

BoBus : interacting with city lines

Guillaume Kremer

Université de Bretagne Occidentale
g.kremer@wanadoo.fr

Dialect : Visualworks 7

Keywords : network management, CHI, routing

Website : <http://sourceforge.net/projects/bobus>

Developer : Guillaume Kremer

Category : student project

License : Open source

Abstract

Bobus is a simple application which goal is to provide easy transport management. It is currently presented on the city of Brest and its bus network and could help you finding your way.

Currently Bobus is just a toy tool made for training purpose. It has been started during my Licence 2 for a CHI¹ project. Very few improvements have been made since.

My recent incursions in distributed systems bring me a new look on it since this software provide an easy toolbox for network management and route computation.

1. Features

Basic

The software provide tools to edit and manage a network topology. With Bobus it is easy for an administrator to add bus stops, lines and to edit them simply by using the map. Networks can also be saved and loaded.

Advanced

Bobus has Dijkstra's algorithm as so the user can compute his direction by selecting his start and his destination.

¹Computer Human Interface

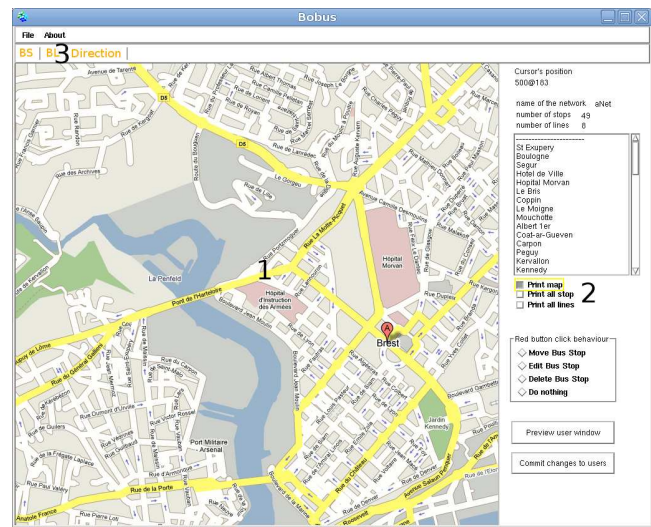


FIG. 1. Screenshot of the interface. Area 1 is the map. Area pointed by 2 is the control panel. In 3 we find the toolbox.

2. Using Bobus

Uses

Bobus can be used both by network managers and also by network's casual users since it embeds usual services for them. Managers can easily modify a network. Users can compute their travel and have a clear vision of a network.

Actually possibilities exist to split the application so that a clear distinction could exist.

Interface

Bobus interface is simple and made of 3 important panels : a map, a control panel for it and a tool board. See Fig. 1 for a screen dump of the interface.

The map

Stops and lines are displayed on the map. The user can also set a background image. For example a map of the area managed. The stops are shown in blue. Their name is printed above them. Each line have a different color. The stops and lines displayed, either all of them or just one, depend of the choice of the user, see following 'Control panel' section.

The map is not just a display. Managing can be executed by mean of the map. The stops can be moved by drag-and-drop, deleted with a click or simply selected for edition. The user shall also use the map during line creation or travel computation by picking the stops on it.

Control panel

This is the place to control how the map view is displayed.

User can decide to switch off the displaying of the map and to only show the path of the network (FIG. 2). This is useful for studying its topology and is similar to common network representation as the London underground² for example.

The user can also decide to have a global view of the network by switching on the buttons 'print all stops' and 'print all lines'. Individual stop or line can also be displayed by selecting it in the list view of the toolbox.

It has been chosen to only use the red button of the mouse for managing stops placement or editions on the map. Therefore user can tune the behavior of the red button while clicking on a stop with the red button in the control panel.

