AN OPEN-SOURCE WEB ARCHITECTURE FOR ADAPTIVE LOCATION-BASED SERVICES

Gavin McArdle, Andrea Ballatore, Ali Tahir, Michela Bertolotto

School of Computer Science and Informatics
University College Dublin
Belfield, Dublin 4, Ireland
gavin.mcardle@ucd.ie, andrea.ballatore@ucd.ie, ali.tahir@ucd.ie, michela.bertolotto@ucd.ie

KEY WORDS: Location Based Services, GIS, Personalisation, Web Technologies, Adaptive Interfaces

ABSTRACT:

As the volume of information available online continues to grow, there is an increasing problem with information overload. This issue is also escalating in the spatial domain as the amount of geo-tagged information expands. With such an abundance of geo-information, it is difficult for map users to find content that is relevant to them. The problem is intensified when considering Location-Based Services. These services, which are dependent upon a user's geographic location, generally operate on portable devices. These devices have a reduced screen size coupled with a limited processing power and so the need to provide personalised content is of paramount importance. Our previous work has focused on examining techniques to determine user interests in order to provide adapted and personalised map content which is suitable to display on portable devices. In this paper, in order to reduce the processing load on the user's device, a novel client server architecture is employed. The framework is designed using open-source, web-based technologies which monitor user locations and interactions with map content overtime to produce a user profile. This profile is then used to render personalised maps. By utilising the power of web-based technologies in an innovative manner, any operational issues between different mobile devices is alleviated, as the device only requires a web-browser to receive map content. This article describes the techniques, architecture and technologies used to achieve this.

1 INTRODUCTION

Information overload is a ubiquitous problem which is increasingly prevalent in the web domain. The issue is also present in the spatial domain where the quantity of spatially-referenced material has increased (Yang and Claramunt, 2005). Techniques which have been utilised in the predominantly text-based web domain are now being explored in the context of geo-spatial content. Traditional systems permit the user to personalise map content explicitly however such techniques can be time consuming and distorted by subjectivity. Furthermore, these approaches often distract the user from the main task at hand (Wu et al., 2008).

Monitoring interest indicators implicitly in order to obtain an insight into user preferences can alleviate this problem. Implicit profiling involves the system monitoring user interaction with an underlying interface which permits the automatic personalisation of information, and in the context of spatial data, the adaptation of map content. As implicit profiling is unobtrusive, it has a 100% completion rate and does not place any additional overhead on the user. This approach is employed successfully in several web-based systems where user actions such as link clicking, bookmarking and printing act as indicators of interest in a web-page's contents (Kelly and Teevan, 2003). Recently in the spatial domain similar interest indicators have been identified via interactions with map content (Brunato and Battiti, 2003; Mac Aoidh et al., 2007; Weakliam et al., 2005).

The need for personalisation of spatial data is further emphasised when dealing with Location-Based Services (LBS). Such services typically operate on mobile devices which often have limited processing power coupled with a restricted screen size and so there is a requirement to reduce the amount of information presented through personalisation and adaptation of spatial data. There are a multitude of diverse platforms and operating systems used in state-of-the-art mobile phones and developing a suitable LBS which can perform in such varied environments is challenging. Several standalone applications have been conceived (Kupper, 2005), however they are typically restricted to one class of device.

An approach to provide personalised LBS to users in a more accessible way is required. The research presented in this paper utilises open-source, web-based technologies which adhere to internet and Open Geospatial Consortium (OGC) standards to achieve this. Utilising web-based technologies permits a flexible client-server architecture to be implemented. Such an architecture, where processing can be distributed between server and client, reduces the computational load on the mobile device. Furthermore, open-source, web-based technologies are ubiquitous and only require the user device to be equipped with a suitable web-browser thereby alleviating most of the cross platform operational issues while also increasing the accessibility. In this paper the power of the client-server paradigm is combined with recent web technologies and applied in the context of adaptive LBS. Existing techniques which implicitly monitor user behaviour and context to provide personalised map content are demonstrated using this new approach.

