Working Paper Series
ISSN 1177-777X

A map-based Place-browser for a PDA

David Bainbridge & Matt Jones & David Arter

Working Paper: 07/2007
October 24, 2007

©David Bainbridge & Matt Jones & David Arter
Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

A map-based Place-browser for a PDA

David Bainbridge
Department of Computer Science
University of Waikato
Hamilton, NZ
davidb@cs.waikato.ac.nz

Matt Jones, David Arter
Department of Computer Science
University of Swansea
Swansea, UK
m.jones@swan.ac.uk,
davea@sucs.org

ABSTRACT

This article describes PlaceBrowser, a PDA based application that allows the user of the application to navigate around an area of geographical interest, such as a city, using a zoomable, panable hierarchy of aerial images, in a fashion similar to Google Maps. The novel aspect to the work is that an area of precise interest within the map can be pin-pointed by the user by directly dragging out a rectangular area on the map. This forms the source to a spatial search that returns landmarks that are then used to trigger a Web based query. The results of this query are displayed to the user. The net effect is that, in response to dragging out a retangular area, web pages that are relevent to this area but have not been explicitly geo-spatially tagged with metadata (longitigude,latitude) are shown to the user.

Keywords: Mobile Searching, Context Aware, Spatial Search.

1 Introduction

This report describes the work undertaken in developing a map-based browser we have called the *PlaceBrowser*. The work fits in to a larger programme of work currently underway at the Future Interaction Lab (FITLab) at the University of Swansea that is studying location/context aware web searching. The work presented here builds on and modifies that of the Questions Not Answers (QnA) system by Jones *et al.* [2].

QnA uses a GPS-enabled PDA upon which is displayed an aerial map of the area the user is in, augmented over time with queries that *other people* have made when they were situated in the same area. Displayed queries are position on the map at the x,y coordinate (longitude,latitude) where the query was instigated. Using the PDA's stylus, if the user clicks on one of the displayed query terms displayed on the map then a web search using that term is initiated and the shown the ranked list of resulting matching documents. By seeding queries with automatically generated geo-spatially tagged "landmarks," our work considers a different dimension to location/context web searching.

The structure of the paper is as follows. First we go through a worked example that illustrates the main capabilities of the implemented system. Next the context to the work is provided, which includes background details to related work and the key data-structure used. Implementation details are then provided before we conclude with a summary of the work and future plans, which includes a field trial of the system with two other mobile searching prototypes.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

