

Specification

Kryštof Hrubý

June 2020

1 Introduction

In order to have a nice biking experience it is quite pleasant to plan the bike route in advance and then view it on a phone during biking. Therefore one could use an interactive map editor to select a path between two or more endpoints. However, the mainstream softwares (e.g. mapy.cz, google maps, ...) are incapable of finding bike paths that would satisfy user's needs (e.g. preferred quality of terrain, maximum altitude, ...) or the functionality is fairly limited. This is the motivation behind the new software which should take account of such user's needs.

2 User requirements

The software's main functionality lies in finding paths between endpoints such that the user's needs (e.g. maximum altitude, ...) with respect to the path are accommodated. Communication with users is done through a web application.

Figure 1 shows use cases the software shall support. A standard map editor shall be present to provide standard functionality such as zooming and browsing in the map. A path shall be defined by two endpoints and possible path points which shall be in turn defined by clicking on a point in a map. It shall be also possible to add special constraints on the path such as maximum altitude. There also shall be a way to import/ export paths in GPX format.



Figure 1: Use case diagram

3 System architecture

The system consists of several important parts - javascript widget, user requests' handler, path search algorithm, handler of map data and preprocessing of map data to provide fast path search algorithm.

3.1 Javascript widget

Javascript widget handles communication on the client side.

3.2 User requests' handler

Main part of the system that handles the user requests and forwards them to the corresponding parts.

3.3 Map data handler

The map data are stored in a PostGIS database. This part provides API to other parts for easier retrieval of map elements. It should conform to the adapter pattern.

3.4 Path finding algorithm

This part handles all requests to find or update the best path between map points. In trivial situation looks for the shortest path. Otherwise it always tries to satisfy user preferences (e.g. maximum altitude for a path).

3.5 Preprocessing

Since the map data are so large, it is exhaustive to search the whole map to find a shortest (best) path between two points using plain A* or Dijkstra algorithms. Therefore, preprocessing is introduced to create additional structures in the map which can result in significant optimization. Such structures can be most visited points, shortcuts, reach of points, etc... There are a number of heuristics which can take advantage of these landmarks and provide an effective way to finding shortest paths.

The preprocessing can only be done for a given set of parameters and takes a lot of time; therefore, it needs to be finished before user requests. The heuristics then need to be able to find best paths even without the most suitable preprocessing.

4 System requirements

Let us go over the use cases in more detail.

4.1 Zoom map

4.1.1 Starting situation

Map editor is on the screen.

4.1.2 Normal

1. User moves mouse cursor over the map editor.
2. User uses the mouse scroll wheel to zoom in or out.
3. System updates the map editor with zoomed in/ out picture.

4.1.3 What can go wrong

If the map is zoomed in/ out to the maximum, nothing happens.

4.1.4 System state on completion

Map is zoomed in or out in the editor.

4.2 Move map in editor

4.2.1 Starting situation

Map editor is on the screen.

4.2.2 Normal

1. User moves mouse cursor over the map editor.
2. User left clicks on the editor and holds the mouse button.
3. User drags move to any direction.
4. System updates the editor to show map in the direction opposite to the cursor movement.

4.2.3 What can go wrong

Nothing can go wrong.

4.2.4 System state on completion

Map editor is updated.

4.3 Search in map

4.3.1 Starting situation

Map editor and search column is on the screen.

4.3.2 Normal

1. User fills in the search column the desired place.
2. System updates the map editor to show the place and its surrounding.

4.3.3 What can go wrong

If the place does not exist, the system alerts the user. If there are more than two such places, the system shows a menu of all options for user to select from them the correct one.

4.3.4 System state on completion

Map editor is updated to shown the desired place and its surrounding.

4.4 Add path point

4.4.1 Starting situation

Map editor and path points menu is on the screen.

4.4.2 Normal

1. User clicks on **Add path point button** between two existing points or before/ after all points (even to select the first point of the path) in the path menu.
2. User clicks on a location in the map editor where the path point is.
3. System recalculates the best path regarding previous points and the newly added one (if there is more than 2 path points).
4. System shows the new path in the map editor (If there is just one point, it shows just the point).

4.4.3 What can go wrong

Nothing can go wrong.

4.4.4 System state on completion

Path is updated in the map editor with respect to a new point.

4.5 Remove path point

4.5.1 Starting situation

Map editor and path points menu is on the screen.

4.5.2 Normal

1. User selects a **path point** in the path menu.
2. User chooses to delete the point with ‘delete’ key or via **delete point button**.
3. System recalculates the best path regarding the currently present points (if there is more than 2 path points).
4. System shows the new path in the map editor (If there is just one point, it shows just the point. If no point, just map is shown).

4.5.3 What can go wrong

Nothing can go wrong.

4.5.4 System state on completion

Path is updated in the map editor with respect to the deleted point.

4.6 Add path constraint

4.6.1 Starting situation

Map editor and path constraints menu is on the screen.

4.6.2 Normal

1. User selects and specifies a constraint in the path constraints menu.
2. System recalculates the best path regarding the new constraint.
3. System shows the new path in the map editor.

4.6.3 What can go wrong

If the new constraint is unreasonable/ impossible, the system sends user an alert about it.

4.6.4 System state on completion

Path is updated in the map editor with respect to the new constraint.

4.7 Remove path constraint

4.7.1 Starting situation

Map editor and path constraints menu is on the screen. At least one constraint different from a default constraint must be present.

4.7.2 Normal

1. User selects and clicks on **Delete/ Reset constraint** button in the path constraints menu.
2. System recalculates the best path with respect to the constraint default form.
3. System shows the new path in the map editor.

4.7.3 What can go wrong

Nothing can go wrong.

4.7.4 System state on completion

Path is updated in the map editor with respect to default constraint form.

4.8 Import path

4.8.1 Starting situation

Map editor and path constraints menu is on the screen.

4.8.2 Normal

1. User clicks on **Import path button** in the system menu.
2. System displays a standard file wizard.
3. User selects a GPX file.
4. System shows the path from the file in the editor.

4.8.3 What can go wrong

If the format of the GPX file is invalid no path is shown and the system alerts the user.

4.8.4 System state on completion

New path is shown in the map editor. If there was a path before, it deleted from the path menu and the map editor.

4.9 Export path

4.9.1 Starting situation

Map editor and path constraints menu is on the screen. A path defined by at least two points exists.

4.9.2 Normal

1. User clicks on **Export path button** in the system menu.
2. System sends user the current path in GPX file.

4.9.3 What can go wrong

Nothing can go wrong.

4.9.4 System state on completion

The system state is the same as was before the export.

4.10 Show altitude profile

4.10.1 Starting situation

Map editor and path constraints menu is on the screen. A path defined by at least two points exists.

4.10.2 Normal

1. User opens path details menu.
2. User selects **Altitude profile button**.
3. System shows a graph with an altitude profile of the current path.

4.10.3 What can go wrong

Nothing can go wrong.

4.10.4 System state on completion

Path details menu is opened and altitude profile shown.