The remainder of this paper is organised as follows: section 2 discusses related work in the area of user profiling, spatial recommendation systems and web-based technologies for developing spatial applications. section 3 outlines a sample application including the web-based architecture with a detailed description of the technologies and core algorithm used. Section 4 details the prototype which has been developed to evaluate the approach using a test scenario, while section 5 identifies limitations and directions for future work. In section 6, conclusions on the paper are made.

2 RELATED WORK

Although there is an extensive body of research in web personalisation (Albanese et al., 2004; Middleton et al., 2002), personalisation in the LBS domain remains relatively unexplored. The systems which have emerged use diverse techniques for both profiling and recommending and have not been widely adopted. For example, Hippie (Oppermann et al., 1999) is a personalised location-based content delivery service, tailored specifically for museums. Interests are inferred by analysing the information which the user viewed, in order to recommend content based on their position within the museum. Other researchers focus on interactions at the map level as a means of determining interests. Weakliam et al. (2005) monitor user interactions with the layers within the map and use this information to recommend semantic groups of objects. Likewise, the system described by Mac Aoidh et al. (2007) monitors user interactions with map content, focusing on the mouse position relative to individual map objects as an indicator of interest. This information is used to automatically generate a user profile from which relevant map objects can be recommended in the future. Similarly, PILGRIM (Brunato and Battiti, 2003) is a location-aware system which utilises the physical position of users as an interest indicator by assuming that proximity to objects is relevant in determining interests.

Such LBS recommender and personalisation systems tend to be standalone applications and restricted to a specific device type for a particular domain. These applications fail to take advantage of recent developments in web-based technologies which have been developed for geo-spatial data management and display. These services offer the potential for multi-platform deployment of LBS systems which can personalise information and adapt a map for users.This web setup offers several advantages over the traditional standalone approach. The use of the World Wide Web to give access to geographical information dates from 1993 (Longley et al., 2005). Since then GIS has successfully adapted to the Internet paradigm and has benefited from the momentum generated by the web, and the history of GIS and that of the Internet have become increasingly intertwined. Today, there is a fast-growing number of GIS applications deployed on the Web, ranging from mapping to routing and geographical yellow pages. The reasons for this success are various: the web is an established, widely used platform where accepted standards allow for smooth integration and manipulation of heterogeneous data types. Furthermore, the interactive and exploratory nature of navigating geo-referenced hyperlinked information is appealing for the end users. The shift from standalone GIS to webGIS is apparent. The use of standalone GIS has dramatically decreased in recent years (Longley et al., 2005).

Furthermore, thanks to the growing spread of Internet-enabled and location-aware smartphones, surveyed by Oliver (2009), WebGIS is increasingly becoming portable. In this fast moving context, the term *GIServices* defines the web applications offering distributed access to centralised spatial contents. Recent advances in client-side technologies allow for the development of complex interfaces that can be run on a web browser without installing additional software components thanks to the use of widely adopted standards. These standards have been utilised in numerous pieces of GIS software. As economic factors drive the shift towards open-source technologies (Sui, 2008) many GIS tools have been released under licenses similar to the GNU Public License (GPL). Since their introduction in the 90s, such licenses permit the sharing and distribution of source code and have boosted the development of open-source technologies. The quality of this software, ranging from mapping tools to topology libraries, has been constantly improving and in some cases has matched the functionality offered by commercial competitors. The significance of open-source software usage and development in the geo-spatial community is increasing. Sanz-Salinas and Montesinos-Lajara (2009) have surveyed this open-source ecosystem.

This paper describes an approach which utilises the benefits of open-source, web technologies to define a three-tier component based architecture to produce an adaptive map-based LBS. The resulting system is suitable for use on multiple devices via a standard web browser thus minimising cross platform operational issues. Map adaptation is achieved using techniques discussed by Ballatore et al. (2010) where an algorithm which implicitly monitors user interaction, in order to recommend tailored spatial content, is described. This algorithm is used in a prototype of the new architecture.

