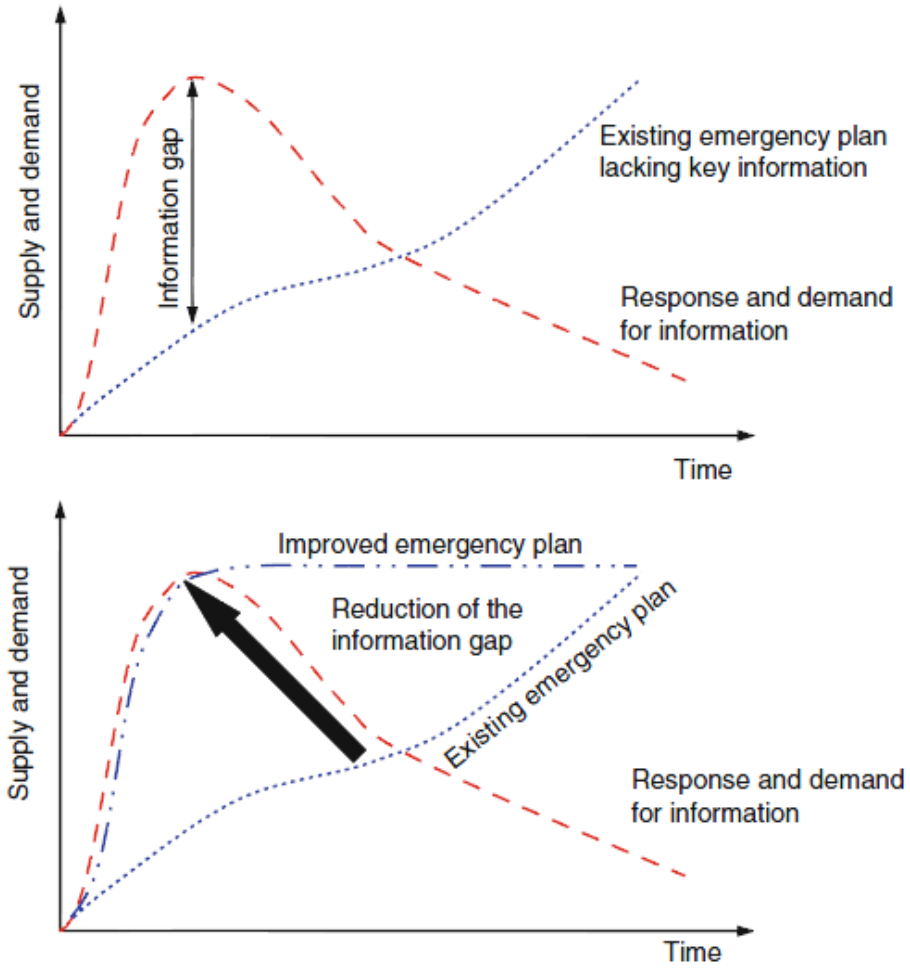


Figure 1. The ''information demand-provision gap'' during an emergency event. (MacFarlane, 2005)

Fig. 1 shows that the demand for information during an emergency usually exceeds the supply. This results in a so-called "demand-provision gap" (MacFarlane, 2005) or "information gap". In most cases, this is not because the information does not exist, but because it is not actually included in the emergency plan (Lumbroso et al., 2011).

Having considered the above problems, I have designed a more effective flood emergency management system which intentionally reduces the information gap and can provide a quick response to decision maker of the public. Citizens' responses could highly affect the analysis or solution in a significant short time period. By gathering dynamic feedback from the public, emergency organizations can publish a more effective and accurate solution to mitigate the effects of an coming disaster. Therefore, the interactive ability of the

developed system is very important.

3. Spatial information technologies for disaster management

This chapter introduces some important spatial information technologies available at present, including OGC service standards, spatial database and routing algorithm problems. These technologies are essential parts of this study.

3.1. OGC service standards

Beginning in the 1980s, people working in the GIS market started to realize the limitations of the variety of non-communicating tools that GIS creators had available to them. This provided the early motivation for the Open Geospatial Consortium (OGC). Nowadays, OGC is a non-profit international industry consortium that includes companies, government agencies and universities that promote the development of standards for geospatial- and location-based services. OGC standards provide interoperable solutions that allow the development of geographic information applications to introduce publicly open and extensible programming interfaces (Pezanowski et al., 2007). In other words, OGC standards greatly contribute to the interoperability between geographical information providers and consumers and promote the integration of data coming from a continuously growing number of sources (Tamayo et al., 2012). In this study, it is desirable to follow up OGC standards in order to use common geographical information forms on the internet. First, let’s introduce a few important OGC Web Services (OWS):

A Web Map Service (WMS) produces maps of spatially-referenced data dynamically that can be viewed through web browsers by users. Basically, the WMS-produced images are rendered in the pictorial format such as PNG, JPEG, or vector-based graphical elements (OpenGIS Web Map Server Implementation Specification, 2006). However, when users need particular sorts of geographic information, WMS is not enough. The Web Feature Service (WFS) standard helps us to produce maps of spatial data in an Extensible Markup Language (XML) format known as the Geographic Markup Language (GML). Feature-based GML data can be successfully distributed via geospatial web services like WFS to establish instantaneous data sharing environment (Hong et al., 2011).

In presently available software, almost all popular GIS tools support OGC standards very well. The well-known ESRI product ArcGIS fully supports the standards. There are a number of open source software systems that comply with these standards as well, such as MapServer, GeoServer, Quantum GIS etc. We can also find many web services following OGC standards like Google Maps, OpenStreetMap(OSM) etc.

3.2. Spatial database support

With the development of GIS, the need for spatial data storage becomes much larger than before. The normal flat file database, which usually contains a simple table, cannot meet this demand sufficiently. Spatial information and geographic objects call for a more reasonable and effective method to be stored and queried. Therefore, spatial databases were introduced. At present, there are up many spatial database systems: Oracle has developed its spatial capability very early; Microsoft SQL Server has had support for spatial types since version 2008; PostgreSQL is a powerful, open source object-relational database which I adopted together with the spatial extension PostGIS in this study. Due to the needs for processing geographic and geometric data, a database with sufficient spatial functions support is necessary or this project.

PostGIS is an open source software component that spatially enables the PostgreSQL database (PostGIS, 2012). In the database, geographic objects can be stored in the Geography Data type or Geometry Data type, including points, lines, polygons and multi-geometries. PostGIS also provides functionality for spatially enabled SQL queries. PostGIS fulfills the OGC's Simple Feature Specficiations for the SQL standard (OpenGIS Implementation Specification for Geographic information, 2010).

There are many types of geographic information software that can work with PostGIS. For example, MapServer can use PostGIS as a data source. GRASS supports PostGIS as a data source. OpenJUMP can connect to PostGIS and visualize the geometry data. Also, the Quantum GIS desktop has very good PostGIS support (PostGIS, 2012).

GeoJSON is another data interchange format that can encode a variety of geographic data structures (The GeoJSON Format Specification, 2008). A GeoJSON object could contain geometry or a feature. The feature supports various geometry types: Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, and GeometryCollection. In GeoJSON, an object is usually represented as title/value pairs (as in 'type': 'Point', 'coordinates': [100.0, 0.0]) (Hong et al., 2011). In this study, I used GeoJSON delivering the data from the server side to the client side.

3.3. Routing algorithm support

3.3.1. Shortest path problem

During development, it was necessary to solve the shortest path problem. There have been lots of open resources available to provide this capability, and pgRouting is the selected choice to help build this application.

PgRouting provides several algorithms to solve the problem with different demands, such as the Dijkstra algorithm, the A-star algorithm and the Shooting-Star algorithm (pgRouting, 2012). To compare the algorithms listed above is not my objective; I chose to adopt the Dijkstra algorithm to develop this application.

3.3.2. Location-Allocation problem

When it comes to the analysis portion of this work, the Location-Allocation (LA) problem is the most important topic. During the last twenty years, lots of research has been done to study more efficient problem-solving techniques using the concept of integrated logistics systems (Zhang et al., 2006). The LA problem is an important branch of routing optimization in an integrated logistics system.

Generally speaking, LA defines the optimal locations of a service in order to serve the public in the most efficient way (Blazevic, 2006). To solve the LA problem, the most popular method is to create linear programming formulations or other selected mathematic programming (Zhang et al., 2006). But in fact, the specific method is usually deals with the specific problem addressed. As I have not found a suitable algorithm to be applied for my demands, it is necessary to develop a method.

Considering the given conditions of this application, complex multivariate equations are not a good choice. So I changed my direction to focusing on logical methods. It is very difficult to transform real demands into an algorithm. For example, it is quite hard to decide from which shelter or warehouse to start the algorithm. So it is necessary to simplify and transform the problem. My friend Fei Yan, who works at Danone France as a manager of logistics department, provided a possible method. Instead of searching a shortest path to do the transportation, the main idea is always to avoid the potential highest waste of transportation costs. Combined with the database, I developed a general algorithm which follows this logical solution. I named it as Fei algorithm (Fig. 2).

Figure 2. The flow of the Fei algorithm.

3.3.3. Assumptions

As with all the other algorithms, the Fei algorithm has its assumptions or limitations:

(1) I assume that on all occasions, transportation to any given shelter needs only one vehicle, or an equal number of vehicles. In other words, on every occasion, transportation has same consumption, no matter how many facilities need to be moved.

(2) I assume that only the distance between the shelter and the warehouse represents the criterion of considering the priority of transportation.

(3) I cannot guarantee the solution to be the most optimal. But usually, it is the best solution according to my chosen criteria.

4. Design and implementation

4.1. Application structure

As shown in Fig. 3, the flood emergency application has three layers.

The first is the client side layer, which includes the interface for citizens and the interface for emergency workers. Each user interface is written in HTML, combined with JavaScript. Users can easily manipulate the application through common browsers.

The second layer is the web server layer, which could be installed on the servers of emergency organization centers. The major capabilities here include receiving and replying to the requests from the users, as well as connecting to the database for further operations etc. These functions are implemented in PHP.

The third layer is the data layer, which contains three data sources: A PostgreSQL database, GeoServer and OSM. Except for OSM, the other two parts are installed on a local computer. But I can assume the PostgreSQL database is installed on the server of the local municipality, and the GeoServer is installed on the server of a climate analysis organization or a similar place. The database store nearly all the spatial data needed, like the city street network, the shelters' data, the warehouses' data etc. The GeoServer mainly holds the flood prediction data.

