SUPPORTING MOBILE GIS APPLICATIONS
BY GEOSPATIAL WEB SERVICES

Thomas Brinkhoff

Fachhochschule Oldenburg/Ostfriesland/Wilhelmshaven (University of Applied Sciences),
Institute for Applied Photogrammetry and Geoinformatics (IAPG), 26121 Oldenburg, Germany -
thomas.brinkhoff@fh-oow.de

Commission Ⅳ, WG-IV-6

KEY WORDS: Location Based Services, Mobile GIS, Data Management, Interoperability, Disaster Management, Internet GIS, Spatial Data Infrastructure (SDI), Wireless GIS

ABSTRACT:

Spatial data infrastructures (SDI) providing geospatial web services as well as mobile GIS software become more and more popular. Fields like disaster management require the use of SDIs as well as of mobile applications. However, the scarce resources of mobile devices in respect of memory size, computing power, display size and communication bandwidth make an integration difficult and obstruct many desirable applications. In order to solve this problem, a three-tier architecture is proposed that allows an effective use of geospatial web services by mobile clients. The paper presents the main tasks and operations of the middle-tier, the so-called Mobile Data Service (MDS). Exemplary requests illustrate its functionality. Finally, essential implementation issues of the MDS are presented.

1. INTRODUCTION

In the last years, we can observe two important trends: First, several *spatial data infrastructures* (SDI) providing geospatial web services have been built up. Especially, public authorities are offering map services as well as geospatial data services. In Europe, this development is currently pushed by the INSPIRE directive ("Infrastructure for Spatial Information in the European Community", http://www.ec-gis.org/inspire). Second, GIS is becoming mobile: *location-based services* (LBS) and other *mobile GIS applications* are increasingly used. They are performed on mobile devices like TabletPCs, PDAs and smartphones.

One field, where both trends are required to be integrated, is *disaster management*. In this context, one crucial factor is the fast and reliable provision of geospatial information. Especially in the case of large disasters, the aides are often not familiar with geographical aspects of the affected region. For instance, they do not know about the local infrastructure or about the current extent of the disaster. Using paper maps is inadequate because aides need various kinds of map themes and require precise and up-to-date geospatial information. Personnel in the field demand for information displayed by mobile devices. Furthermore, they must be enabled to add and communicate their observations and data from sensors carried along.

In case of a disaster, many different organizations must cooperate: fire departments, authorities for disaster control, the army, other public authorities and utility providers are typical examples for such organizations. For an effective cooperation of such institutions, geospatial data and services must not be delivered in proprietary or organization-specific data formats. Instead, common standards must be observed. Thus, an SDI for disaster management has to provide geospatial information by using OGC, ISO and W3C standards. Such an SDI is currently designed and developed by the cooperative project *„Open Disaster Management with Free GIS"* (OK-GIS) that is funded by the German Federal Ministry for Education and Research (Brinkhoff et al., 2008a; http://www.ok-gis.de).

An SDI should provide data for all types of users and devices. However, mobile devices like PDAs and mobile phones are limited by scarce resources. Besides restricted computing power, other important aspects are a relatively small and slow memory, a small display size, special input devices (e.g., a stylus must be supported instead of a mouse or a keyboard), the need to save energy, and varying and often low transmission bandwidths. Therefore, standard web services like the Web Map Service (WMS), the Web Feature Service (WFS) and the Web Coverage Service (WCS) cannot always be used directly by mobile devices. Instead, an adaptation of the services is required. This adaptation concerns (among others) the usage of the services (for example the protocol), the selection and adjustment of parameter values, and the data formats.

Furthermore, a large variety of device categories, platforms, and operating systems exists for mobile devices. Therefore, not all devices can use the web services in the same way. A management component is required that configures the clients in an adequate manner.

In this paper, a solution for integrating mobile devices into an SDI is proposed that covers the before-mentioned problems and requirements. In the next section, the overall architecture is introduced. The main components of this architecture are presented in more detail in the third section. Then, some implementation issues are discussed. The paper concludes with a summary and an outlook to future work.

2. THE ARCHITECTURE

This section gives an overview of our architecture for supporting the use of geospatial web services by mobile GIS applications.