3 SYSTEM

The system proposed is an open web platform for spatial personalisation and visualisation. Based on a client-server architecture, the system delegates the task of computing complex algorithms, generating visual output and dealing with costly operations to the server. The clients, on the other hand, send requests to the server and render the output for the user. Within this domain interoperability is crucial. Firstly, input and output spatial data have to be defined in well-known formats. Secondly, the web-based approach minimises the coding overhead necessary to port the application to different mobile and desktop platforms. Last but not least, Open Web Services and an API expose the system functionalities on the Internet and allow external applications to interact with the system. The following section describes in detail the system architecture, while section 3.2 moves on to describe the personalisation algorithm.

3.1 Architecture

The architecture of the system proposed is structured in 3 tiers: client tier, middle tier and data sources. This approach emphasises the independence of the various components of the system, that can be deployed and combined in different contexts. Figure 1 illustrates the interaction among tiers, outlining the main components of the system and their logical position. The following sections describe these tiers in detail.

3.1.1 Client Tier The user interaction with the system takes place in this tier. The user visualises web pages containing spatial information on their device through a common web browser. These web pages display interactive maps and monitor certain actions performed by the user, such as mouse clicks, zoom, etc. Such actions get sent to the *Personalisation and Visualisation Service*. In a typical scenario a web page displays a dynamic map served by the *Map Renderer* and other information from the *Personalisation and Visualisation Service* (both situated in the Middle Tier).

3.1.2 Middle Tier This tier contains the core services and functionalities of the system. A web application hosts the Web Server Pages which constitute the access point to the system for the users and the *Personalisation and Visualisation Service*, in which several web services are deployed and exposed on the Internet.

This *Personalisation and Visualisation Service* tracks the user sessions, handles the user profiles, logs all of the relevant actions performed by the clients and keeps track of the user location when available. The personalisation and visualisation algorithms are also implemented within this service to take advantage of the server-side computational power and full access to

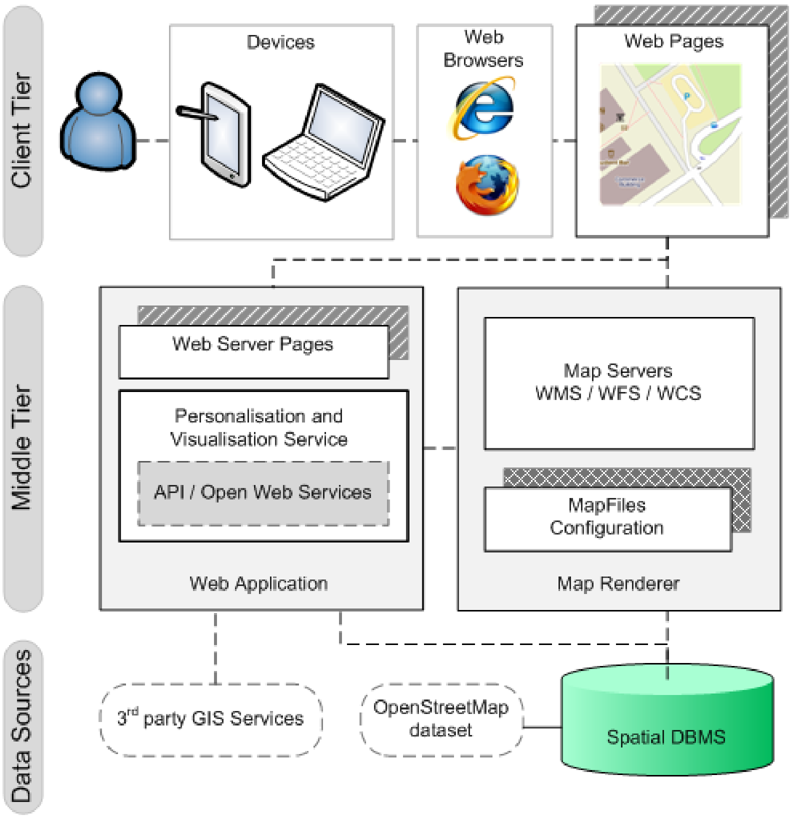


Figure 1: System architecture

the spatial datasets and recorded interaction. The *Personalisation and Visualisation Service* can be accessed by external applications through an API defining functions with parameters and the format of the returned values. The personalisation algorithm described in Ballatore et al. (2010) has been implemented in this service and it is outlined in section 3.2.