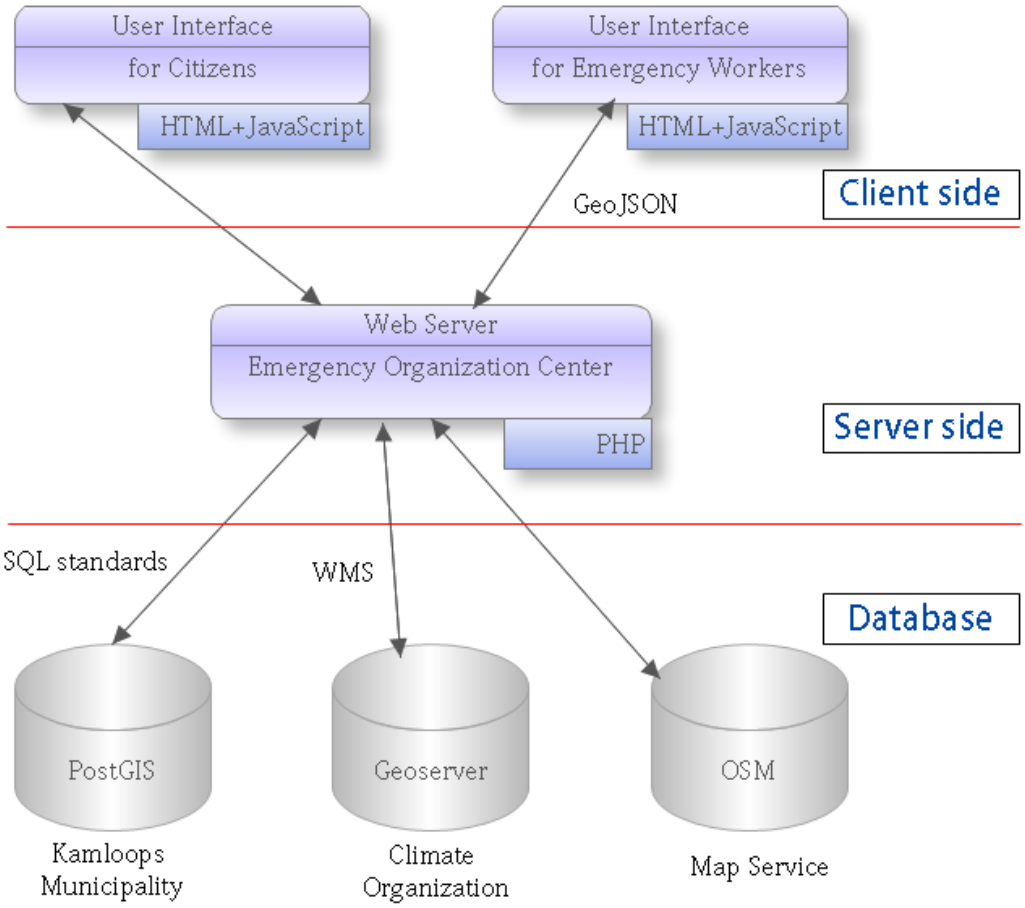


Figure 3. The complete structure of the flood emergency application.

4.2. Study area

The study site selected for this thesis is Kamloops (Fig. 4), a Canadian city situated in south central British Columbia, with about 86,000 residents in 2011. The main area of the city is located in a valley near the confluence of the north and south branches of the Thompson River. In some sense it is built on the river delta, or the flood plain of the river. For a long time, flooding has been an important problem in Kamloops, which mainly results from the annual snowmelt. When there is larger snow accumulation, a cooler spring which postpones the snow melts, and consecutive days of high temperatures in the beginning of summer, heavy flooding will definitely occur (GeoTour Guide for Kamloops, 2008). To protect the communities from flood damage, the local municipality has used dykes extensively.

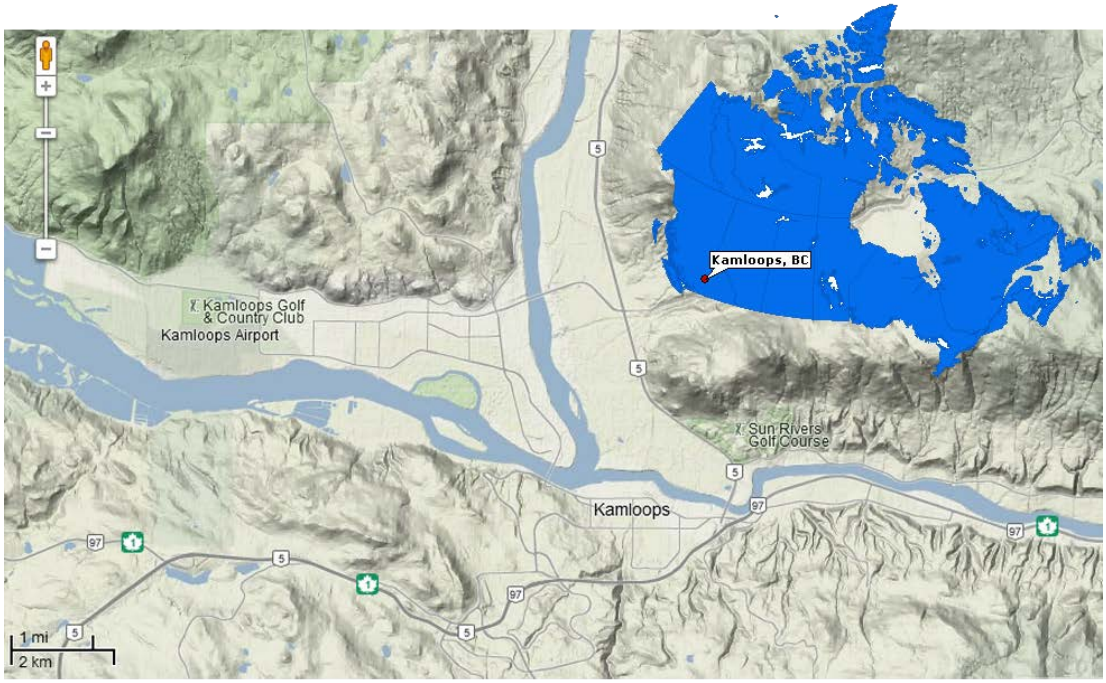


Figure 4. Kamloops, British Columbia, Canada. Edited based on Google Maps terrain.

4.3. Development environment

In this section, I will introduce the main components for developing the application. All the tools used are open source utilities.

4.3.1. MS4W

The MapServer for Windows (MS4W) package includes various tools for MapServer development on Windows. Two important parts were required: Apache HTTP Server and PHP. With Apache, the web server was implemented on my local computer; while as a middleware, PHP helps us connect the user side with the database easily.

In this case I used MS4W 3.04 with Apache 2.2.21 and PHP 5.3.10 included.

4.3.2. GeoServer

GeoServer is an interactive platform that allows users to publish, share, and edit geospatial data on the web. Not only does it fulfill OGC standards like WFS, WMS, and WCS etc. but GeoServer also supports a variety of data formats, such as KML, GML, JPEG, PNG, Shape files etc. (GerServer, 2012)

In this case I used GeoServer 2.1.3.

4.3.3. PostgreSQL and PostGIS

I have introduced PostgreSQL in the previous chapter. This database management system is widely used on various operating systems. It implements the majority of the SQL: 2008 standard, so it is very easy to use.

The spatial extension PostGIS provides support for geographic objects. This extension can import spatial data such as points, lines, polygons etc. into the database and edit them (PostGIS, 2012).

In this case I used PostgreSQL8.4 and PostGIS 1.5.3.

4.3.4. Other utilities

Besides those components listed above, the development process has involved some other necessary utilities such as ExtJS, GeoExt, Notepad++, OpenJUMP, Quantum GIS etc.:

- ExtJS: a pure JavaScript application framework for building interactive web applications using techniques such as Ajax, DHTML and DOM scripting.
- GeoExt: an extension for ExtJS that includes geospatial information in the user interface to build desktop style GIS applications with JavaScript.
- pgRouting: an extension of the PostGIS / PostgreSQL geospatial database to provide geospatial routing functionality.
- Notepad++: a convenient source code editor for Windows. In this case all the JavaScript and PHP codes were developed in Notepad++.
- OpenJUMP: a GIS software package with various functions. In my case, it played the role of a GIS data viewer, which means I imported spatial data from the PostGIS database into OpenJUMP to visualize them.
- Quantum GIS: a powerful desktop GIS application that provides data viewing, editing, and analysis capabilities (Quantum GIS, 2012).

4.4. Data sources

All the data needed are free and came from Internet. The road network data came from GeoBase (GeoBase, 2009), the flood plain Shapefile comes from Kamloops municipality (Kamloops Municipality, 2012), and the background city map comes from OSM (OpenStreetMap, 2012).

4.4.1. The city roads

The road network data for Kamloops was downloaded from GeoBase (GeoBase, 2009), which is a key component of the Canadian Geospatial Data Infrastructure. Basically, GeoBase

provides various sorts of territorial knowledge of Canada, including administrative boundaries, digital elevation data, land cover, national hydro network, national road network, satellite imagery etc.

I selected the Shape file data (Edition 8.0) for British Columbia to download.

There are several ways to deal with the original data; here I imported it into Quantum GIS to select just the roads to catch only in the area of City Kamloops (Fig. 5).

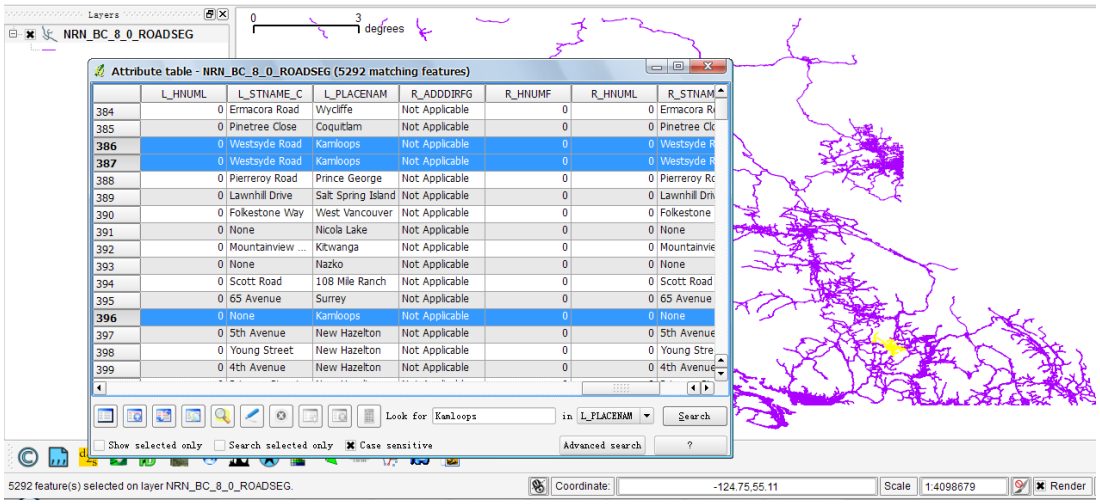


Figure 5. In Quantum GIS, The street network data for Kamloops was selected and saved as a new vector file.

The next step was to import the Kamloops roads data into PostgreSQL. I will describe this in the application development part, because the database should be routable before importing the road data.

4.4.2. Flood extent data

From the website of Kamloops municipality we can find different kinds of geospatial data in various formats. I used the 20 years flood extent and 200 years flood extent data sets that have been produced to help the emergency workers to analyze the situation.

The data provided are in the formats of SHP, GDB, DXF and KML. I selected the Shape file format to publish it through GeoServer.

4.4.3. OpenStreetMap

OSM is a collaborative project to create a free editable map of the world. Map data is available for download in a variety of formats such as XML, PNG, JPEG, SVG and so on (OpenStreetMap, 2012). I selected OSM to generate my background city map both for citizens and emergency workers.

4.5. Database preparation: pgRouting enabling

The preparation of database is very important to its correct functionality. I was guided through this process through a provided tutorial (A Beginner's Guide to pgRouting, 2011).

4.5.1. PgRouting enabling

The first step is to create a pgRouting-enabled database. One can either create a new database based on the PostGIS template and then load three .sql files (routing_core.sql, routing_core_wrappers.sql, routing_topology.sql) from my pgRouting download to support routing functions, or directly manipulate using the command line as below:

```
createdb -U postgres routing
psql –U postgres –d routing –f postgis.sql
psql –U postgres –d routing –f spatial_ref_sys.sql
psql –U postgres –d routing –f routing_core.sql
psql –U postgres –d routing –f routing_core_wrappers.sql
psql –U postgres –d routing –f routing_topology.sql
```

Here I named the database "routing". After its establishment, it is ready to import the road data.I can either use the PostGIS Shapefile Loader, or use the shp2pgsql utility to transform the Shapefile to SQL format under the command line as below:

```
shp2pgsql -s 4326 kam public.kstreet > kstreet.sql
```

Then it is easy to load the SQL file into the PostgreSQL.