©2007 University of Waikato

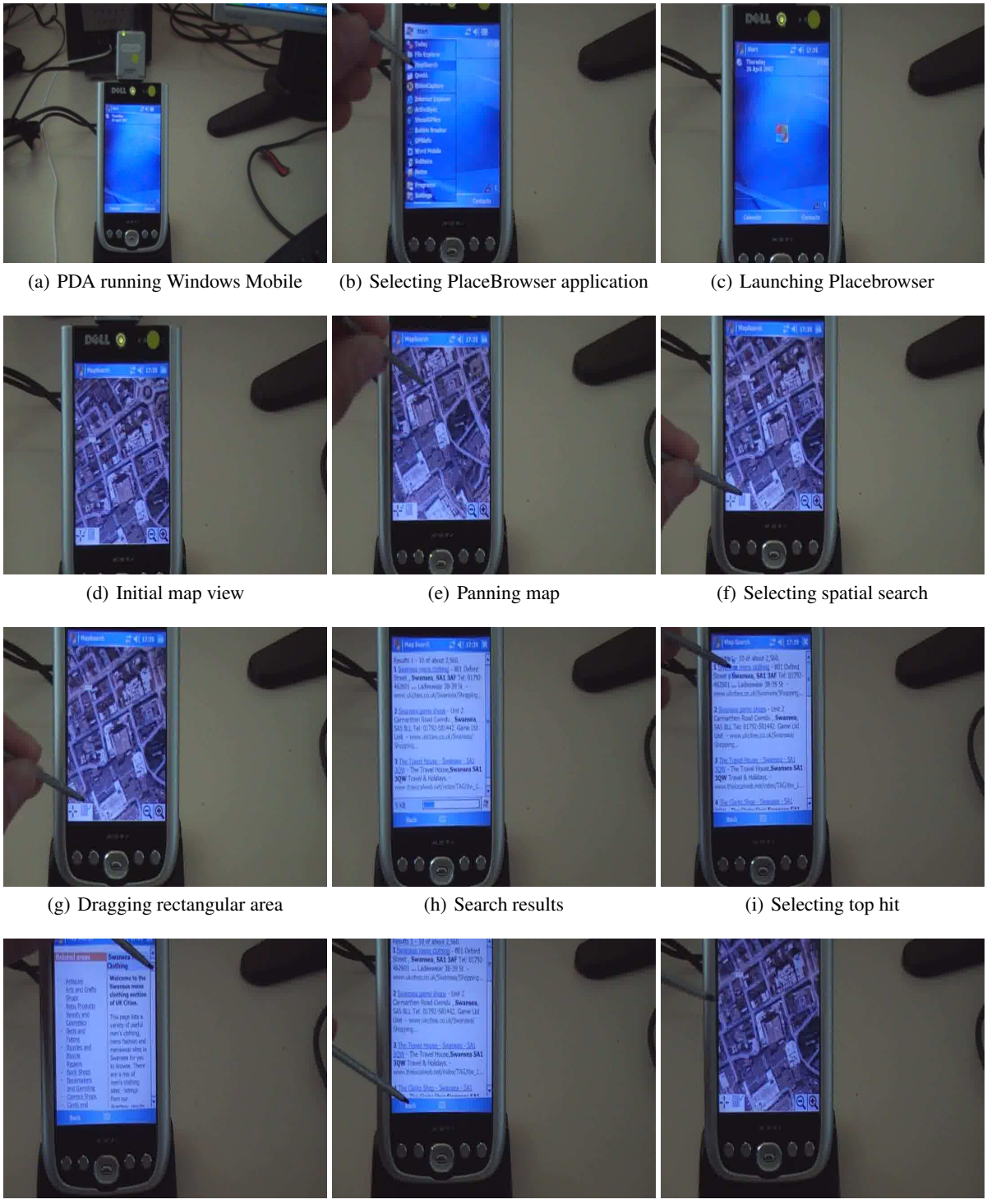
2 Worked Example

In PlaceBrowser, an aerial map is provided (*à la* GoogleMaps) with panning and zooming capabilities. No textual details are shown, however, other than street name information that comes as part of the rasterised hybrid aerial view. The user spatially navigates around the map and when they see an area of interest, they draw a rectangular box enclosing it. This triggers a web search where behind the scenes GIS-encoded data about street/placenames is used for query terms, and the ranked results displayed to the user.

Figure 1 shows a sequence of still shots taken from a video of a typical user session.[1] First the user launches the application, which is located in the top-level applications menu as seen through Figures 1(a)–(c). After a short delay PlaceBrowser open up with an aerial view, in this case Swansea city centre (downtown) although it can equally be tied to the GPS position of the user—the receiver can be see back in Figure 1(a)—although startup takes longer as it needs to lock on to satellite transmissions.

At the bottom of the PlaceBrowser application are some overlay buttons. Two on the left, and two on the right. The left-most ones control pan and spatial search, and the right-most control zooming in and out. Be default the application starts in pan mode, and in Figure 1(e) the user is dragging the stylus on the map to change the area displayed. In Figure 1(f) they are changing the mode to spatial search. Now when they drag the stylus across the

[1]The original video footage can be viewed through *www.nzdl.org/epsrc-fellow-2007/pda*.



(a) PDA running Windows Mobile (b) Selecting PlaceBrowser application (c) Launching Placebrowser

(d) Initial map view (e) Panning map (f) Selecting spatial search

(g) Dragging rectangular area (h) Search results (i) Selecting top hit

(j) Swansea clothing for men (k) Closing the browser (l) Returning to map view

Figure 1: Snapshot sequence of using the PlaceBrowser.

map, as they are doing in Figure 1(g), instead of panning a rectangular outline is drawn to signify the area that a spatial search will be based on. The user continues adjusting the position of the stylus until they are satisfied with the area outlined.

