A Service-Oriented Platform to Map Cycling Routes

George Henrique Rangel Costa
Department of Computer Science
Santa Catarina State University
Joinville, Brazil
dcc6ghrc@joinville.udesc.br

Fabiano Baldo
Department of Computer Science
Santa Catarina State University
Joinville, Brazil
baldo@joinville.udesc.br

Leandro Lopes Taveira
Department of Computer Science
Santa Catarina State University
Joinville, Brazil
leandrolt@gmail.com

***Abstract*—Over the last few years bicycles have been gaining more attention than motorized vehicles due to the benefits that they offer. Both private and public sectors are joining efforts to stimulate its usage, and an important step in this process is to bring cycling information closer to the population by making it easily available. Computer systems can be used to mitigate such gap by providing information like cycle routes, bicycle stop places and so forth. Considering current technologies it is mandatory that not only personal computers but also smartphones and other mobile devices should be able to access such information. However, to deal with this range of devices it is necessary to use technologies that support interoperability. In this context, a platform using service-oriented architecture and standard communication protocols can wrap a geographic database and a map server to transparently support the presentation of geographic and non-geographic information related to cycling. This paper presents the design and development of such a solution, which aims to increase the availability of cycling information and reduce the coupling between its storage and presentation. To assess this approach, a test scenario is built using real data and two applications are developed – one for Android smartphones and another one for web browsers.**

***Index Terms*—web mapping; web service; web map service; cycling routes.**

I. Introduction

In order to move inside the cities in a faster way people are buying more cars and motorcycles. However, the direct consequences of this attitude are the increase of traffic jam and air pollution [14]. In this scenario, bicycles represent a healthier and environmentally beneficial option. The expansion of cycling routes is an important action to encourage people to use such transport, but this action should not be taken alone. Initiatives to increase people's awareness regarding the existence and location of cycling routes are also necessary and indirectly contribute to increase the number of people using bicycles.

Computer systems can contribute to these initiatives since they facilitate manipulation of maps and related data, i.e., geographic and non-geographic information, respectively. Moreover, this information should be available to various types of electronic devices, such as smartphones and personal computers, in order to disseminate cycling information to all interested people no matter where they may be. Therefore, a problem to be solved is how to facilitate access to cycling information in order to increase availability and reduce the coupling between its storage and presentation.

Several technologies can be used, but none of them are, individually, a complete solution. For example, map servers answer the question of how to provide maps with cycling routes. However, it is also necessary to access non-geographic information, as the route's description and photos of points of interest, which map servers commonly do not support.

Another point that should be considered is the interoperability, which is essential to distribute information to different devices. In the last years, one of the most prominent technologies to enable interoperation is the Service-Oriented Architecture (SOA). One of the greatest advantages of SOA is that it uses standard communication technologies. A platform based on this architecture could answer client applications requests in a standardized way, transparently obtaining geographic information from a map server and non-geographic information from a relational database.

Based on these assumptions, the objective of this work is to design and develop a service-oriented platform that provides cycling information to electronic devices.

The methodology that drives this work begins with a questionnaire applied to groups of cyclists from Joinville-Brazil. This questionnaire aims to identify the difficulties faced by cyclists in accessing cycling information. The analysis of the collected answers is the basis for defining the platform requirements. Then, a literature review of relevant concepts and technologies is done, as well as the analysis of related work. After collecting the needed assets, the platform is designed and implemented. Finally, a test scenario is built and verification and validation tests are proposed in order to evaluate the results of the proposed approach. These tests are executed in an application for smartphones with Android operating system and in an application for web browsers.

The paper is structured as follows. Section 2 presents a brief review of the technologies used to exchange geographic information. Section 3 presents other initiatives for providing geographic and non-geographic data on the web. Section 4 details the development of the service-oriented platform. Section 5 explains how the verification and validation tests were performed. Finally, conclusion and possible future work are drawn.