If I look through the table kstreet, it is worth noting that the data type of column "the_geom" is geometry. The units for SRID 4326 are degrees, so the calculation of length will be in degrees. Obviously this result is not convenient for us, so here I need to convert the geometry type to a geography type, which can be calculated in meters by PostGIS. (Regina and Leo, 2011) I thus add a new column with the geography data type:

```
ALTER TABLE kstreet ADD COLUMN the_geom_geography geography;
UPDATE kstreet set the_geom_geography = geography(the_geom);
```

Actually, this problem can be solved by forming the query as below.

```
select ST_Length(the_geom_geography) from network;
```

It is easy to add a new column called length into the network table which contains the length of each segment in meters.

Next, following the “A Beginner’s Guide to pgRouting”, I created a node table for road segments and a routable topological network table for all the roads.
First I created start and end point geometries:

```
CREATE OR REPLACE VIEW road_ext AS
    SELECT *, startpoint(the_geom), endpoint(the_geom)
    FROM kstreet;
```

Then, I created a table containing all the unique network nodes (start and end points) and gave them each an ID:

```
CREATE TABLE node AS
  SELECT row_number() OVER (ORDER BY foo.p)::integer AS id,
         foo.p AS the_geom
  FROM (
    SELECT DISTINCT road_ext.startpoint AS p FROM road_ext
    UNION
    SELECT DISTINCT road_ext.endpoint AS p FROM road_ext
  ) foo
  GROUP BY foo.p;
```

Finally, I combined the road_ext view and the node table to create the routable network table:

```
CREATE TABLE network AS
  SELECT a.*, b.id as start_id, c.id as end_id
  FROM road_ext AS a
    JOIN node AS b ON a.startpoint = b.the_geom
   JOIN node AS c ON a.endpoint = c.the_geom;
```

Next, I inserted column shape_leng into the network table as below, representing the length of each segment in the network.

```
ALTERTABLE network ADD COLUMN shape_leng double precision;
UPDATE network SET shape_leng = ST_Length(the_geom_geography);
```

Now, the preparation of the data structure needed to provide routing capabilities is finished. I can visualize it through OpenJUMP to have a clear picture. In OpenJUMP I need to create a connection with my PostGIS database, and can then run queries.

It’s common to run two queries to get my result. The first will show the whole road network of Kamloops:

```
select * from kstreet;
```

And the second query can show my specific nearest route with a given start ID (e.g. 1000) and the end ID (e.g. 3110):

```
SELECT *
 FROM network
 JOIN
 (SELECT * FROM shortest_path('
   SELECT gid AS id,
    start_id::int4 AS source,
     end_id::int4 AS target,
     shape_leng::float8 AS cost
   FROM network',
   1000,
   3110,
   false,
   false)) AS route
 ON
network.gid = route.edge_id;
```

As a result, I get a specific output describing the prescribed route (Fig. 6).

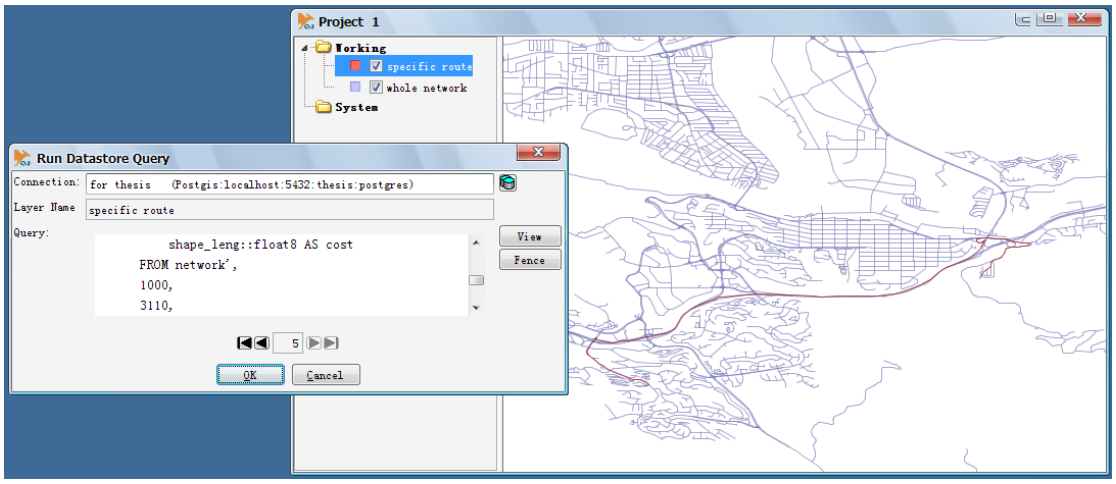


Figure 6. The output of a routing query in OpenJUMP.

4.5.2. Publishing the flood data on GeoServer

Future flood information or prediction is very important, because it directly influences emergency workers' decisions on the availability of the shelters. While flood data can be easily published by the meteorology organization through GeoServer, in this case I also play the role of the publisher of this data.

GeoServer has an easy-to-use interface. For the purposes of this study, I imported flood polygons of 20-year and 200-year floods scale, which were downloaded from the Kamloops municipality. Also, in conjunction with contour data, I created some maps of different

elevations, like 346m, 351m, 401m, 421m etc. As shown in Fig. 7, there are previews of 346m elevation and the 200-year scale flood.

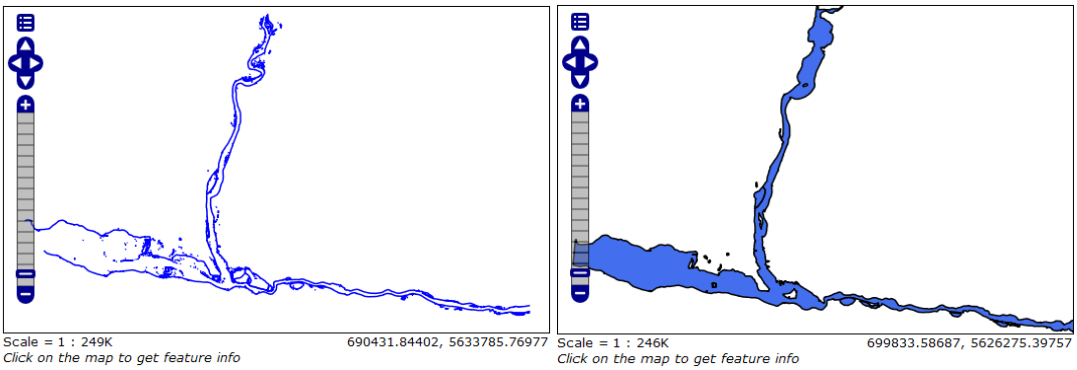


Figure 7. 200-year flood extent in the Thompson River at Kamloops, BC (right) and 346m elevation contour (left).

4.6. Development for citizens

When developing the application, material from the Free and Open Source Software for Geospatial (FOSS4G) 2011 Workshop (pgRouting Workshop Manual, 2011) was the main reference for building its structure.

For citizen users, I assume they have several needs regarding flooding and city route information. Basically, a person or a family probably wants to know the answers to the following questions:

- What's the shortest path to a specified location from my home?
- How long is the route?
- Are there any shelters I can go to if the flood is coming?
- How can I get to the nearest shelter?

Based on people's needs, I put some functions into the user interface for citizens as shown in Fig. 8.

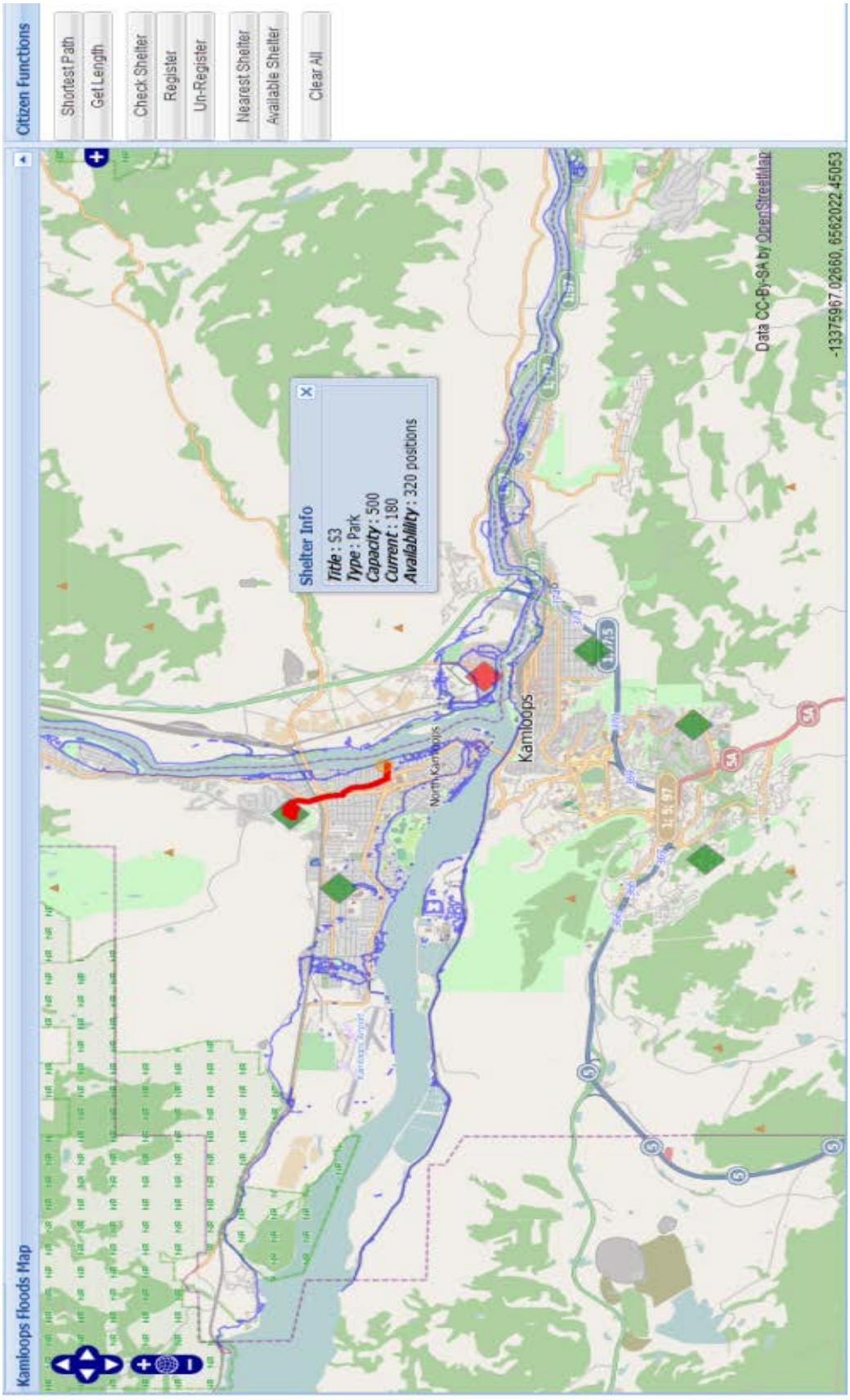


Figure 8. My application's user interface for citizens. In the figure, there is a route shown to the nearest available shelter from the given point.

4.6.1. Functions introductions

Here I list the functions as the button names that are shown on the web page:

- Shortest path: This is the basic map querying capability of the application. Users can select any two points on the map to get the shortest route between them. It is noticeable that the result is not the straight-line distance but strictly follows the road network. Due to the algorithm I used, the application always searches for the nearest road segment from the given point as the starting edge, so it is recommended to drop the point not too far from a road. Otherwise, it would not show you a satisfactory answer; for instance, the route from the city center to the top of a nearby mountain might be inaccurate.
- Get length: Based on the shortest path function, here the application can return the distance of the route in meters.
- Check shelter: Through this function, the application will show the information of the presently available shelters, including the shelter's name, type, capacity, current registration population and available positions. The currently unavailable shelters would be shown in red, which are classified by the emergency workers.
- Register: For dynamically sharing the population distribution, it is suggested that citizen users register at one suitable shelter for themselves. In order to register successfully, the number of registered people should be no more than the currently available positions; otherwise users must find another shelter which has remaining availability. Once the registration is done, the shelter's availability information will be altered accordingly.
- Un-Register: When registered citizens want to alter the shelter they wish to go to, the cancellation could be done here.
- Nearest Shelter: For users who want to find the nearest shelter from the given position, the route can be shown here. It is notable that the unavailable shelters are shown here, but it is easy to identify their status by their color of green or red.
- Available Shelter: It is more useful to check on the nearest available shelter. Given the number of participants, the result will show you the shortest feasible route to a shelter. The unavailable shelters and the shelters with fewer positions have been considered in the process.
- Clear All: Clear all the features on the screen.