Upon lifting the stylus away from the screen, the search is initiated. Using an Internet connected (*e.g.* wireless), after a short delay the results of the query are shown. Figure 1(h) shows the result page for the spatial query (centred in our example on the large roofed building position in the middle of the screen) and Figure 1(j) the page that results from choosing the top match, a page about Swansea Clothing For Men, which is indeed a store housed in the building in question. Finally, in Figure 1(k) the user has clicked on the browser close button at the bottom of the screen to return back to the map view.

3 Background

Location/Context-aware browsing is an emerging field in which Jones, along with other colleagues at the FITLab, are making significant progress.

At the core of the PlaceBrowser work is a spatial search capability. KD-Trees are an efficient, well known and much studied technique for this [1, 3] which—unlike GIS-enhanced features for relational database systems such as PostGIS[2]—meets our requirement of something that can be performed locally on the mobile device. Using KD-Trees, construction of this spatial data-structure is $O(N \log N)$ for N points. Worst case for spatial searching is $O(N \log N)$ which corresponds to the user drawing a rectangle that *entire* map. This would require a certain amount of zooming out in the PlaceBrowser application to achieve this, and so is highly unlikely in our application. Best case is $O(\log N)$ with the majority of typical searching only slightly greater than this on average. For these reasons KD-Tree was chosen as as a basis for this work.

Two GIS-labelled dataset were experimented with: postcodes (zip-code in US *parlance*) and streetnames—the latter being fairly straightforward to derive once the GIS-labelled postcode were obtained. Unfortunately GIS-labelled postcodes for the Swansea area (or any UK city for that matter) were not readily available and some effort was expended deriving a suitable solution. In the first instance, a sweep of gazetteers in the public domain yielded nothing of any use. Further investigation uncovered the main impediment: despite the Ordnance Survey and the Royal Mail being state owned entities, government directives dictate that they should be profitably, and this in turn has led to them only making such data available commercially. In this regard, the UK is not as enlightened as its US where legislation ensures that "copyright protection is not available for any work of the United States Government" effectively putting it in the public domain.

The UK situation is in fact quite perverse:[3] when a council signs off on a newly named street in their city, they are legally required to provide detailed GIS data to two state-owned entities: Ordnance Survey and the Royal Mail. These organisations assimilated this data into their wider dataset, which is by all accounts, a fairly automated process. Next comes the twist: should a council wish to provide a map-based web service (say establish the closest bus stops to your home, based on street address or postcode information) the council has to pay the state agencies for access to data that, for the main part, they created! And it is not just a question of a modest one-off annual fee—the cost is based on the number of requests to the central database, which of course if directly proportional to how often people use the council's website. The royal mail is making nearly 2M pounds profit per year. Trying to rectify the situation has resulted in a mess that has snowballed over the last five years with ever-spiralling costs to the UK tax-payer, and led to the Free our Data campaign spearheaded by the Guardian newspaper and other concerned groups. At the time of writing (August 2007) the issue remains unresolved.

A solution to acquire the necessary GIS information was devised using information available through map-based web sites such as GoogleMaps and StreetMap. Details of this along with other aspects of the implementation are given in the next section.

4 Implementation

Development work was carried out in C# using Visual Studio for the Pocket PC. The PDA used was a GPS-enhanced Dell Axim X51v running a 624 MHz processor with 64 Mbytes of main memory, and 190 MBytes storage capacity; a further 512 Mbytes was available on the GSP SD card. Its screen resolution was 640 X 480 and an Internet connection (through a host-machine) was available using either bluetooth or the PDA's docking cradle. Wireless was also available on the device, but the environment the PDA was operated in did not permit the device to be connected.