Adding a stop

The network manager can add a stop by clicking the 'BS' button of the tool box. The modal dialog

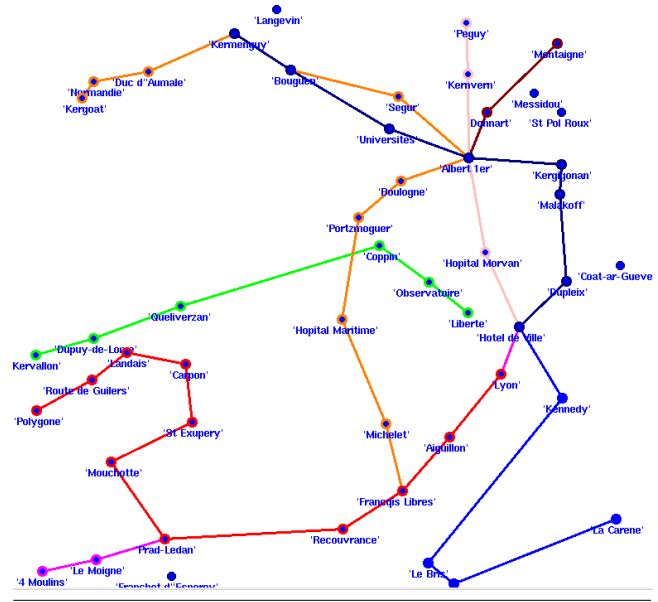


FIG. 2. An example of network's topology obtained with Bobus. This are the bus lines in Brest's downtown

appearing is shown in FIG. 3. One has to acknowl-

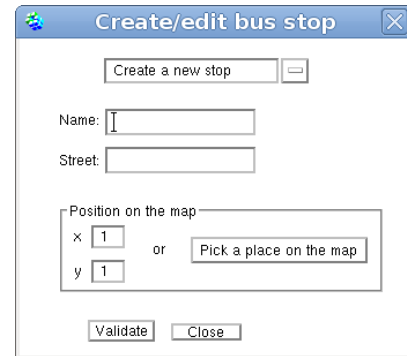


FIG. 3. The interface for creating or editing a bus stop.

edge the name of the bus stop, its street and its position. That one can be send either by filling the blank or by picking it on the map.

This interface exists for creating stops as well as editing them. Edition is done by choosing an existing stop in the combo box.

Adding a line

Clicking the 'BL' button pop the create or edit line interface(FIG. 4). The user has to fill the name

²see <http://www.tfl.gov.uk/assets/downloads/standard-tube-map.pdf> for the example of the London underground

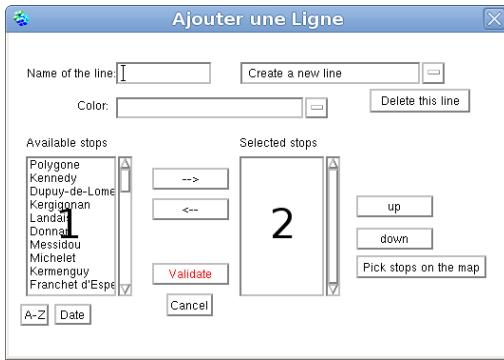


FIG. 4. The form for creating or editing a line.

of the network, its color, and to choose the stops belonging to it.

The stops placed in the list 1 are the existing stops that have not been placed on the currently edited line yet. The order of the stops has no importance here but one can choose to sort them by date or lexicographically by clicking respectively on the 'Date' or 'A-Z' button. At the beginning of a new edition, this list contains all the existing stops.

The list 2 contains the stops added to the current line. Here the stop order has its importance since it determines the order followed by the bus. Therefore stops can be put up or down. This is done by selecting the right stop and by clicking the 'up' or 'down' button.

A selected stop in list 1 can be transferred in list 2 by clicking the → button. Respectively a stop can be removed from the line by clicking on the ←. A more convenient method offered by Bobus is to click a bus stop on the map after clicking the 'Pick stops on the map' button. The selected stop is then removed from list 1 and added to list2 if it has not been selected yet.

To create the line, click the 'Validate' button. Similarly to the stop interface, one can edit a line here by selecting it in the combo box.