2.1 Requirements

The main requirements for the design of such architecture are:

- Several types of mobile devices with different capabilities should be supported (e.g., TabletPC notebooks, PDAs and smartphones).
- The use of services should be simplified as much as possible (e.g., by providing the HTTP Get protocol instead of the HTTP Post protocol) and should be as similar to the original usage as possible (e.g., by using same parameter names and types).
- The data formats of the transmitted documents should be simply processible for a specific device (e.g., by using a graphic format that can be rendered instead of the Geography Markup Language GML).
- The content of the resulting documents should be adapted to the requirements of a specific device (e.g., to its screen size).
- Transactions that change or add data should be supported by mapping these modifications to the corresponding (WFS) transaction requests.

2.2 Overview

The resulting architecture for integrating mobile devices is three-tier like a similar approach for the support of heterogeneous mobile devices in (Brisboa et al., 2007): In addition to the geospatial web services and the clients, we introduce a *Mobile Data Service (MDS)* as middle-tier. Its functionality is generally applicable by mobile devices (and also by desktop computers). More application-specific tasks are bundled in an accompanying layer, which is based on the MDS and extends or adapts its functionality. In case of disaster management, we call this extension *Operational Control Service (OCS).*

In order to provide adequate results, the MDS must identify the requesting mobile device. This is achieved by a device identifier that is transmitted with each service request. This device ID is used (a) for determining the type and the properties of the device (its so-called *profile*) in a device database and (b) for logging the state of the device. The latter information is required for synchronization purposes and for providing data in several steps (e.g., if the mobile device moves and the required map depends on its current position). Thus, a *virtual client* reflecting the current state of an individual mobile client is instantiated when the client first registers with the MDS and exists until the client logs out of the system.

The selection of services and data sources is an important task. In an SDI, this step can be done by the client using services according to the *OGC Catalogue Services* specification (CSW) (OGC, 2007). In order to relieve clients from this task, the MDS provides *context documents* that inform clients about the relevant services and their adequate use.

Figure 1 gives an overview of the three-tier architecture, its main components and the essential data flows.

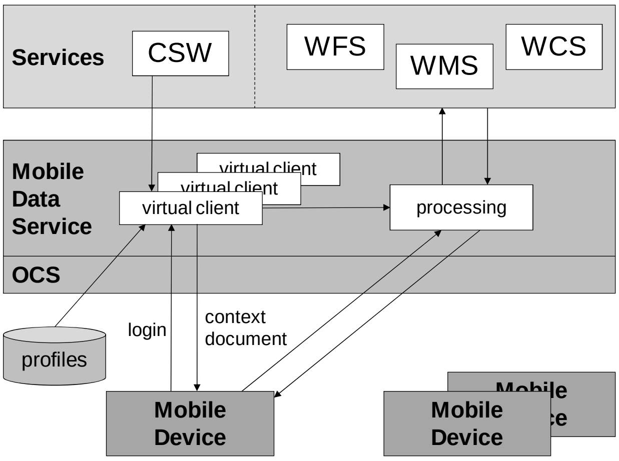

Figure 1. Overview of the proposed architecture

3. MAIN COMPONENTS

In this section, the most important components and tasks of the MDS are presented.

3.1 Geoapplication Context

As mentioned in the previous section, one important requirement of the MDS is to provide a context document with sufficient information about required or recommended web services. For this purpose, we introduced a so-called *Geoapplication Context (GAC).* The GAC configures a mobile geospatial application for a specific mobile client. The *OGC Web Map Context* (OGC, 2005b) has a similar, but more restricted objective, namely to configure WMS datasets. The GAC has also some similarities to the context profile proposed by Predic et al. (2006).

A GAC provides information about map services and geospatial data services relevant for the current application, about corresponding data sources, and about suitable processing services. Furthermore, forms are specified that allow the input and/or modification of alphanumerical data. Additionally, input and/or modification operations of geometrical primitives are declared.

The MDS computes the GAC dependent on the device type, the role of a user, the current situation (e.g., the current disaster and/or location), and the data already available on the mobile device.