There were two parts to the software written: a batch driven, command-line program written in Perl that "page-scraped" online map-service sites to develop the necessary gazetteer; and the PlaceBrowser application written in C# that used this gazetteer in combination with map data and the Internet connection to provide the desired functionality. All code is stored in a Subversion repository hosted by Swansea University's Computer Society.[4]

4.1 Gazetteer generation

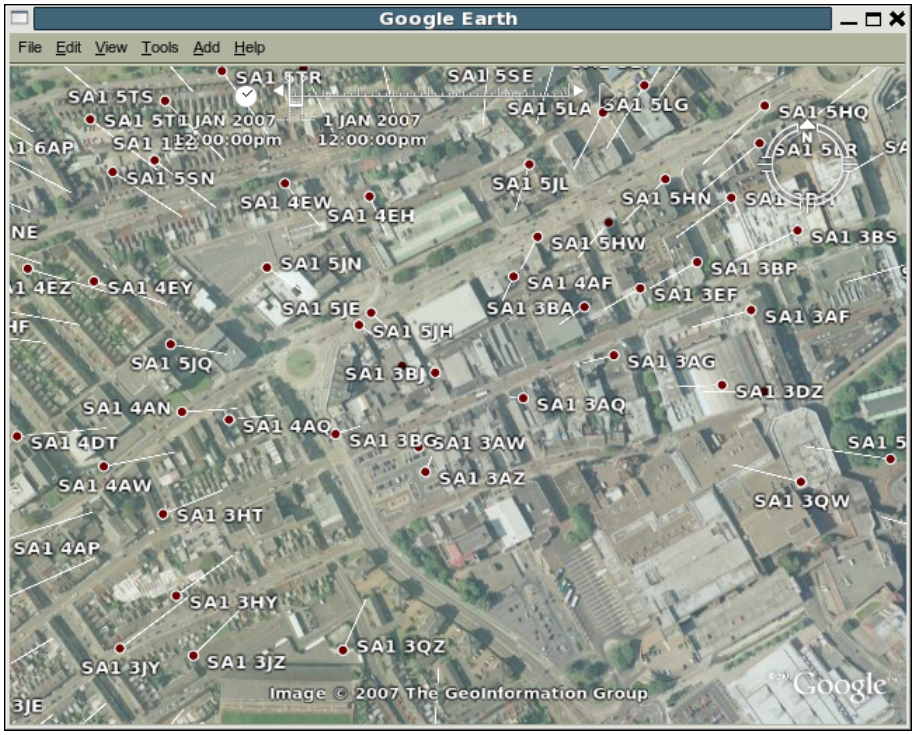
Figure 2 shows the street and postcode gazetteers that were developed and experimented with. Perl code was written to collate this information. Output is in the XML-based KML (Keyhole Markup Language) format. Keyhole was the development name for what eventually became GoogleEarth, and KML was developed to allow users to exchange map annotations. It is useful for our purposes as it is straightforward to parse by our application, with the added bonus that is can be easily displayed visu-

[2] See *postgis.refractions.net* for more details.

[3] See *www.guardian.co.uk/technology/2006/nov/16/ epublic.guardianweeklytechnologysection* for more details.

[4] The code can be checked out with the command "svn co https://sucs.org/ davea/svn/qna/gazetteer". Currently there is no anonymous checkout, you'll need a username and password. Contact Dave Arter davea@sucs.org for this.

(a)



(b)

Figure 2: Gazetteer generated location information: (a) Streetname (b) Postcode.

ally through the GoogleEarth applications—for checking purposes and the generation of figures for papers written about the work! For further information on KML see the Wikipedia entry.[5]

The Perl code was written using the coding practises of the Greenstone digital library project, of which the principle author of this paper is a key member. Document ingest is a significant portion of the processing work done in forming a digital library, and in Greenstone this stage is implemented in Perl. Multimedia documents, including maps, is the focus on on-going work in the Greenstone project, and there is the potential for the software developed for the PlaceBrowser to be further developed through this connection.

Having sourced the setup file (setup.bash), which augments the user's environment so the gazetteer generating software can be run, a six step process produces the required KML files:

1. Sample a rectangular grid (longitude,latitude) over a chosen area (*e.g.* Swansea) using *street-maps.co.uk* to generate a set of webpages about each sample point; save this data locally.

2. Page-scrape Street-Maps' downloaded pages for postcode data, and store in a flat-file database.

3. Use *street-maps.co.uk* again to refine precise (longitude,latitude) of postcode, as the chosen co-ordinate in Step (1) is unlikely to be the precise location of a defined postcode. Inside the web pages initially downloaded from the site is a table entry that states what the *nearest* postcode to this location is. The longitude and latitude position for this item is what is used to query Street-Maps for a second time. Again a local copy of the generated page is saved locally.