Computing a bus trip

Bobus can be used to compute a direction. The form used is shown in FIG. 5. User has to choose the start and destination stops. It can either choose the right ones in the combo box or select them

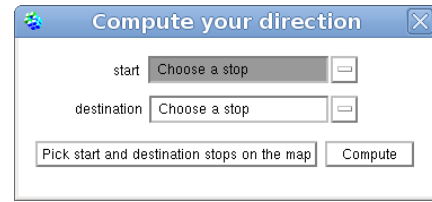


FIG. 5. The form for computing a trip based on the lines and stops of the network.

by picking near them on the map. A new form is displayed if a route exists between the start and destination stops which shows the result of the computation, see FIG. 6 for a sample. If no route exists an error is displayed.

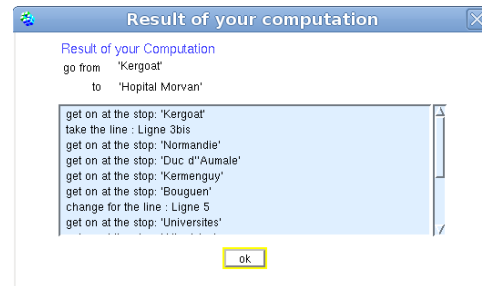


FIG. 6. Result of a route computation.

3. Internal structure

Bobus implements the MVC pattern. Both the admin and the user part are views which represent the same model. Changes in the network's topology are done at a high level by means of the admin view interface. Modifications at a higher level bring changes to the model and can therefore be reflected at the same time to the final bus users. This easy dispatching can be seen as a real-time network consideration (FIG. 7) while used in a distributed context, see following section : Future of Bobus.

MVC flexibility can also help to easily bring new services and evolutions without breaking down the model. The Company can add functionality to suit its need concerning printing information or public communication.

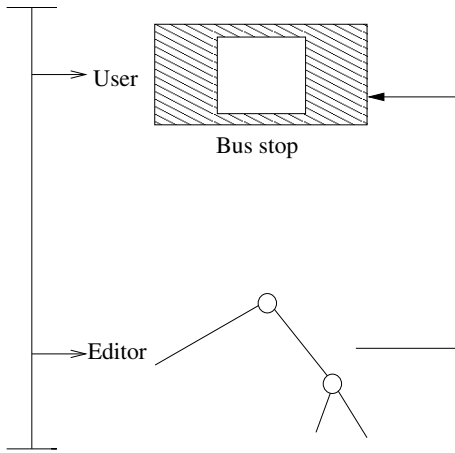


FIG. 7. Network manager acts on the topology and modifies in real time the model displayed on the view of the bus user.

4. Future of Bobus

Applications are straight-forward. One can imagine the user part of this application in a bus stop displayed on a flat touch screen (FIG. 8) and the admin part used in the network's HQ. Bobus



FIG. 8. An example of implantation, the network displayed in the stop is a Bobus view. Information displayed can change in real time.

shows a little preview of this now. Clicking the 'Preview user window' button shows us a typical bus user view displayed in a bus stop. One has to commit changes made in the main interface to have them displayed to final users.

Since Bobus has to be with traveling and mobility, it makes sense to port it on smart phones. If the device has a GPS, it could automatically find a way toward a destination point or inform the user

about near transport services.

Features should be added. Since we implement Dijkstra's shortest path algorithm one could want to tune the cost of each edge to use it efficiently.

For now, the time dimension is absent. Time tables could be introduced and administrators should be able to send and commit information like delays to bus stops. Bus lines could be punctual and exist for a limited amount of time as are lines put in place during public events.

Finally, Bobus may take advantage of the Seaside framework for displaying web contents as Google maps in place of the bitmap shown in the map.

5. Availability

Parcels, screenshots, samples and binary images are available on the project website :

<http://sourceforge.net/projects/bobus/>.

The old version of Bobus is also presents there.