The *Web Server Pages* define the dynamic contents that are served to the clients. A Web Server Page can aggregate spatial and non-spatial data from heterogeneous sources, tailored for a specific user. Such a page gets rendered and delivered to the client. The navigation logic among these pages is defined and controlled by the Web Application.

At the same architectural level a *Map Renderer* is deployed. This component is able to load spatial data and to render them according to a dynamic map file configuration, which defines what data have to be rendered and with which visual style. The map file configurations are generated dynamically by the *Personalisation and Visualisation Service* according to the profile of the user receiving the spatial data. The *Map Renderer* hosts several map servers, which are spatial web services specifically designed to provide the clients with spatial information in standard and widely-adopted formats.

Both the *Personalisation and Visualisation Service* and the *Map Renderer* interact with the Data Sources to obtain and store the spatial information needed to execute their tasks.

3.1.3 Data Sources The main function of this tier is to provide the other tiers with spatial information. Its main component consists of a *spatial DBMS*, which stores spatial datasets (e.g. vector maps) and data related to the user profiles. The spatial DBMS is used by the *Personalisation and Visualisation Service* to run the personalisation logic, whilst the *Map Renderer* accesses it to perform the visualisation. The relevant spatial datasets can be imported from an external data provider (e.g. OpenStreetMap, etc) and stored in the spatial DBMS and updated on a regular basis.

Another apparent phenomenon on the Internet is the growing availability of commercial GIServices accessible for free (e.g. Virtual Earth, Google Earth, Google Maps and Cloudmade). These services rely on high-end commercial infrastructures to provide access to various datasets through Web APIs, mostly for routing, geocoding and reverse geocoding. The Web architecture described in this section lets the Web Application request data from such web services for with little development cost.

3.2 Personalisation Algorithm

This personalisation algorithm described by Ballatore et al. (2010) has been implemented in the *Personalisation and Visualisation Service*. The user sees the world as a set of *items*. An item is a point of interest representing a geographical entity (either a point or a polygon) which the user might have interest in. Certain types of item (such as shops, cinemas, etc) can be associated with relevant resources on the Web (such as web sites or web services). These items are displayed on an interactive digital map, rendered by the *Map Renderer*.

The *Personalisation and Visualisation Service* assigns an interest score to each item located within the current *interest radius*. The interest radius is inversely proportional to the current user speed. The interest score α for the item i is calculated with Equation (1) taking into account both historical interactions (*interaction*) and current user routing distance from the item (*proximity*)

$$\alpha_i = P_i P_R + I_i I_R$$
$$P_R + I_R = 1, \quad P_i \in [0,1], \quad I_i \in [0,1] \tag{1}$$

P_R and I_R are respectively the *proximity ratio* and the *interaction ratio*, meaning the weight the system attributes to each indicator. Those ratios are dynamic and change over time: the more the user interacts with items within the interest radius, the more the scales will be tipped toward P_R to emphasise proximity, and vice versa. Furthermore, the proximity score P_i and the interaction score I_i are normalised between the maximum and minimum score among the items within the proximity score. Given the volatile nature of user interests (Wu et al., 2008), a time decay function based upon the days elapsed since the last interaction with the item i is also applied on $|I_i|$ and $|P_i|$.

When a certain condition occurs in the user context and profile (e.g. user moves to a new area), the *Personalisation and Visualisation Service* triggers an adaptive action. For example, when the user either starts to explore a new area or alters the user profile by interacting with the map, the client requests new *recommended items*. The items having the highest interest score at a given time for a given location are sent to the map.

4 PROTOTYPE

After analysing the requirements for the architecture described in section 3.1, a survey of the available open-source technologies has been carried out to identify a suitable set with which a prototype can be developed. The open-source technologies which have been selected to implement a system prototype are depicted in section 4.1. Section 4.2 then describes a typical session in which a user interacts with the personalised map.