 ISBN: 978-1-61208-251-6

II. Technologies for Geographic Data Interchange

Geographic Information Systems (GIS) are special types of software devoted to store, analyze, and display geographic information in digital format [11]. There are different types of GIS, each one applied to support a subset of functionalities related to geographic data manipulation. One example of specialized GIS ia a map server. It can be considered a GIS with limited capabilities, because it only provides the means to visualize, on the Internet, geographic information usually stored in geographic databases [12] [16].

Map servers have a client-server architecture where at the client side there is a web mapping application that runs on a web browser. Such client applications use an Application Programming Interface (API) where the users make requests, commonly asynchronous, for images that correspond to parts of a map. The map server receives these requests, generates the images and sends them back to the web mapping application [5] [17].

In order to establish proper communication between client and server it is necessary that both use the same protocol. In this sense, Web Map Service (WMS) is a standard for geographic information retrieval in either vector or graphic formats. Its main operations are GetMap and GetFeatureInfo, which return the map and additional information about a point, respectively. There is also the GetCapabilities operation, which provides meta-data about the service, such as the parameters accepted by the other operations. The WMS operations are called through requests in Uniform Resource Locator (URL) format, which simplifies the adoption of this standard for web mapping applications [18].

III. Initiatives for Providing Cycling Geographic Information on the Web

As the number of people using smartphones and other devices with access to the Global Positioning System (GPS) grows, web mapping applications become increasingly popular and every day new ones are developed. Many of these new applications are devoted to stimulate the usage of bicycles and try to address problems faced by cyclists.

Some applications offer features such as route planning. CycleStreets [4], whose target audience are people that live in England, is an example. Given the desired origin, destination and speed the system provides route options ranging from ”fastest” to ”quietest”, considering shortcuts and lanes exclusive for cyclists. It also shows useful non-geographic information, such as the number of traffic lights along the route. In addition to planning routes, users can view photos and descriptions of points of interest. The application has implementations for web browser and smartphones with Android or iOS operating systems, which facilitates and encourages its use. Developers who want to use the services offered by CycleStreets should use an API that communicates with their server using GET requests [3]. However, the maps must be obtained from an external source, such as OpenStreetMap (OSM) [13].

Another application is called Biketastic [2] and aims to facilitate sharing of cycling routes with cyclists in the region of Los Angeles-USA. Using a smartphone application, cyclists can record their path while riding a bicycle. Besides the geographic information, the application may use the device’s microphone to capture the environment noise level and the accelerometer to identify possible holes or other defects on the path’s surface. Users can add photos and videos of points of interest if they wish. At the end of the ride all this information is sent to a server in order to become available to other users, which can access it using either a smartphone or a web browser [15]. This application does not provide ways for other developers to access its cycling information.

According to the issues presented above, it is clear that CycleStreets and Biketastic contribute to promote the usage of bicycles in the regions where they are offered. Therefore, it is important to develop similar solutions for other regions and countries. However, some points regarding both works should be considered. CycleStreets does not display the points of interest registered by users on top of calculated routes; a feature that could help users choose the desired route. On the other hand, Biketastic sorts the routes by the month of submission; a feature that makes it more difficult to find routes in specific regions.

Also, both works do not provide a search engine to find routes based on characteristics such as surface type, lack of proper route signaling, crossed districts, difficulty and so forth. Moreover, they do not differentiate urban cycling routes from rural trails. For the purpose of this work, the designed platform should provide these features. This work aims to stimulate the usage of bicycle not only for work but also for fun. Therefore, users should be capable of searching for cycling routes inside the city, as well as for long distance trails to enjoy the countryside.

Finally, only CycleStreets provides an API for developers. However, it does not provide access to all information needed to build a client application, like the maps. This shortcoming forces that every developed client must connect to another source of data. In order to avoid this, the proposed platform should provide all information that a client application needs to display cycling routes, including the maps.

IV. Approach for Providing Cycling Information

Besides being an alternative to motorized vehicles, bicycles can also be used for purposes such as have fun, get fit, competitions and so on. For each purpose, different geographic and non-geographic cycling information become useful and all this information should be offered by the proposed platform, such as the route’s name, length, surface type, districts that it crosses, photos of points of interest and so on.