4.6.2 Design

4.6.2.1. Client side: HTML+JavaScript

First of all, I need to create a HTML file to be read by browser, some JS files are imported, and CSS files need to be contained. Basically, the necessary codes looks like this:

```
<script type="text/javascript" src="Floods_User.js"></script>
```

In this part, all the important code is compiled through JavaScript. Here are the necessary

steps:

- Create the user interface (UI), which includes the map panel and the interface. ExtJS is introduced to help establish the general frame. Considering the need to deal with geographic data, GeoExt is adopted into my application.
- Build the map and other features with the OpenLayers standards, which are widely used in the application. For example, under the standards I create the basic map with the OSM data; I add controls to the map such as mouse movements; I reach the GeoServer's layer data by WMS service etc.
- Provide the users' controls. To make the application available, the basic controls must be implemented. Here, button events, drawing points, drawing segments etc. are all fulfilled by JavaScript code.
- Establish the connection with the PHP server. One of the most difficult parts is the data transfer between the client side and the web server. Data is obtained from PHP in the form of GeoJSON, which provides a collection of geometric features. In the application, the geometric data of routes are transferred through GeoJSON. Also, the information about shelters is provided by GeoJSON, such as the title and capacity of a shelter.
- Share some calculations. It is very important to put calculation tasks on the client side in order to maintain efficiency. For instance, when it comes to the positioning problem, it is solved through JavaScript instantly.
- Other functions; such as the widget shown when I check the shelters' information.

4.6.2.2. Web server: PHP

As a middleware, the web server has the responsibility to receive the user requests, to fetch the required data from the database, and send back the data to the client side. It is implemented in PHP.

Generally, more than half of the code in PHP deals with the database. Various queries are performed through SQL standards in the PHP environment. This allows the application receive the data from the client side, such as the coordinates, registration information and so on.

During the development, I created several PHP files to fulfill different requirements. The application points to different PHP files according to different functions. The following are some important objectives that need to be implemented through PHP codes:

- Find the nearest edge. When looking for the shortest path, in order to fulfill the input conditions, it's necessary to prepare the data of the starting edge and the ending edge. So first of all, the distance is queried between the given point and each edge in the database to get the nearest edge.
- Find the shortest path. The critical capability of the application is to find the shortest path between two points. Of course, query needs to be run based on different conditions. The result is a group of edges, which consists of the whole route.
- Read the shelter information. When the user wants to see the details of the shelters, it is necessary to query the information of shelters from the database. Basically, the title,

the type, the capacity, the current number of registration and current available positions are provided to the user.

- Registration. When the user has decided to register at one shelter with available persons, it is necessary to modify the data in the database. At the same time, the shelter's information on the small widget panel will be updated.
- Generate the GeoJSON data. After getting the geometric data from the database, it will be transformed into the form of GeoJSON, which is easy to transfer to the client side.

4.6.2.3. Database: PostGIS

In order to fulfill user needs, besides the basic network table and other supporting tables, a table including the shelters' information is necessary. Here a table is created named kshelter, which contains shelters' geometry attributes, ID number, title, capacity, type, current registration, availability and so on (Fig. 9). It can be altered by citizens and emergency workers.

Data Output | Explain | Messages | History

	the_geom geometry	id integer	title text	capacity integer	current_number integer	shelter_type text	availability boolean	needs_temp integer	source integer
1	010100002(	1	xixi	300	50	Park	f	50	2773
2	010100002(	4	xixixi	300	10	Park	t	0	2505
3	010100002(	5	S3	500	88	Park	t	0	562
4	010100002(	3	S1	400	282	Park	t	0	1078
5	010100002(	6	S2	400	384	Open area	t	0	3050
6	010100002(	2	haha	300	100	Park	t	0	2133

Figure 9. The kshelter table in PostgreSQL.

4.7. Development for emergency workers

For emergency workers, there are more powerful capabilities that need to be implemented as shown in Fig. 10. Considering the needs of citizens, listed are some objectives for my application, which are the necessary conditions to maintain the shelters' data, provide the service to the public, and help with allocation decisions:

- Able to show current flooding prediction on the map.
- Able to check the information of all shelters.
- Able to add and delete shelters, edit shelters' information.
- Able to mark the availability of shelters, in case shelters are affected by the flood.
- Able to add and delete warehouses, edit the warehouses' information.
- Able to figure out an efficient allocation plan, based on warehouse inventory and shelters' needs.
- Able to draw distribution routes of the allocation plan.
- Able to show the table of the allocation plan.

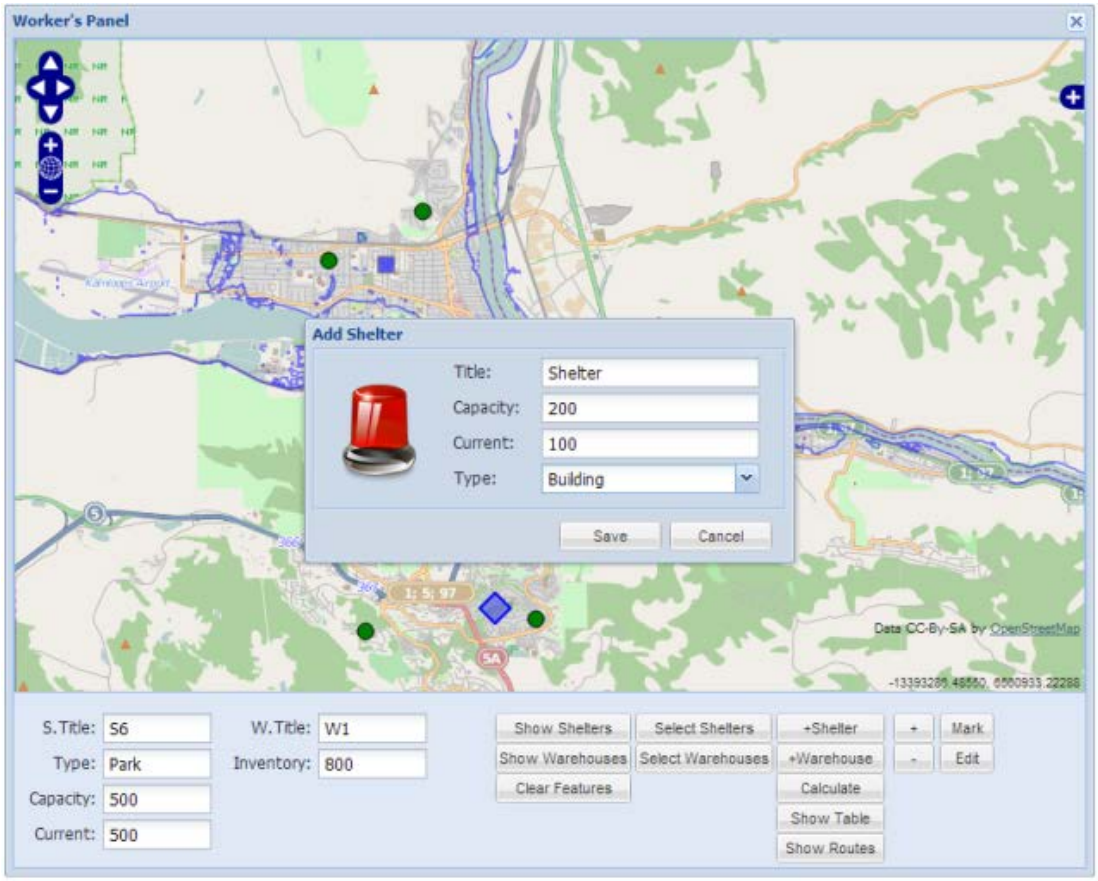


Figure 10. The user interface for emergency workers. The figure shows the function used to add a new shelter.

4.7.1. Manipulation introductions

The following are the functions provided to the emergency workers:

- Show Shelters: Present all the shelters on the map. Available ones are shown using green spheres and the unavailable ones using red spheres.
- Show Warehouses: Present all the warehouses as blue cubes on the map.
- Clear Features: Clear all the features on the screen.
- Select Shelters: By activating this function, the user can check the information describing any shelter with a mouse click. The related data will be shown in the panel.
- Select Warehouses: By activating this function, the user can check the information of any warehouse with a mouse click. The related data will be shown in the panel.
- +Shelter: By activating this function, the user can drop a point on the map to add it as a shelter. Combined with "+" button for use.
- +Warehouse: By activating this function, the user can drop a point on the map to add it as a warehouse. Combined with "+" button for use.
- +: After dropping a point on the map, this function helps to add the shelter or warehouse with a small window in which the worker can edit the target information.

- -: After selecting a shelter or a warehouse on the map, this function helps to delete the selected target.
- Mark: Once a shelter is selected, this function is able to mark the shelter as available (in green) or unavailable (in red).
- Edit: When selecting a shelter or a warehouse, this function is able to modify the information of the selected target.
- Calculate: With the current conditions (the given registration in shelters and the inventory in warehouses), calculate a solution of allocation routes and amounts to obtain the optimized plan.
- Show Table: Presents the solution in table, which lists the departure warehouses, destination shelters, and transportation amounts.
- Show Routes: Presents the transportation routes on the map.

4.7.2. Design

4.7.2.1. Client side: HTML+JavaScript

Similarly, I need to create a HTML file for emergency workers. Just like the interface for citizens, there are some necessary JavaScript and CSS files included in the HTML file. And, the important code is compiled through JavaScript as well. Here are the necessary steps that are required:

- Create the UI, which includes the map panel and the interface. Compared with the UI for citizens, I have a different layout for emergency workers. Likewise, ExtJS is introduced to help establish the general frame. And GeoExt is incorporated in this application.
- Build the map and other features with OpenLayers, which is almost the same task as the development for citizens.
- Provide the users' controls. For emergency workers, I need to implement many more controls: button events, drawing points, drawing segments, data input, data output, table control and other complex manipulations such as the modification of the data.
- Manage the connection with the PHP server. In the development for emergency workers, the data transferred between the database and client side increases dramatically.
- Introduce interactive windows. When emergency workers add, delete, and modify data, it is necessary to bring in some interactive functions. Pop-up windows are very suitable for these needs, so a window is created for shelters and a window for warehouses is created separately. They can be called repeatedly during the manipulation.
- Present the solution as a table. With the help of ExtJS, it's possible to output the supply solution in a table. Receiving the GeoJSON data from the PHP level, the needed data can easily be shown with grid panel.