4.1 Technologies

This section describes the main open-source technologies that have been chosen for the implementation of a prototype of the system. This survey has mostly focused on software packages released under a GPL-like license, adopted by active online communities, supporting standard formats and reasonably stable and reliable. The areas surveyed are related to spatial/GIS services, spatial DBMS, spatial object-relational mapping and agile web development.

MapServer (http://mapserver.org) is an open-source geo-spatial web mapping and rendering tool which complies with OGC standards such as Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS) and Geography Markup Language (GML). Developed at the University of Minnesota with help from NASA, it has built-in support for various vector and raster formats. The core functionality of Mapserver is its MapFile, a configuration file defining the raster and vector layers along with their visual styling. In the system proposed in this paper it is possible to tailor a MapFile for a specific user, updating visual style and structure of the map dynamically. The conceptual simplicity of the MapFile is one of the main advantages of MapServer over similar systems. A significant feature of the MapFile is also the direct connection and support for many DBMS such as PostGIS and Oracle Spatial. An extensive description of MapServer can be found in Kropla (2005).

Within the architecture proposed in this paper, MapServer implements the *Map Renderer*.

Grails (http://grails.org) is an open-source web application framework based on the design paradigm *Convention over Configuration* (Richardson, 2009). Grails embraces agile methodologies and naturally complements Java application development, exploiting the features of Groovy, an open-source dynamic language for the Java Virtual Machine. Groovy provides features offered by other dynamic languages such as Ruby, Python or Smalltalk (Koenig et al., 2007). In its default configuration, Grails relies on Hibernate for the object-relational mapping. Grails offers a seamless integration with *Ajax* (Asynchronous JAvascript + Xml), one of the main technologies whose adoption led to the Web 2.0. Thanks to Ajax, web pages can request data from the server asynchronously in the background without breaking the interaction flow to reload the page. The use of Ajax has led to the development of complex web-based interactive interfaces.

Within the system proposed in this paper, Grails implements the Web Application, with its Web Server Pages and the *Personalisation and Visualisation Service*.

PostGIS (http://postgis.refractions.net) is an open-source plugin that adds support for geographic objects to the PostgreSQL object-relational database. PostGIS constitutes a spatial extension for PostgreSQL and can be used as a back-end spatial database for spatial and geographical applications. PostGIS supports the OpenGIS Simple Features Specification for SQL. It has been developed by Refractions Research as a project in open-source spatial database technology. PostGIS is being increasingly adopted as an alternative to costly commercial products such as Oracle Spatial and Microsoft SQL Server Spatial. For this reasons, several open-source GIS projects support it natively. PostGIS includes user interface tools, topology support, data validation, coordinate transformation and programming APIs. Performances of PostGIS on spatial queries have been compared with MySQL, DB2 and Oracle by Zhou et al. (2009).

In the system proposed in this paper, a PostgreSQL with PostGIS constitutes the Spatial DBMS.

Hibernate Spatial (http://www.hibernatespatial.org) is an extension to Hibernate for handling geographic data, developed by the GIS company GeoVise. Hibernate Spatial provides a standardised, cross-database interface to manipulate geographic data. The *Hibernate Query Language* is extended with geo-spatial data types following the OGC Simple Feature Specification. Optimised SQL code is generated for the underlying spatial DBMS, such as Postgis and Oracle Spatial. Hibernate Spatial has been successfully used by Hespanha et al. (2008).