On the other hand, one inherent requirement of this work is that various electronic devices should be able to access the data. Recently, smartphones have gained a lot of attention. These devices support different wireless technologies for Internet access. However, the cost to connect to the Internet using 2G and 3G networks and the connectivity problems outside

 ISBN: 978-1-61208-251-6

the cities' downtown are constraints to be overcome in order to provide a platform with constant and complete access.

Due to these issues, the first step is to understand what are the users' needs. To do so, a questionnaire was built and put available on the Internet (the online version of the questionnaire is no longer available, but the questions are listed, in Portuguese, in [6]). This questionnaire comprises questions related to data and connection availability, as well as questions about geographic and non-geographic information needed by the several kinds of users and about pervasive computing (useful on smartphone access).

Approximately 300 cyclists were invited to participate and 84 have filled out the questionnaire. By analyzing their answers it was possible to confirm their interest in accessing the platform through smartphones, especially with those devices that use Android (preferred by 39.29% of the cyclists) and iOS operating system (preferred by 38.10% of the cyclists). Despite both operating systems being almost equally requested, Android allows applications to be tested on real devices for free, and the SDK is available for Windows, Linux and MAC OS X [9]. Meanwhile, iOS requires a paid subscription to allow applications to be tested on real devices, and the SDK is available only for MAC OS X [1]. Android is also the most used operating system, with more than 60% of the market share [7]. For these reasons, only the application for Android operating system has been implemented.

The questionnaire also shows users' interest on using the sensors present in today's smartphones to show their current speed and location. Furthermore, 96.43% of the cyclists desire to record their paths, and 95.24% would like to send the recorded information to be stored in the platform in order to enrich its database.

Internet connectivity seems to be an obstacle since 97.65% of the cyclists are interested in copying the information provided by the platform to their own devices in order to avoid constant Internet connection. Only 25.24% of the cyclists want to have offline access to the photos of points of interest, so this functionality will not be implemented.

Considering all the questionnaire results it was possible to realize that the cyclists want to use their mobile devices mainly to access data about the countryside near the city where they live, so as to learn the routes and ride for fun. Meanwhile, as observed in the analysis of related work, the platform should allow developers of client applications to implement functionalities such as search for routes according to a set of predefined characteristics and differentiate between urban and rural routes. Also, it should provide all information needed for a client application to display cycling routes in a map.

Based on the literature review and on the questionnaire assessment now it is possible to define the system requirements.

A. Requirements

Regarding functional requirements, first the platform should allow client applications to search routes according to characteristics such its type (rural or urban), surface type or lack of proper signage. Also, when a user selects a route its geographic and non-geographic information must be provided to the client application, as well as all photos associated with its points of interests. Finally, the platform should enable the client applications to copy all information regarding a specific route (except the photos, as explained in the previous section) to the electronic device. Doing this, the client device does not need to maintain any external connection.

Regarding the platform's non-functional requirements, it should be accessed either via personal computers – for example, by a web mapping application – or via other types of electronic devices. Also, it should provide all information through a single interface. Finally, the platform must use standardized communication technologies.

These functional and non-functional requirements are the basis for the specification of 11 use cases that guide the project of all elements that comprise the platform.

B. Service Interface

First, it is important to mention that the platform was designed using the service-oriented approach. Therefore, the requirements were translated into service operations, including their input and output parameters. This approach was used in order to provide clients with a single facade to access all information needed to display maps and cycling routes. So, the only way that a client application can access the platform is through its interface provided as a web service. The platform's web service interface is shown in Figure 1. Among the 13 operations listed, those that support displaying geographic and non-geographic information are *getMap*, *getOSMMap*, *getImage*, *getNonGeographic* and *getFeatureInfo*.

The *getMap* operation returns an image containing the routes and points of interest that exist in a given geographic area. However, these routes should be displayed on top of a map or satellite photo. This data can be obtained from external servers such as OpenStreetMap and Google Maps. In order to simplify the clients' implementation the platform transparently provides maps to the clients using the OpenStreetMap, since its license of use allows this [19]. This feature is implemented through the *getOSMMap* operation. When a client make a call it receives a single image containing the OpenStreetMap's map with the routes already overlaid. Client applications are still free to implement their own access to other external map servers. However, they should use the *getMap* operation and manually overlay the images. It should be noted that the images from Google Maps are not provided through the platform as with OpenStreetMap because its license of use does not allow it [10].