4.7.2.2. Web server: PHP

The importance of the portions of this application implemented in PHP has been mentioned. Actually, when developing the part for emergency workers, there were many more functions to be created. Based on the design for citizen users, many database controls need to be built, such as bringing in the warehouses, the data manipulation on both shelters and warehouses, the allocation solution analysis and so on:

- Manipulation of shelters: As an emergency worker, a basic responsibility is to deal with the shelters. At this position, the worker should have the opportunity to integrate various kinds of information, such as flooding predictions, population distribution, city topography, that can help to decide the shelters' establishment, availability, registration status etc. So, any necessary functions that control the shelters should be provided to the emergency workers. Through the application, workers can add and delete the shelters, and modify the shelters' information. Additionally, availability can be decided, which represents an important sign or warning to the public. The manipulation on the UI is implemented through JavaScript, but the real modification behind the scenes is implemented in PHP.
- Manipulation of warehouses: The introduction of warehouses is quite important, because in addition to the management of shelters, the emergency workers have a duty to decide the distribution of supporting resources. Likewise, similar controls on warehouses need to be added. For the information on warehouses, there are two special attributes representing the current inventory and the temporary inventory.
- Algorithm implementation: I have talked about the Fei algorithm; it was necessary to transform the idea into code. During development, I used several methods to create the whole algorithm. Frequently used operations are connecting with the database, including read data, modify data, create table, and delete table. A double for-loop was adopted to go through all the possibilities for one shelter. Basically, operating efficiency was not considered primarily, but considering the nature of the application, it is acceptable.
- Generate the solution routes: After the algorithm's operation, the solution will be stored in the allocation table. Initially, through PHP will return a group of routes in the form of GeoJSON. In other words, it contains all the resulting supply routes. Following the worker's request, the routes can be shown on the map of UI.
- Return the solution data: In addition to the routes, detailed data might be requested. There is another PHP file provided to read the allocation's data and return it back to emergency workers. These will be presented in a table as mentioned above. The data transferred doesn't include geometry data.

4.7.2.3. Database: PostGIS

When it comes to the needs of emergency workers, the demands of database operations increase significantly. First of all, there must be a new table representing the warehouses. It contains basic information like geometry data, warehouse title, current inventory, temporary

inventory, target number and ID number.

A table was introduced called find_distance, which mainly stores the distance between the shelter and the nearest warehouse, as well the 2nd nearest one. The table contains the source number of the shelter, the target number of the warehouse, the distance, the shelter's title and the warehouse's title.

Another important table stores the data of the final solution as shown in Fig. 11. It is the allocation table which stores the shelter source number, the warehouse target number, the supply amount, the warehouse title and the shelter title.

Data Output | Explain | Messages | History

	warehouse_target integer	shelter_source integer	supply integer	destination text	start_point text	id integer
1	1205	1078	200	S1	W2	1
2	1746	2505	10	xixixi	W2	2
3	2012	562	88	S3	W4	3
4	2012	2133	100	haha	W4	4
5	2012	1078	82	S1	W4	5

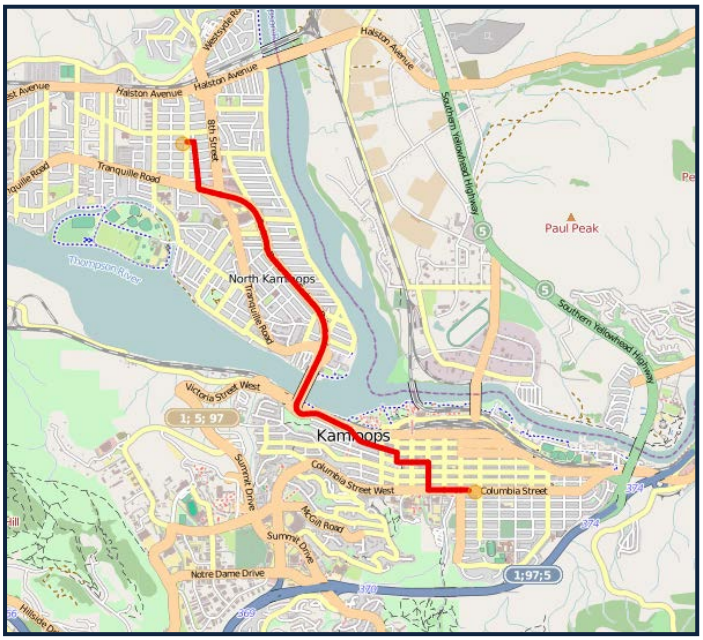
Figure 11. The allocation table in PostgreSQL. It contains the shelter source number, the warehouse target number, the supply amount, the warehouse title and the shelter title.

5. Application presentation

In this chapter, I will demonstrate how the application works, step by step. An introduction will be given for the UI provided for citizens and the UI provided for emergency workers.

5.1. User interface for citizens

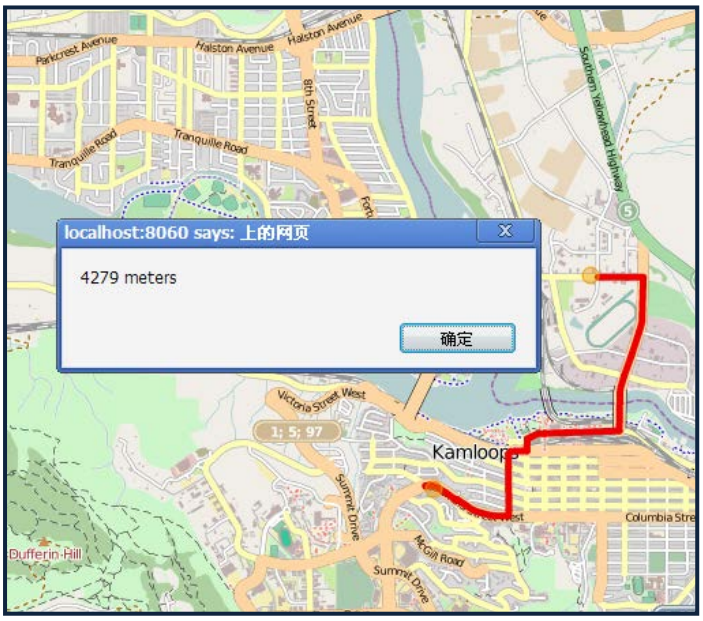
a. Shortest path



The basic function for citizens is to search for the shortest path between two locations, given by mouse clicks. As long as the given points are in the range of the street network data, users can obtain the shortest path result as shown in Fig. 12(a).

Figure 12(a). A shortest path result.

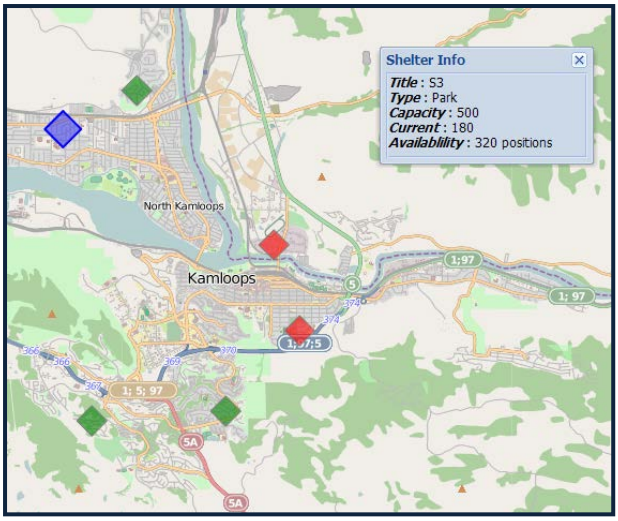
b. Route length



If a user needs the exact distance between two locations, the application can provide the distance reply by dropping two points on the map and performing the calculation as shown in Fig. 12(b).

Figure 12(b). Get the length of a route.

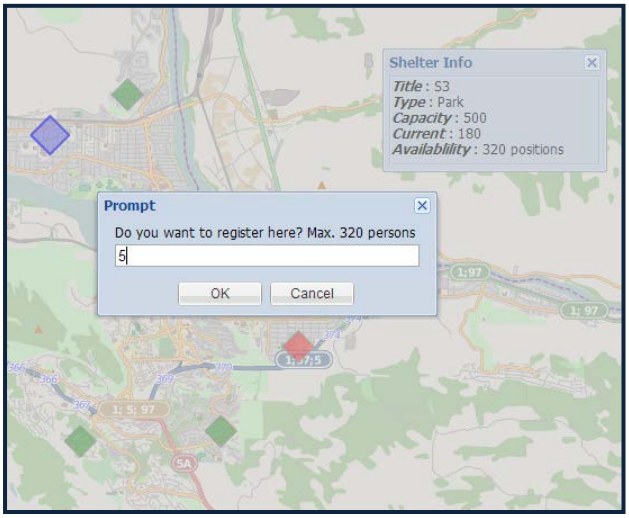
c. Check shelter



The main task of the application is to provide shelter-related information. Users can check the availability of shelters on the map. As shown in Fig. 12(c), the available shelters are marked as green, the unavailable shelters are marked as red, and the selected shelter is marked as blue. Shelter attribute information is shown alongside, such as title, type, capacity, current population and available positions.

Figure 12(c). Check shelter-related information.

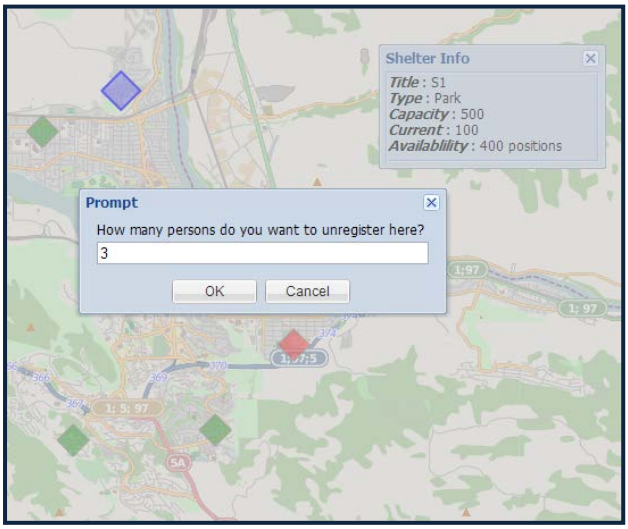
d. Register



When a user has decided which shelter to go to, he or she can register at the selected shelter with the total number of persons needing shelter. If there are not enough positions available, it is suggested to find another shelter.

Figure 12(d). Register at a selected shelter.

e. Unregister



If a user wants to cancel the registration/decrease the number of persons registered at shelter, they can unregister from the selected shelter with a numeric input.

Figure 12(e). Unregister a number of persons from a shelter.

f. Recommend a shelter

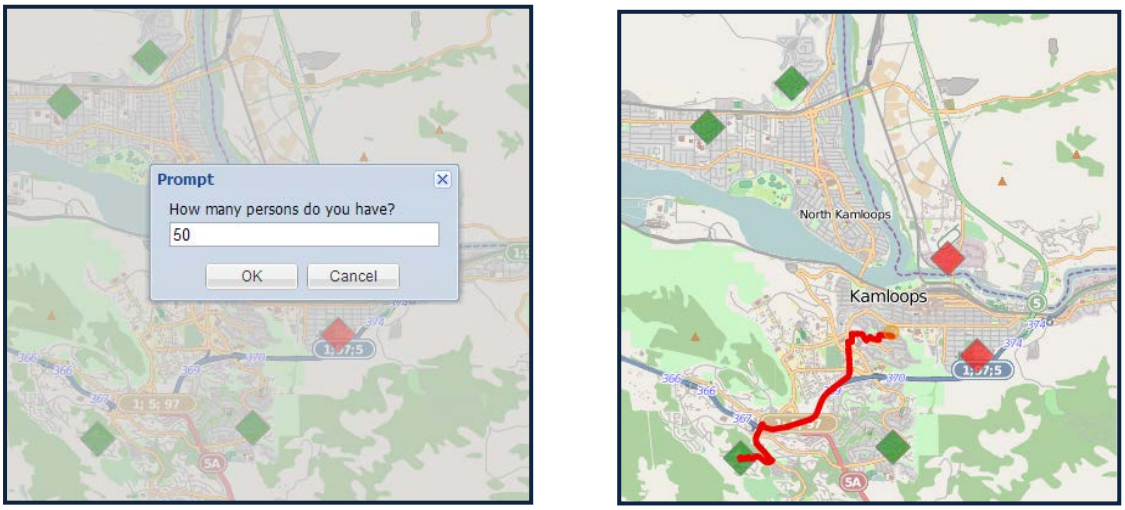


Figure 12(f). Recommend a shelter.