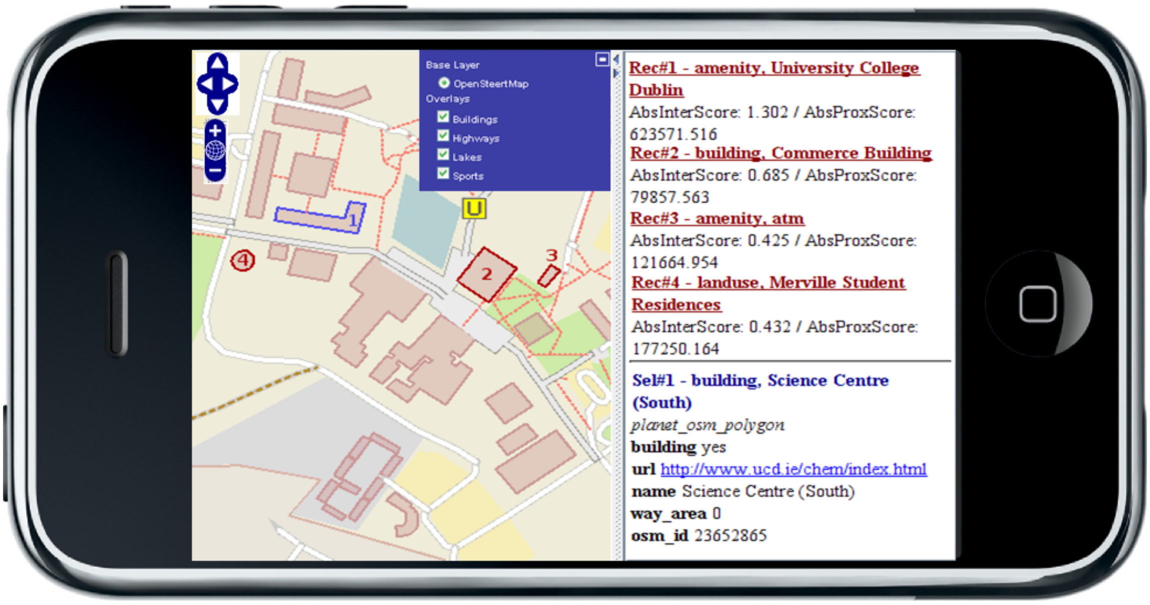
Within the system proposed in this paper, Hibernate Spatial bridges Grails with the Spatial DBMS.

OpenStreetMap (OSM) (http://www.openstreetmap.org) is a collaborative mapping project founded at University College London in 2004. OSM follows the Wikipedia model to create a vector dataset that is free to use, editable and licensed under a new map copyright scheme, similar to the GPL license for software. Although OSM does not have consistent quality assurance procedures, which makes it unsuitable for certain critical applications, its fast-growing free dataset has attracted the attention of the research community over the past 5 years. The success of OSM is largely due to the high cost and the restrictive licensing of most European datasets. The project is described extensively by Haklay and Weber (2008).

Within the system proposed in this paper, a section of the OSM vector map has been stored in the Spatial DBMS.

4.2 User Session

While the system operates with any web-enabled device, the prototype has been implemented for a smartphone. The prototype



Rec#1 - amenity, University College Dublin
AbsInterScore: 1.302 / AbsProxScore: 623571.516
Rec#2 - building, Commerce Building
AbsInterScore: 0.685 / AbsProxScore: 79857.563
Rec#3 - amenity, atm
AbsInterScore: 0.425 / AbsProxScore: 121664.954
Rec#4 - landuse, Merville Student Residences
AbsInterScore: 0.432 / AbsProxScore: 177250.164

Sel#1 - building, Science Centre (South)
planet_osm_polygon
building yes
url http://www.ucd.ie/chem/index.html
name Science Centre (South)
way_area 0
osm_id 23652865

Figure 2: Campus Map on iPhone

consists of an interactive web mapping application to assist users of University College Dublin as they navigate around the campus. The user commences their session by logging into the system, using the smartphone's web browser. The user views the personalised map on the smartphone. This map is personalised and adapted for the user based on their profile built during previous interactions using the personalisation algorithm described in section 3.2. The recommendations are highlighted on the map and detailed information is displayed on a panel to the right. The application, in this case on an Apple iPhone, is shown in Figure 2. A user can zoom in, zoom out, pan and has control over the visibility of layers such as buildings, roads, lakes and amenities. A user can see their position and recommendations on the screen. The recommendations are updated when certain interactions occur. All the user interactions such as map navigation and clicks on items are recorded along with the user location and time so that their profile is continuously updated. A user terminates their session by closing the web browser on the smartphone.