The non-geographic information is obtained using operations like *getImage* – that returns any image related to a route or point of interest – and *getNonGeographic* – that returns all relevant non-geographic information regarding a route, such as difficulty level and description. Finally, the *getFeatureInfo* operation outputs the information about which cycling routes and points of interest exist in a given geographic coordinate.

In order to support routes search, a set of operations whose name starts with "get" were designed, such as *getDistricts* and

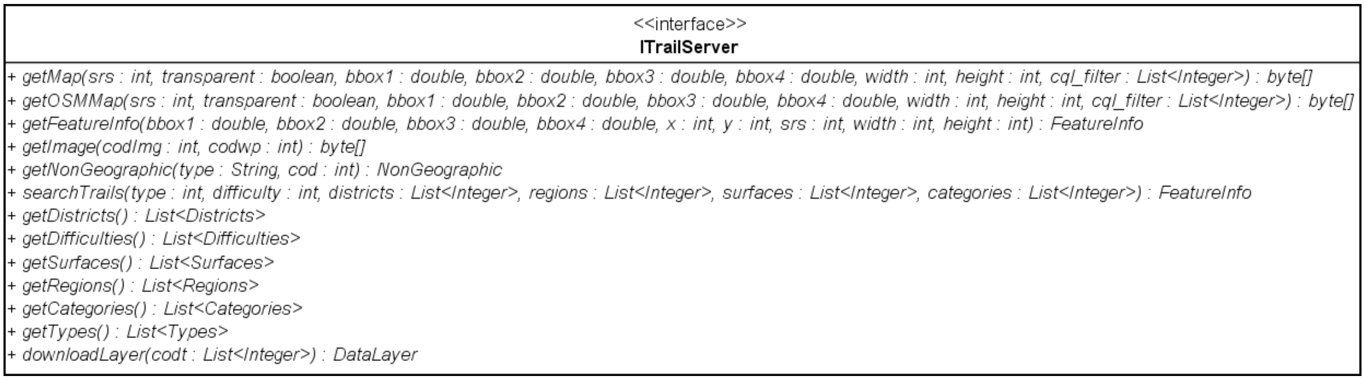

Fig. 1. Web Service Interface.

getTypes. These "get" operations return lists of values that are used to fill in lists of search options. Based on these lists the client application can then call the operation *searchTrails*, which returns a list with all routes that fit the selected search options.

Lastly, the operation that supports the requirement of copying a route to the mobile device is called *downloadLayer*. It returns a list with all geographic and non-geographic information related to the selected route, including its points of interest. This data can then be locally stored in a database at the electronic device, and can be displayed even without Internet connection.

C. Service Architecture

The operations listed in the web service interface are implemented in a three-layer architecture – business, data and persistence – which can be seen in Figure 2. This is a reduced version of the entire class diagram that highlights where each operation presented in the interface is implemented, as well as important classes and packages. It is important to note that the presentation layer is client dependent, so it is out of the platform's scope. Client applications are represented in the diagram by the *Client* package.

The web service interface shown in Figure 1 is placed in the business layer. Although this layer contains few classes, it processes all the client requests. The access to OpenStreetMap (represented in the diagram by the *OSM* package) using the *getOSMMap* operation is also implemented on it. As explained in the previous section, client applications can connect to other services – such as Google Maps – by themselves, and this is represented in the diagram by the *GMaps* package, connected to *Client*.

The data layer implements all the data objects used by the platform, as well as all operations that manipulate cycling information. The persistence layer manages the database access. Finally, the database reflects all these classes and their relationships in a relational schema.

D. Implementation

The platform shown in Figure 2 was implemented using the Java programming language in the Netbeans Integrated Development Environment (IDE). The cycling data was stored in the PostgreSQL Database System, which supports geographic data through the extension PostGIS. The JPA library was used to persist data in the database. GeoServer was chosen as the map server since it supports PostgresSQL and implements the WMS standard [8]. Both GeoServer and the platform must run in an application server and Tomcat was chosen for that.