This function suggests a particular shelter to a user. Given the number of persons coming, the application can recommend a nearest available shelter and provide the route to it. As shown in Fig. 12(f), the red shelters are not in use, and the closer green shelter does not have enough positions available. So the user can consider the suggestion and make a choice.

5.2. User interface for emergency workers

a. Preparation

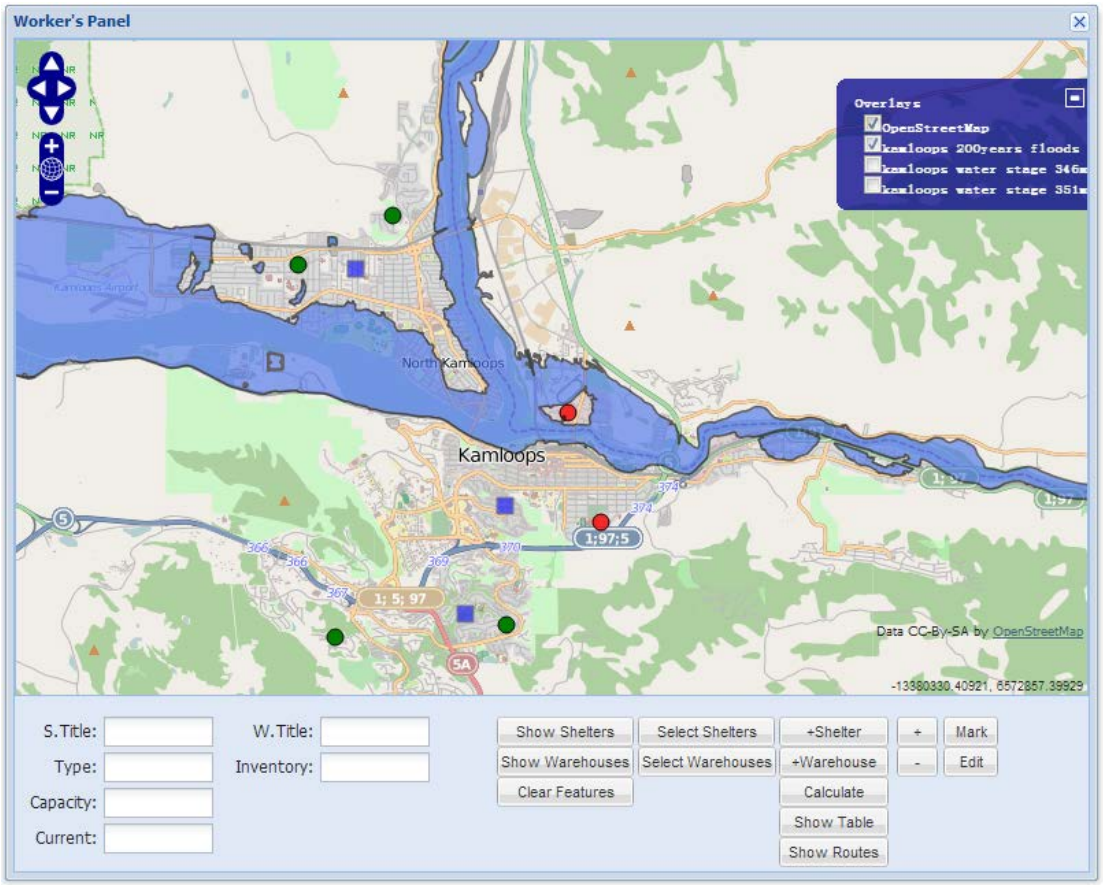
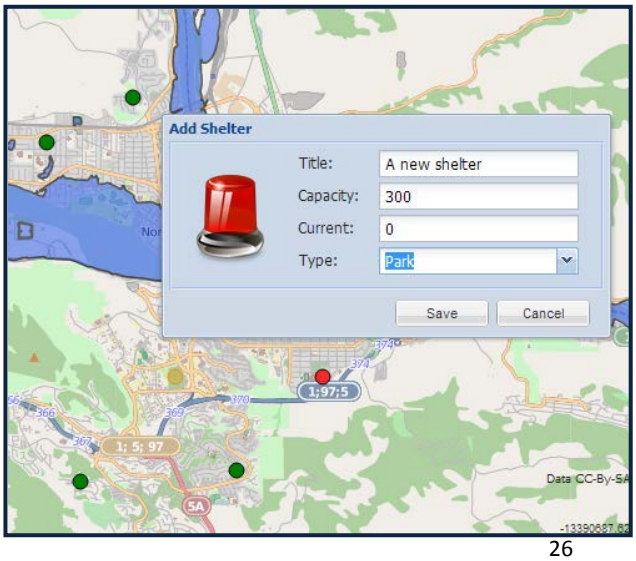


Figure 13(a). Showing flood prediction information and the shelters/warehouses distribution.

First, emergency workers can view coming flood prediction or other flood related data sets by selecting layers. Also, the shelter and warehouses distribution can be shown on the map. As shown in Fig. 13(a), shelters are marked using small circles: red means unavailable and green means available. The blue cubes denote the warehouses. We must store the flood data sets provided by the local climate organization, and the shelters/warehouses data stored in the municipal database.

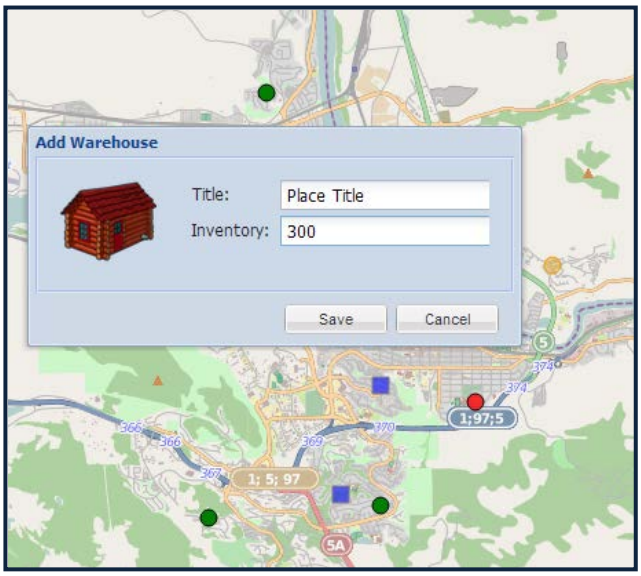
b. Manipulation of shelter data



Based on the flooding situation, emergency workers need to deal with shelters using operations, such as add a shelter, delete a shelter, edit the shelter information, or mark the selected shelter as available or unavailable. Fig. 13(b) shows adding a shelter, in which the yellow circle represents the new point.

Figure 13(b). Add a new shelter to the database.

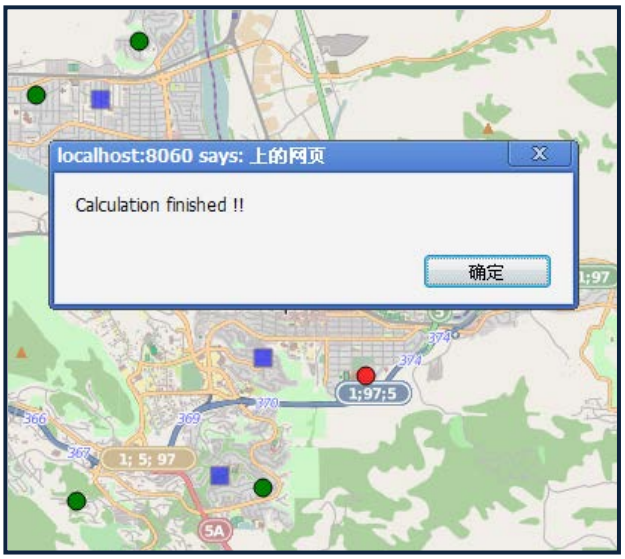
c. Manipulation of warehouse data



Similarly, based on the flooding situation, emergency workers also need a set of operation to deal with warehouses, such as add a warehouse, delete a warehouse, or edit the warehouse information. Fig. 13(c) shows adding a warehouse, in which the yellow circle represents the new place.

Figure 13(c). Add a new warehouse to the database.

d. Solution calculation



When all shelter data and all warehouse data are prepared and ready, emergency workers can start a calculation to get the optimized solution of facilities transportation.
Usually, it takes a few seconds to get a response.

Figure 13(d). A successful calculation response.

e. Solution presentation

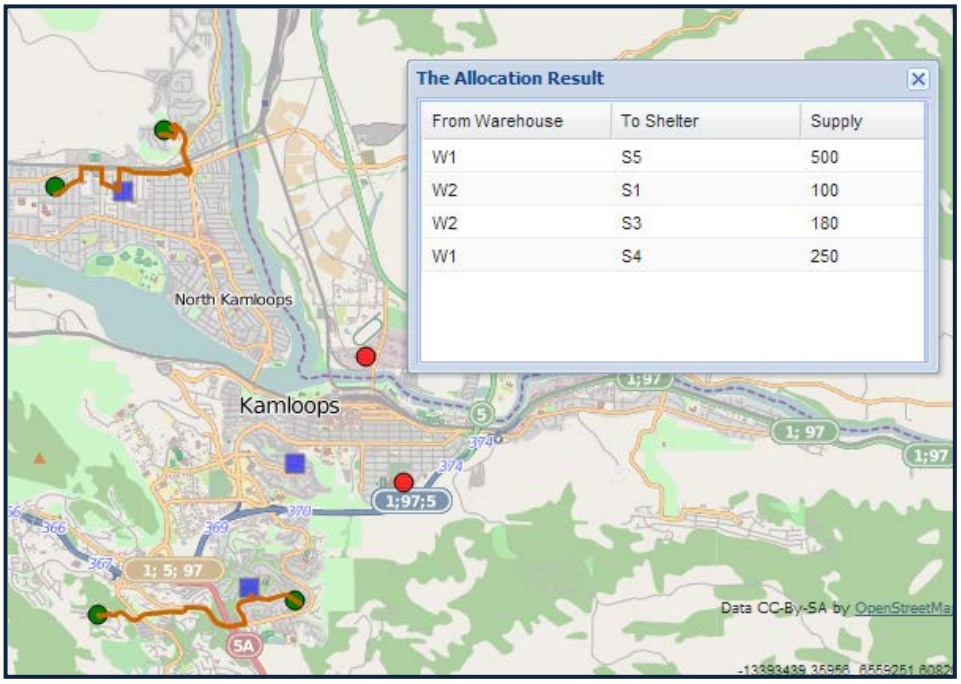


Figure 13(e). The recommended routes and the allocation table for facilities transportation.

After the response, it is calculated to present the result in two ways: By drawing routes on the map, and showing the possible allocation table. As shown in Fig. 13(e), the allocation table demonstrates the four transportations that are a part of this solution. Each shows the departure warehouse, the destination shelter and the supply amount. On the map, we can see four routes which match the table content.

6. Results and discussion

The application as implemented has met the design goals specified.
Through the application, I can provide the citizens with several useful functions:

- Checking the real-time status of shelters;
- Searching the available routes to a specific/nearest shelter;
- Registering at the wanted shelter etc.

The application also provides the emergency workers with powerful analysis abilities:

- Given a flood prediction data set, emergency workers can define the availability of shelters;
- Based on information gathered from citizens, emergency workers can get the solution of facilities supply etc.

This application has the ability to solve allocation problems under various conditions:

a. Large shelter

As shown in Fig. 14, shelter S5 has a high demand. We can see from the allocation result table that the nearest warehouse, W1, cannot fulfill the demand, so other warehouses are needed to support the shelter.