5 DISCUSSION

While there are benefits of the approach described here, it must also be recognised that there are some limitations which must be addressed. This section discusses these issues and how they can be overcome through further development and refinement. Furthermore, details of the evaluation which is required to assess the benefits of the approach are also provided.

The client-server architecture offers many advantages. In particular, it does not require any additional software on the user device. However, there are some limitations with this approach. The client-server architecture has well know bottle-neck issues and it is recognised that beyond a certain number of clients, performance of the server is compromised (Vatsavai et al., 2006). This is potentially troublesome in a geo-spatial application where complex analysis is carried on the server. However, as the power of portable devices improves, there are possibilities to resolve this issue by utilising load-balancing techniques described by Vatsavai et al. (2006), where some computation can be carried out on the user device.

The system described in this paper assumes that the user has a permanent connection to the Internet. This permits a constant stream of information to be exchanged between client and server. However, there are occasions when connection is interrupted or unavailable and contingencies to resolve this limitation are required. One possible solution involves developing client-side caching, whereby events and interactions are stored on the client and transmitted to the server when connection is available. This permits the server to continue user profiling and provide adapted and personalised maps using a complete interaction history. In order to improve map rendering and performance at the interface level, tile caching needs to be implemented on the server-side. Software such as TileCache (http://tilecache.org) works by prefetching map content which is most likely to requested next by the user. This improves performance of the interface and the response time of map interactions.

Now that a stable architecture has been implemented it is possible to use this as a test-bed for future developments. In particular, the algorithm described in section 3.2 can be refined to improve its performance. For example, at present the recommended items are calculated on the interaction history of a single user, however collaborative recommender systems in the web domain benefit from building collective group profiles (Schafer et al., 2007). By defining a user profile similarity function it is possible to cluster users who have similar interests and add new elements to the interest score calculation. The area of group profiling techniques is being surveyed in order to apply them in the geo-spatial domain and extend the algorithm described in this paper. Other techniques such as the introduction of negative scores to map objects is also being explored in the context of this algorithm.

Evaluating the performance of an adaptive LBS is crucial and non-trivial. Currently, a tool to generate user profiles by simulating common usage patterns of the system is being implemented. These profiles represent stereotypical behaviour of certain user categories within the campus prototype that has been developed (e.g. undergraduate students, lecturers, etc.) and will be used as a dataset for future preliminary experiments. These experiments will assist in highlighting both the strong and weak points of the approach and will help refine the design of experiments involving human subjects. Evaluation metrics for recommendations are also being examined. Adomavicius and Tuzhilin (2005) have outlined the advantages and limitations of empirical evaluation through such metrics. These considerations will be taken into account in the next steps of the design and development of the web-based personalisation system.

6 CONCLUSION

This paper presents a web-based system, which uses open-source technologies, to provide personalised and adapted map content to users on portable mobile devices via LBS. A new modular architecture, which comprises of reusable components has been described. The power of the client-server architecture within web-based GIS has been clearly outlined. Using open-source, web technologies, which adhere to OGC standards, removes interoperablily issues which currently exist when developing standalone LBS applications for specific devices. In such cases, the user devices require minimal software to run web based applications. Web technologies also permit the exploitation of existing non-commercial web services within the architecture. For example, external route planning services can easily be incorporated into the design. The system utilises an existing personalisation algorithm to carry out user profiling and map adaptation. The innovative architecture described here provides new opportunities to test this algorithm while also permitting the development of new LBS personalisation techniques.

ACKNOWLEDGEMENTS

Research presented in this paper was funded by a Strategic Research Cluster grant (07/SRC/I1168) by Science Foundation Ireland under the National Development Plan. The authors gratefully acknowledge this support.

References

Adomavicius, G. and Tuzhilin, A., 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering 17(6), pp. 734–749.

Albanese, M., Picariello, A., Sansone, C. and Sansone, L., 2004. Web personalization based on static information and dynamic user behavior. In: Proceedings of the 6th annual ACM international workshop on Web information and data management, ACM, pp. 80–87.