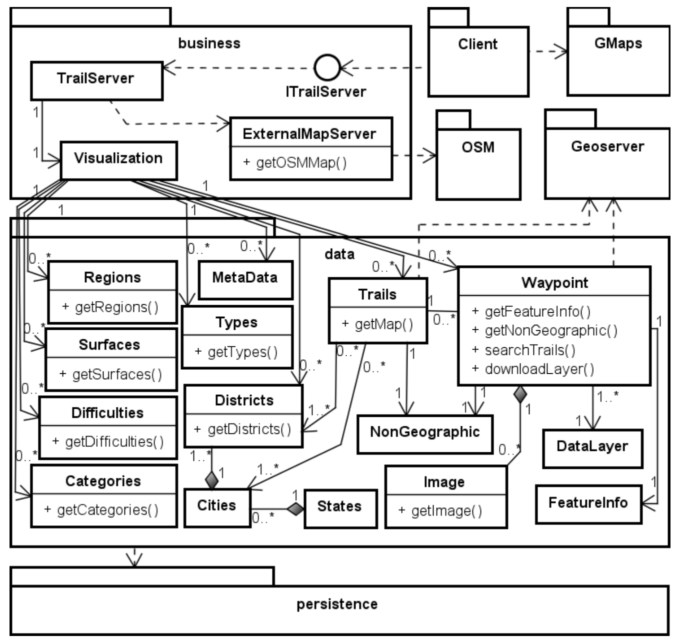

Fig. 2. Platform's Structural Project.

Regarding the implementation, some aspects should be considered. The output of the *getMap*, *getOSMMap* and *getImage* operations are images. However, in SOAP all communications are made through envelopes written in XML format. The SOAP standard has an extension to binary data that uses additional structures to represent the image, but this makes the message even bigger [21]. To reduce the amount of data sent over the network for each request, the adopted solution was to convert the images into byte arrays using Base64. This array is then written in the XML envelope like any other textual data.

 ISBN: 978-1-61208-251-6

Nevertheless, this part of the message needs to be decoded by the client application.

It is also important to mention that both *getMap* and *getFeatureInfo* operations pass clients' requests to the map server as WMS requests. Although it is possible to enable clients to direct access the map server through WMS operations, here it is not allowed because this platform intends to provide a single and centralized interface where client applications can have access to all needed data. This approach facilitates the development of client applications because it hides unnecessary technical issues.

V. RESULT ASSESSMENT

The methodology that guides the platform's evaluation is composed of verification and validation tests that cover all operations. The verification test used the black box approach, where the requirements compliance is verified by comparing the expected output with the real output. On the other hand, the validation test intended to demonstrate that the platform fulfills its users' expectations, i.e., that it works as expected in everyday situations.

To apply this methodology a test scenario was built using information collected by a group of students involved in the project. They mapped, using GPS devices, a set of urban and rural cycling routes from the city of Joinville. In total, 56 routes and 271 points of interest were recorded. At the end of the collecting process all geographic and non-geographic information was stored in the database designed for this project.

For the verification test a tool called SoapUI was used to create SOAP envelopes and send them to the platform, as well as analyze the SOAP envelopes received back. In each of the 14 test cases the contents of the received envelopes were compared to the information stored in the database and everything was 100% compatible.

For the validation test it was necessary to ensure that different types of devices are capable to access the platform. Thus, two client applications were developed, one for smartphones with the Android operating system and another one for web browsers. The next sections describe the development of each client application, as well as present their test cases.

A. Smartphone Application

The decision of using the Android operating system is based on the questionnaire results, as explained in Section 4. The questionnaire also justifies the platform's requirement to allow the copy of a route to the client device. As a way to test if the platform correctly implemented this non-functional requirement – the *downloadLayer* operation – the smartphone application was divided into two working modes: online and offline. Both modes have access to the same functionalities. The only difference is in the source of the cycling information. In online mode all data is obtained from the platform, requiring an active Internet connection. In offline mode only the information already stored in the smartphone is used, not requiring an active Internet connection. Users can use

Fig. 3. Smartphone application accessing the platform.

the application's menu to switch the working mode whenever they want. Note that before viewing a route in offline mode the users must switch to online mode and use the copying functionality, which will connect to the platform and request all data related to the selected route. This data will be then saved in the smartphone's database and can be visualized when in offline mode. Also note that these two working modes were implemented only in the smartphone application to serve as an example of how the *downloadLayer* operation could be used by a client.