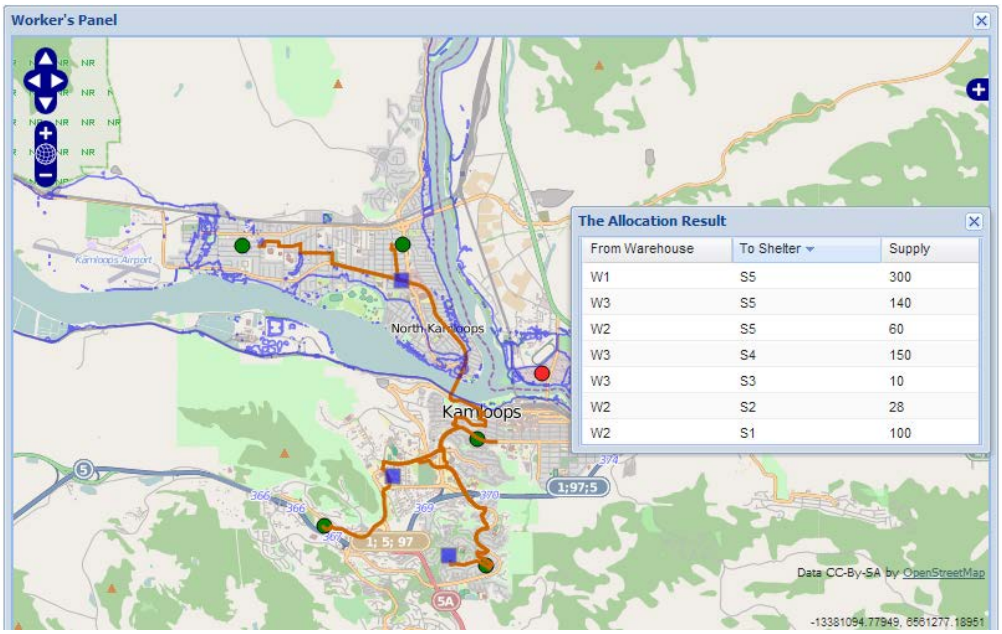


Figure 14. An example allocation solution for a large shelter.

b. Large warehouse

If I have a warehouse with enough resources, located at a good position as shown in Fig. 15, then this will be a very ideal condition. We can see from the figure that warehouse W3 sits in the middle of the city, which has a large amount of facilities. So, the application will provide a very good solution, where W3 will support most of the shelters.

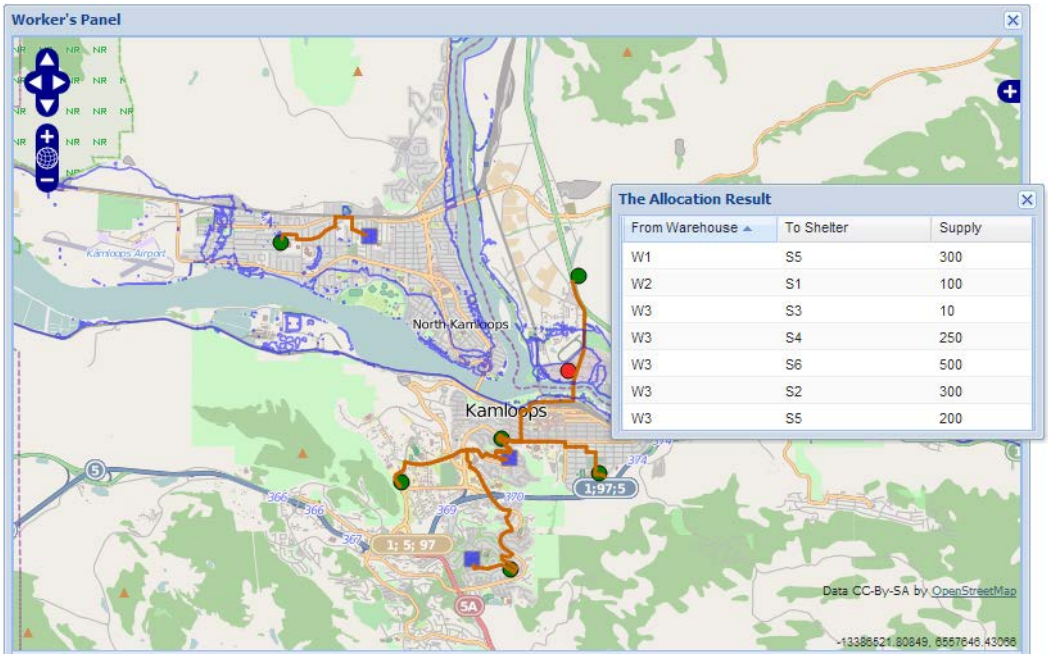


Figure 15. The allocation solution for a large warehouse.

c. Common conditions

Actually, optimal solutions come from careful preliminary work. As shown in Fig. 16, the application provides a highly efficient solution as long as the warehouses are designed well and store enough facilities, which means the warehouses should be placed at suitable locations so that each warehouse can cover a nearby area.

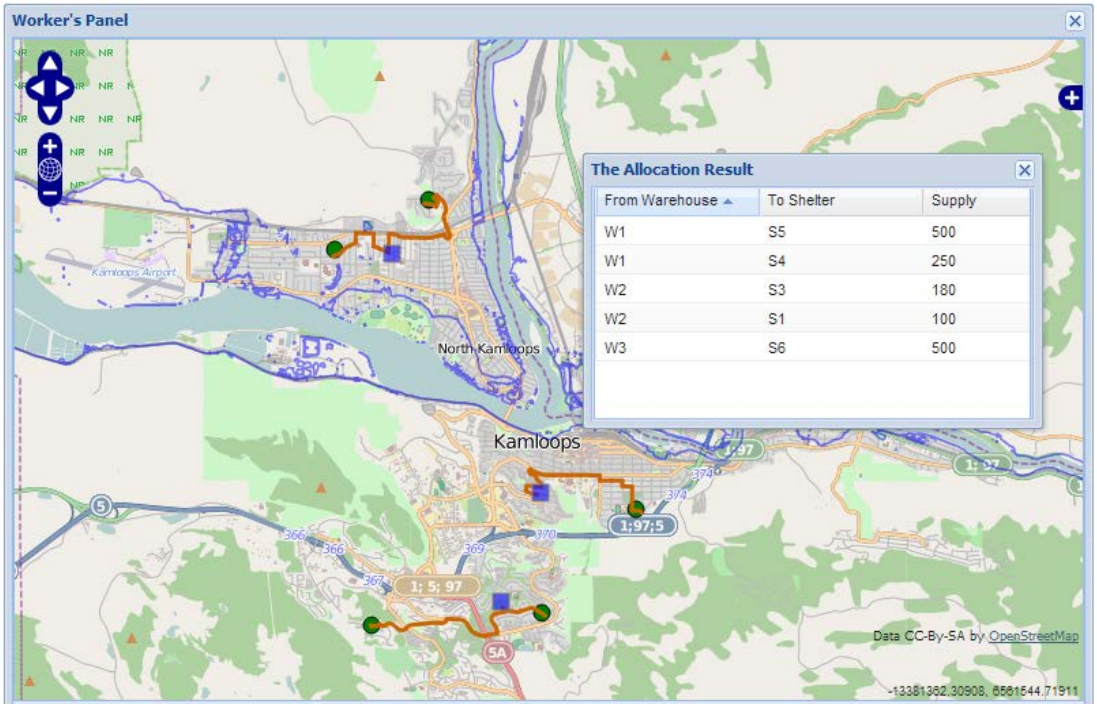


Figure 16. The allocation solution with a well-located warehouses and shelters distribution.

As mentioned above, White et al. (2010) have introduced a simple flood risk management system based on PPGIS. It allows people to submit a location's coordinate, together with the user's profile, and a comment he or she makes. Even a georeferenced discussion module can be included so people are free to discuss related topics. Further, spatial data layers can be added, such as spatial policy information, which allow people to declare and illustrate their opinions. However, this is only a one-way style of communication, and actually the information gathered from the public is not enough to help with either flood analysis or facilities management. In this study, I was trying to collect data from the public, and also publish related information to the public, which contributes to a real effort to mitigate flood-related disasters.

Another GIS-based generic real-time risk assessment framework and decision tools has shown itself to be very useful when facing environmental emergencies (Jiang et al., 2012). It is able to delineate the warning area and evaluate the hazards to functional area, societal impact, and human health and the ecological system. The stand-alone system and software components in this case were implemented on a GIS platform, which means it is specially-designed for decision makers. It is noticeable that while it includes powerful analysis tools, it still lacks public feedback. So in this study, although it is only a simple tool for analysis, I have tried to make the connection between the public and the emergency organization to share the information from both sides.

Compared to previous studies, this application has not only taken advantage of PPGIS,

but also has utilized IT knowledge and web services. It provides adequate information sharing between two important parties: citizens and emergency organizations. Considering what Lumbroso et al. (2011) pointed out, that there is a lack of response information, I have tried to include this information from the public into further emergency planing. By reducing the information gap during the response phase, a much more effective emergency plan can be created. I believe this is an important way to help with decision making for flood emergencies.

The application was developed following OGC standards, which means it's able to work with other map services providers, and it's easily understood and extensible by other people. Also, PostgreSQL had been selected as its database, so it maintains consistent data processing rules with other databases. Based on these conditions, this application has good portability to be used in different cities or areas, as long as the relational map data and street network data are available.

However, the application is built on a relatively simple model. It's still not close enough to the reality. To clarify the applicability and improve the application, there are several factors needed to be considered further:

(1) Applicability

The application was for study purpose and developed without a communication with Kamloops municipality, so any use of the application and related data needs the authorization of the municipality. Furthermore, any guidance to the public should be deliberate, confirmed, and responsible. It directly contributes to the safety of people.

(2) One-way problem

The application had not considered one-way problem, which means the road network can be used without limitations in direction. But in reality, some streets are designed for one direction use. If this factor is included, the routing algorithm needs to be improved further.

(3) Traffic factor

In reality, the communication system must be more complicated. Any accident or traffic jam can affect the validity of provided solutions. Under special conditions like a coming flood, there will be a high possibility of car accidents. The current application doesn't include this problem.

In future development, I can collect more detailed registration information from citizens, such as planned departure time, departure position, etc. But even if the information is provided, it is still a very complex problem to solve. Even better would be to have data describing the real-time traffic situation.

(4) Flood influence

It might be possible that some shelters have been affected by flooding. I have solved this problem in a simple way: Disable the affected or risky shelter. But actually, the city road network could be partially influenced by flooding as well. The current application cannot provide a solution that accounts for these conditions.

In future designs, a new attribute can be added into the road network table in the database, representing the availability of road segments. Given a shape file of the coming flood prediction, I can use the "Intersect" function to get the potentially affected road segments and disable them in the road network. Through this method, I can take this into account without

modifying other functions of the application.

(5) Internet load

When a scenario works, it's probable in a short time period, a large number of users are trying to access the servers, which could bring a heavy pressure to the system. It assumes that the hardware is strong enough to sustain the operation of the application.

7. Conclusions

By implementing the precepts of PPGIS, an interactive, web-based application was developed to administrate Kamloops' flood emergency service to the public. The application was mainly built using the JavaScript and the PHP language. A GeoServer database and OSM map data were adopted to support the background processing for the application. The PostgreSQL database was used on a large scale, with the spatial extension PostGIS to handle the spatial data types. With a coming flood, citizen users can register at one shelter and get the suggested path to that shelter; emergency workers have the power to manage the information describing-shelters and warehouses. The application can provide an efficient solution of the relevant Location-Allocation problem. This application facilitates data sharing between different parties. The information gathered from the public will help to make decisions in light of the coming flood. This application may promote the development of future natural risk management in a more effective direction, and it is helpful to connect with web sensors in vulnerable areas to help emergency workers evaluate flood risk. It can work with real-time traffic surveillance data to help people decide their best escape route. Furthermore, the idea of incorporating human-triggered input can be applied in other WebGIS fields, such as public health systems and land use planning, etc. Future research should also pay attention to mobile users, because Internet-enabled and location-aware mobile devices have played an important role both in normal life and under emergency situations in recent years, and with continue to do so in the future.