4. Page-scrape Street-Maps' pages for the *true* position of postcodes (longitude,latitude), and save this information in a separate flat-file database.

5. Use Google Maps to generate pages with street name information in it per postcode in the flat-file database. This is done by asking Google to generate direction from getting from point A to point B. The points can be given as postcodes. Make both the start and finish points the same, and save the generated web page locally to disk.

6. Page-scrape Google Maps' pages for street name (the only street mentioned in the "directions" section of the page).

Gnu's DataBase Management library (GDBM) was chosen as the flat-file database system used. One of the modules to Perl (GDBM_File) allows you to map (using the keyword `tie`) a GDBM database to an instance of the language's built-in associative array data-type. For the purposes of programming, it is as if you have the data resident in memory, accessed by string-index. The GDBM_File Perl module takes care of mapping the in-memory data-structure to the database format.

[5] http://en.wikipedia.org/wiki/Keyhole_Markup_Language

Despite the street name gazetteer being visually more appealing than its postcode counterpart (Figure 2) and therefore to more naturally convey better information for initiating web searches once a user has dragged out a rectangular region, trials showed that it did not yield as high a precision as postcodes; consequently the postcode version of the gazetteer was used in the final version.

4.2 PDA application

Implementation of the placebrowser application was straightforward overall, given the existing functionality of the Question not Answers system that allowed for panning and zooming a tiled hiearchy of aerial images. The coding to include the necessary user interaction sought for the placebrowser application was accomplished through inherience—no structural changes to the existing codebase were made—the key work being the inclusion of a rubber-banding feature to draw the area for the spatial search. Double buffering was used to make this flicker free.

The other key component to the implmentation was reading in the gazetteer to build the KD-tree. Standard .NET classes for XML parsing were used to read in the KML file, and the necessary 2-dimensional tree built, alternating between $x-$ and y-dimensions as a node's key at successive layers of the tree. Spatial search (also called orthogonal range in some fields) is a standard application of recusrive programming.

One technical hitch encountered during the programmning centred on the web browser form supplied by the .Net framework. As far as we could deduce, constructing a URL with CGI symbols such as `?` and `&` caused the panel to fail to load the stipulated URL, instead returning an error message. Different search engines were trialled, but to no avail. Careful consideration was given to the problem: different methods for encoding these special characters were utilized, however the problem remained. If the constructed URL was instead displayed on the screen, and then this was manually copy and pasted into the PDA's provied web browser (IE) then the search was performed correctly, and the appropriate result page returned. The closed source nature of the .Net framework was a considerable frustration here. Further effort was spent looking for descriptions (and solutions) to the problem on the web, but this turned up nothing useful. It is the conclusion of the authors that the problem is a bug in the Microsoft Compact Framework's .Net form for web display.

A workaround was developed to allow the project to proceed. This was:

1. opening a URL connection

2. downloading the file

3. modifying the downloaded HTML so relative links work correctly

4. instructing the .Net browser form to open up the local modified file instead.

Fixing relative links was done using a regular expression that matched and then replaced particular HTML attributes. This could have equally been done with the addition of a `<base>` tag instead.

5 Conclusion

This technical report has described the work undertaken in developing PlaceBrowser, a PDA based application that allows a user to pan and zoom a geographic area represented through a series of aerial photos. The city of Swansea in Wales was chosen as the sample location. Using the application, the user can trigger a spatial search by directly drawing on the screen a rectangular area that is of interest to them. Based on "landmarks" represented as a KD-tree, those points falling in the area of interest are used as query terms to an web search engine. Through empirical experimentation we found postcodes of the area to work well. These postcodes, along with their longitude and latitude positions were collated through a process of sampling web pages from *street-maps.co.uk* and *maps.google.com*.

References

[1] J. L Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, September 1975.

[2] Matt Jones, George Buchanan, Richard Harper, and Pierre-Louis Xech. Questions not answers: a novel mobile search technique. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 155–158, New York, NY, USA, 2007. ACM Press. ISBN 978-1-59593-593-9. doi: http://doi.acm.org/10.1145/1240624.1240648.

[3] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, MA, 1990.