A total of 30 test cases have been created for the validation test. Figure 3 presents one of these test cases where the client application is in online mode. In Figure 3.a the output from operation *getMap* is shown. When the user touches the map with a finger the application calls the *getFeatureInfo* operation and the balloon in Figure 3.a indicates that this operation is being executed. Figure 3.b shows the result of the *getFeatureInfo* operation. When the user selects one of those routes its non-geographic information is requested using operations such as *getNonGeographic*, *getType* and *getDistrict*, which are presented in Figure 3.c.

This example demonstrates the use of 77% of the operations defined in the platform's interface. Due to space restrictions the other test cases are not presented here, but their results show that all information – including both text and images (binary data) – is properly transferred to the device and displayed on its screen.

B. Web Browser Application

The web browser client was developed in order to enable users to access the platform through personal computers. However, since personal computers are usually used at home, many features – especially the pervasive ones – would not make sense if implemented, such as show the users' coordinates or alert them about the proximity of points of interest. This client has been implemented as a web mapping application and its purpose is to take advantage from personal computers' screen size and thus better present cycling routes on top of maps or satellite images. Its features include search for cycling routes based on their characteristics and display information about routes and points of interest. This client uses HTML and

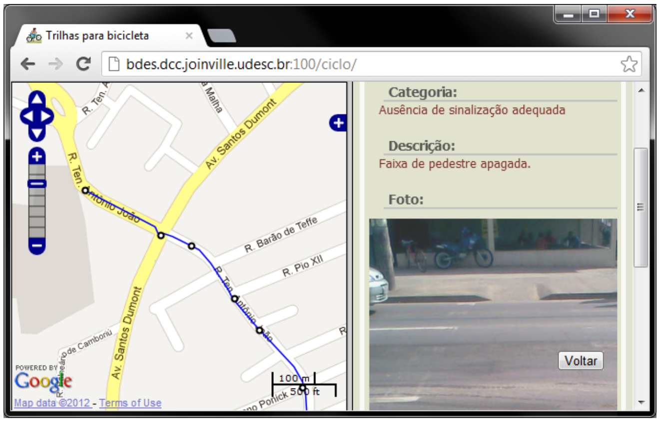

Fig. 4. Web browser accessing the platform.

PHP languages, and the map manipulation is controlled by the OpenLayers library. It can be accessed at [20], and Figure 4 exemplifies its use.

For the validation test 12 test cases were created. The different number of test cases when compared to the smartphone application is explained by the features that could not be implemented in the browser. In Figure 4 the non-geographic information about a point of interest is presented on the right side and the geographic information on the left. The test case in Figure 4 illustrates some operations not used in the smartphone test case shown in Figure 3, such as *getImage* and *getCategory*.

VI. Conclusion and Future Work

Bicycles are healthier than motorized vehicles and therefore its use should be encouraged. Initiatives like Biketastic and CycleStreets contribute in this direction, but limitations were identified in both works. In order to solve them, this paper presented a service-oriented approach to disseminate information about cycling routes in a device independent way.

The verification test cases have shown that the platform complies with the requirements and that it sends the correct information to clients. Besides that, the validation test cases have shown that client applications – both web browsers and mobile devices – are able to interpret and make use of the received data. Therefore, it is assumed that this work facilitates the access to cycling information and reduces the coupling between storage and presentation.

Regarding future work, it is suggested to implement features for adding and editing data allowing cyclists to actively contribute to the growth of the database. Also, it could be interesting to study how to convert the collected cycling routes into a routable graph and how to efficiently identify, with minimum user intervention, points of interest and non-geographic information.