References

A Beginner's Guide to pgRouting, 2011. Available from: http://underdark.wordpress.com/2011/02/07/a-beginners-guide-to-pgrouting/ [Accessed May 2012].

Akay, A,E., Wing, M.G., Sivrikaya, F., Sakar, D., 2012. A GIS-based decision support system for determining the shortest and safest route to forest fires: a case study in Mediterranean Region of Turkey. Environmental Monitoring and Assessment 184:1391–1407.

Alexander, D., 2002. Principles of emergency planning and management. Terra Publishing, Harpenden and Oxford University Press.

Alexander, D., 2005. Towards the development of a standard in emergency planning. J Disaster Prev Manage 14(2):158–175.

Berry, R., Higgs, G., 2012. Gauging levels of public acceptance of the use of visualisation tools in promoting public participation; a case study of wind farm planning in South Wales, UK, Journal of Environmental Planning and Management, 55:2, 229-251.

Blazevic, A., 2006. Using Location-Allocation to optimize the Location of Fire Tanker Airbases in California. Available from: http://gradworks.umi.com/14/37/1437455.html [Accessed July 2012].

Brown, G., Montag, J.M., Lyon, K., 2012. Public Participation GIS: A Method for Identifying Ecosystem Services, Society & Natural Resources: An International Journal, 25:7, 633-651

GeoBase, 2009. Available from http://geobase.ca/geobase/en/about/index.html [Accessed March 2012].

GeoServer, 2012. Available from: http://geoserver.org/display/GEOS/Welcome [Accessed May 2012].

GeoJSON Format Specification, 2008. Available from: http://geojson.org/geojson-spec.html [Accessed May 2012].

GeoTour Guide for Kamloops, 2008. Available from: http://www.kamloops.ca/pdfs/brochures/KamloopsGeoTour.pdf [Accessed March 2012].

Kahn,M.E., 2005. The death toll from natural disasters: the role of income, geography and institutions. The Review of Economics and Statistics, 87(2), 271–284.

Kamloops Municipality, 2012. Available from: http://www.city.kamloops.bc.ca/index.shtml [Accessed March 2012].

Hong, J.H., Liao, H.P., 2011. Incorporating visualized data completeness information in an open and

interoperable GIS map interface. Journal of the Chinese Institute of Engineers 34(6), 733-745.

Jiang, J.P., Wang, P., Lung, W.S., Guo,L., Li, M., 2012. A GIS-based generic real-time risk assessment framework and decision tools for chemical spills in the river basin. Journal of Hazardous Materials. 227–228 (2012), 280– 291.

Lemmens, M., 2007. Disaster response starts with a map. GIM Int 21(4):13.

Li, K.Z., Wu, S.H., Dai, E.F., Xu, Z.C., 2012. Flood loss analysis and quantitative risk assessment in China. Natural Hazards, 63(2), 1083-1113.

Lumbroso, D., Stone, K., Vinet, F., 2011. An assessment of flood emergency plans in England and Wales, France and the Netherlands. Natural Hazards, 58(1), 341-363.

MacFarlane, R., 2005. A Guide to GIS applications in integrated emergency management. Emergency Planning College, Cabinet Office, P127.

Musungu, K., Motala, S., Smit, J., 2011. Participatory approach to data collection for GIS for flood risk management in informal settlements of Cape Town. Geomatics Division, University of Cape Town, Cape Town, ZA. Available from http://africancentreforcities.net/papers/63/ [Accessed August 2012].

OpenGIS Implementation Specification for Geographic information, 2010. Available from: http://www.opengeospatial.org/standards/sfs [Accessed April 2012].

OpenGIS Web Map Server Implementation Specification, 2006. Available from: http://www.opengeospatial.org/standards/wms/ [Accessed April 2012].

OpenStreetMap, 2012. Available from: http://www.openstreetmap.org/ [Accessed April 2012].

Pezanowski, S., Tomaszewski, B., MacEachren, A.M., 2007. An Open GeoSpatial Standards-Enabled Google Earth Application to Support Crisis Management. Lecture Notes in Geoinformation and Cartography, 225-238.

PgRouting Workshop Manual, 2011. Available from: http://workshop.pgrouting.org/ [Accessed May 2012].

PgRouting, 2012. Available from http://www.pgrouting.org/ [Accessed May 2012].

PostGIS, 2012. Available from: http://postgis.refractions.net/ [Accessed April 2012].

Quantum GIS, 2012. Available from: http://www.qgis.org/ [Accessed April 2012].

Regina, O. Obe., Leo, S. Hsu., 2011. PostGIS in Action. Manning Publications.

Sieber, R., 2006. Public Participation Geographic Information Systems: A Literature Review and

Framework. Annals of the Association of American Geographers, 96(3), 491–507.

Stewart, E.W.J., Jacobson, D., and Draper, D., 2008. Public participation geographic information systems (PPGIS): challenges of implementation in Churchill, Manitoba. The Canadian geographer, 52 (3), 351–366.

Tamayo, A., Granell, C., Huerta, J., 2012. Measuring complexity in OGC web services XML schemas: pragmatic use and solutions. International Journal of Geographical Information Science, 26(6), 1109-1130.

Tumay, A., Dincer, K., Tanyer, S.G., 2002. Usage of GIS and RS Technologies in Disaster Management. IEEE International Symposium on Geoscience and Remote Sensing (IGARSS) 2447-2449.

Vescoukis, V., Doulamis, N., Karagiorgou, S., 2012. A service oriented architecture for decision support systems in environmental crisis management. Future Generation Computer Systems 28, 593–604.

White, I., Kingston, R., Barker, A., 2010. Participatory geographic information systems and public engagement within flood risk management. Journal of Flood Risk Management, 3(4), 337-346.

Zhang, Q., Ding, Q.L., 2006. Heuristic algorithm for Location-Allocation problem based on wavelet analysis in integrated logistics distribution. WMSCI 2006: 10th World Multi-conference on Systemics, Cybernetics and Informatics, Vol VI, Proceedings, 28-33.

Zlatanova, S., Fabbri, A.G., 2009. Geo-ICT for Risk and Disaster Management. Geospatial Technology and the Role of Location in Science, Chapter 13. GeoJournal Library 96.

Institutionen för naturgeografi och ekosystemvetenskap, Lunds Universitet.

Student examensarbete (Seminarieuppsatser). Uppsatserna finns tillgängliga på institutionens geobibliotek, Sölvegatan 12, 223 62 LUND. Serien startade 1985. Hela listan och själva uppsatserna är även tillgängliga på LUP student papers (www.nateko.lu.se/masterthesis) och via Geobiblioteket (www.geobib.lu.se)

The student thesis reports are available at the Geo-Library, Department of Physical Geography and Ecosystem Science, University of Lund, Sölvegatan 12, S-223 62 Lund, Sweden. Report series started 1985. The complete list and electronic versions are also electronic available at the LUP student papers (www.nateko.lu.se/masterthesis) and through the Geo-library (www.geobib.lu.se)

225 Johanna Engström (2011) The effect of Northern Hemisphere teleconnections on the hydropower production in southern Sweden
226 Kosemani Bosede Adenike (2011) Deforestation and carbon stocks in Africa
227 Ouattara Adama (2011) Mauritania and Senegal coastal area urbanization, ground water flood risk in Nouakchott and land use/land cover change in Mbour area
228 Andrea Johansson (2011) Fire in Boreal forests
229 Arna Björk Þorsteinsdóttir (2011) Mapping *Lupinus nootkatensis* in Iceland using SPOT 5 images
230 Cléber Domingos Arruda (2011) Developing a Pedestrian Route Network Service (PRNS)
231 Nitin Chaudhary (2011) Evaluation of RCA & RCA GUESS and estimation of vegetation-climate feedbacks over India for present climate
232 Bjarne Munk Lyshede (2012) Diurnal variations in methane flux in a low-arctic fen in Southwest Greenland
233 Zhendong Wu (2012) Dissolved methane dynamics in a subarctic peatland
234 Lars Johansson (2012) Modelling near ground wind speed in urban environments using high-resolution digital surface models and statistical methods
235 Sanna Dufbäck (2012) Lokal dagvattenhantering med grönytefaktorn
236 Arash Amiri (2012)Automatic Geospatial Web Service Composition for Developing a Routing System
237 Emma Li Johansson (2012) The Melting Himalayas: Examples of Water Harvesting Techniques
238 Adelina Osmani (2012) Forests as carbon sinks - A comparison between the boreal forest and the tropical forest
239 Uta Klönne (2012) Drought in the Sahel – global and local driving forces and their impact on vegetation in the 20th and 21st century
240 Max van Meeningen (2012) Metanutsläpp från det smältande Arktis
241 Joakim Lindberg (2012) Analys av tillväxt för enskilda träd efter gallring i ett blandbestånd av gran och tall, Sverige
242 Caroline Jonsson (2012) The relationship between climate change and grazing by herbivores; their impact on the carbon cycle in Arctic environments
243 Carolina Emanuelsson and Elna Rasmusson (2012) The effects of soil erosion on nutrient content in smallholding tea lands in Matara district, Sri Lanka
244 John Bengtsson and Eric Torkelsson (2012) The Potential Impact of Changing Vegetation on Thawing Permafrost: Effects of manipulated vegetation on

summer ground temperatures and soil moisture in Abisko, Sweden

245 Linnea Jonsson (2012). Impacts of climate change on Pedunculate oak and Phytophthora activity in north and central Europe

246 Ulrika Belsing (2012) Arktis och Antarktis föränderliga havsistäcken

247 Anna Lindstein (2012) Riskområden för erosion och näringsläckage i Segeåns avrinningsområde

248 Bodil Englund (2012) Klimatanpassningsarbete kring stigande havsnivåer i Kalmar läns kustkommuner

249 Alexandra Dicander (2012) GIS-baserad översvämningskartering i Segeåns avrinningsområde

250 Johannes Jonsson (2012) Defining phenology events with digital repeat photography

251 Joel Lilljebjörn (2012) Flygbildsbaserad skyddszonsinventering vid Segeå

252 Camilla Persson (2012) Beräkning av glaciärers massbalans – En metodanalys med fjärranalys och jämviktslinjehöjd över Storglaciären

253 Rebecka Nilsson (2012) Torkan i Australien 2002-2010 Analys av möjliga orsaker och effekter

254 Ning Zhang (2012) Automated plane detection and extraction from airborne laser scanning data of dense urban areas

255 Bawar Tahir (2012) Comparison of the water balance of two forest stands using the BROOK90 model

256 Shubhangi Lamba (2012) Estimating contemporary methane emissions from tropical wetlands using multiple modelling approaches

257 Mohammed S. Alwesabi (2012) MODIS NDVI satellite data for assessing drought in Somalia during the period 2000-2011

258 Christine Walsh (2012) Aerosol light absorption measurement techniques: A comparison of methods from field data and laboratory experimentation

259 Jole Forsmoo (2012) Desertification in China, causes and preventive actions in modern time

260 Min Wang (2012) Seasonal and inter-annual variability of soil respiration at Skyttorp, a Swedish boreal forest

261 Erica Perming (2012) Nitrogen Footprint vs. Life Cycle Impact Assessment methods – A comparison of the methods in a case study.

262 Sarah Loudin (2012) The response of European forests to the change in summer temperatures: a comparison between normal and warm years, from 1996 to 2006

263 Peng Wang (2012) Web-based public participation GIS application – a case study on flood emergency management

264 Minyi Pan (2012) Uncertainty and Sensitivity Analysis in Soil Strata Model Generation for Ground Settlement Risk Evaluation

265 Mohamed Ahmed (2012) Significance of soil moisture on vegetation greenness in the African Sahel from 1982 to 2008