Ballatore, A., McArdle, G., Kelly, C. and Bertolotto, M., 2010. RecoMap: An Interactive and Adaptive Map-Based Recommender. In: SAC 2010: Symposium On Applied Computing (in press), ACM.

Brunato, M. and Battiti, R., 2003. PILGRIM: A location broker and mobility-aware recommendation system. In: Pervasive Computing and Communications, IEEE, pp. 265–272.

Haklay, M. and Weber, P., 2008. OpenStreetMap: user-generated street maps. IEEE Pervasive Computing 7(4), pp. 12–18.

Hespanha, J., van Bennekom-Minnema, J., Van Oosterom, P. and Lemmen, C., 2008. The Model Driven Architecture approach applied to the Land Administration Domain Model version 1.1-with focus on constraints specified in the Object Constraint Language. In: FIG Working Week, pp. 14–19.

Kelly, D. and Teevan, J., 2003. Implicit feedback for inferring user preference: a bibliography. ACM SIGIR Forum 37(2), pp. 18–28.

Koenig, D., Glover, A., King, P., Laforge, G. and Skeet, J., 2007. Groovy in action. Manning Publications.

Kropla, B., 2005. Beginning MapServer: Open Source GIS Development (Expert's Voice in Open Source). Apress Berkely, CA, USA.

Küpper, A., 2005. Location-based Services: Fundamentals and Operation. John Wiley & Sons Inc.

Longley, P., Goodchild, M., Maguire, D. and Rhind, D., 2005. Geographical information systems and science. John Wiley & Sons Inc.

Mac Aoidh, E., Bertolotto, M. and Wilson, D. C., 2007. Analysis of implicit interest indicators for spatial data. In: Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems, ACM, pp. 1–4.

Middleton, S., Alani, H., Shadbolt, N. and De Roure, D., 2002. Exploiting synergy between ontologies and recommender systems. In: 11th International WWW Conference, Semantic Web Workshop, Citeseer, pp. 41–50.

Oliver, E., 2009. A survey of platforms for mobile networks research. ACM SIGMOBILE Mobile Computing and Communications Review 12(4), pp. 56–63.

Oppermann, R., Specht, M. and Jaceniak, I., 1999. Hippie: A nomadic information system. Lecture Notes in Computer Science pp. 330–333.

Richardson, C., 2009. ORM in dynamic languages. Communications of the ACM 52(4), pp. 48–55.

Sanz-Salinas, J. and Montesinos-Lajara, M., 2009. Current Panorama of the FOSS4G Ecosystem. Novática pp. 43–51.

Schafer, J., Frankowski, D., Herlocker, J. and Sen, S., 2007. Collaborative filtering recommender systems. Lecture Notes In Computer Science 4321, pp. 291–324.

Sui, D., 2008. The wikification of GIS and its consequences: Or Angelina Jolie's new tattoo and the future of GIS. Computers, Environment and Urban Systems 32(1), pp. 1–5.

Vatsavai, R., Shekhar, S., Burk, T. and Lime, S., 2006. UMN-MapServer: A high-performance, interoperable, and open source web mapping and geo-spatial analysis system. Lecture Notes in Computer Science 4197(2006), pp. 400–417.

Weakliam, J., Bertolotto, M. and Wilson, D., 2005. Implicit interaction profiling for recommending spatial content. In: GIS '05: Proceedings of the 13th annual ACM international workshop on Geographic information systems, ACM, pp. 285–294.

Wu, D., Zhao, D. and Zhang, X., 2008. An Adaptive User Profile Based on Memory Model. In: Web-Age Information Management., pp. 461–468.

Yang, Y. and Claramunt, C., 2005. A hybrid approach for spatial web personalization. Lecture notes in computer science 3833, pp. 206–221.

Zhou, Z., Zhou, B., Li, W., Griglak, B., Caiseda, C. and Huang, Q., 2009. Evaluating query performance on object-relational spatial databases. In: 2nd IEEE International Conference on Computer Science and Information Technology, pp. 489–492.