Acknowledgment

The authors would like to thank CAPES Brazilian Research Agency for partially funding this work.

References

[1] Apple, "iOS Developer Program," "https://developer.apple.com/programs/ios/ [accessed: 2012-12-13]."
[2] Biketastic, "BikeTastic," "http://biketastic.com [accessed: 2012-12-12]."
[3] CycleStreets, "Cycle routing API," "http://www.cyclestreets.net/api/ [accessed: 2012-10-22]."
[4] CycleStreets, "UK-wide cycle journey planner and photomap," "http://www.cyclestreets.net [accessed: 2012-10-22]."
[5] G. C. Parma, "Mapas cadastrais na internet: servidores de mapas," Proc. Brazilian Remote Sensing Symp. (SBSR 07), INPE, Apr. 2007, pp. 1311-1319.
[6] G. H. R. Costa, "Acesso a informações cicloviárias em dispositivos móveis: uma implementação para smartphones," Graduation Final Year Project. Dept. Comput. Sci., Santa Catarina State Univ. (UDESC), Joinville, Brazil, 2011.
[7] Gartner, "Gartner Says worldwide sales of mobile phones declined 2.3 percent in second quarter of 2012," "http://www.gartner.com/it/page.jsp?id=2120015 [accessed 2012-12-15]."
[8] GeoServer, "Features," "http://geoserver.org/display/GEOS/Features [accessed: 2012-10-22]."
[9] Google, "Download the Android SDK," "http://developer.android.com/sdk [accessed: 2012-07-25]."
[10] Google, "Google Maps/Google Earth APIs terms of service," "http://developers.google.com/maps/terms#section_10_2 [accessed: 2012-10-22]."
[11] H. D. Parker, "The unique qualities of a geographic information system: a commentary," Photogrammetric Engineering and Remote Sensing, ASPRS, Nov. 1988, vol. 54, pp. 1547-1549.
[12] J. I. Miranda and K. X. S. Souza, "Como publicar mapas na web," Proc. Brazilian Remote Sensing Symp. (SBSR 03), INPE, Apr. 2003, pp. 349-355.
[13] K. Curran; J. Crumlish; G. Fisher. "OpenStreetMap," Int. Journal of Interactive Communication Systems and Technologies (IJICST), 2012, vol. 2, no. 1, pp. 69-78.
[14] Ministry of Environment. Dept. of Climate Changes and Environmental Quality. Brazil. "1^o inventário nacional de emissões atmosféricas por veículos automotores rodoviários: relatório final," "http://www.antt.gov.br/index.php/content/view/5632/1__Inventario_Nacional_de_Emissoes_Atmosfericas.html [accessed: 2012-10-22]."
[15] S. Reddy et.al., "Biketastic: sensing and mapping for better biking," Proc. Int. Conference on Human Factors in Computing Systems (CHI 10), ACM Press, Apr. 2010, pp. 1817-1820.
[16] S. T. Siddiqui. "Software Tools Required to Develop GIS Applications: An Overview," Proc. Int. Conference on Advanced Computing and Communication Technologies (ACCT 12), IEEE, Jan. 2012, pp. 51-56.
[17] S. Tu et al. "Design Strategies to Improve Performance of GIS Web Services," Proc. Int. Conference on Information Technology: Coding and Computing (ITCC 04), IEEE, Apr. 2004, vol. 2, pp. 444-448.
[18] Open Geospatial Consortium Inc. "OpenGIS web map server implementation specification," "http://www.opengeospatial.org/standards/wms [accessed: 2012-10-22]."
[19] OpenStreetMap. "Legal FAQ," "http://wiki.openstreetmap.org/wiki/Legal_FAQ [accessed: 2012-10-22]."
[20] UDESC - Santa Catarina State Univ., "Ciclovias em Joinville - SC," "http://bdes.dcc.joinville.udesc.br:100/ciclo [accessed: 2012-12-13]."
[21] World Wide Web Consortium. "SOAP version 1.2 usage scenarios: sending non-XML data," "http://www.w3.org/TR/2003/NOTE-xmlp-scenarios-20030730/#S19 [accessed: 2012-10-22]."