A GAC consists of map layers and corresponding alphanumerical data sources. The separation between the graphical layer and the data source (in contrast to GML that allows mixing both) is introduced because of the limited capabilities of mobile devices: They well support the rendering of graphical file documents as well as the access to alphanumerical data stored in local databases. A mixture, however, would require the rendering of graphical data stored in databases or the retrieval of accompanying attributes from relatively large data files (i.e. in case XML data files like SVG, the evaluation of XPath expressions). Both operations cannot efficiently be performed on mobile devices.

Figure 2 shows a fragment of a GAC. Besides obvious attributes like `id` and `title`, a `Layer` element may contain various attributes controlling the display and interactive behavior. The attributes `minScale` and `maxScale`, e.g., controls the visibility of layers for a level-of-detail mechanism like the corresponding parameters in the OGC Styled Layer Descriptor (SLD) specification (OGC, 2005a). In case of mobile devices, however, we cannot expect that all layers are available on the device. Therefore, it is reasonable to define a tolerance buffer where the client is free to decide whether to request a new layer or to depict the current layer with some acceptable decrease of quality. The attributes `minScaleHint` and `maxScaleHint` define this tolerance range.

```
<Layer  id="LB"  title="roads"  pickable="0"
   minScaleHint="0.1"  maxScaleHint="0.5"
   minScale="0.2"  maxScale="0.4"
   dataSource="roaddb"
   ...
/>
...
<DataSource  name="roaddb"
   format = "sqlite"
   src = "http://<MDS>/GetData..."
   modifyForm = "modifyFeature.html"
   newForm = "newFeature.html"
/>
```

Figure 2. Example of a `Layer` element with an associated data source

The `dataSource` attribute of a layer establishes the relation to the corresponding alphanumeric data by referring to the `name` attribute of a `DataSource` element. The optional attributes `modifyForm` and `newForm` refer to HTML pages for editing the alphanumeric data properties of existing and newly defined features respectively.

3.1.1 Tiling: For mobile devices with only small memory capabilities, a tiling of maps is important. Therefore, the GAC offers also information about a suitable tessellation of a (raster or vector) layer into tiles. Tiles of a layer can be loaded if necessary (e.g., after changing the position or the zoom level) and can be unloaded for reducing memory consumption. Figure 3 gives an example of a layer tessellated into rectangular tiles.

```
<Layer title="roads"
       minScale="0.2" maxScale="0.4">
    <Tile src="..." x="20" y=" 0"
          width="20" height="20"/>
    <Tile src="..." x="20" y="20"
          width="20" height="20"/>
    ...
</Layer>
```

Figure 3. Example of a tiled layer

3.1.2 Processing Services: Processing services for a mobile client are declared by `Service` elements in the GAC. Examples for such services are OpenLS routing, geocoding and reverse geocoding services (OGC, 2005c). Additionally, there may be special services provided by the MDS, e.g., for synchronization or polling.

Some typical service declarations are depicted in Figure 4. A `type` attribute is used to distinguish the different services. According to the type of service, further attributes are present. The routing service (`type="LS"`) for example refers to a layer (in the `annLayer` attribute) where the calculated routes should be stored.

```
<Service  type = "LS" annLayer="routes"
          src = "http://<server>/rs" />
<Service  type = "GEOCODER" annLayer="routes"
          src = "http://<server>/rs" />
<Service  type = "MDS"  timeInterval="1000"
          src = "http://<server>/MDS" />
```

Figure 4. Example of service declarations

3.2 Data Adaptations

Device types differ in technical capabilities like performance, display resolution and transmission bandwidth. The computing power of mobile devices is restricted. Their display size is small and does not allow the rendering of detailed maps. Furthermore, the speed and throughput of wireless connections are still considerably lower than of traditional network connections. Therefore, the MDS has to adapt the geospatial data according to those restrictions. This can be done by adapting the *parameters* required for calling web services and by *post-processing* the results of those services.

3.2.1 Parameters: An example for the first approach is the use of suitable parameters for the `GetMap` request of a WMS. The objective is to receive raster maps as result that fit to the display size and resolution of the mobile device calling the service. In case of cooperating clients and organizations (like in disaster management), such parameters should not be determined by each user himself. Instead, a centralized solution is desirable that guarantees a certain quality for each of the mobile devices used by the involved institutions.

For determining a suitable parameter set, the database storing the profiles of the mobile devices is evaluated. Furthermore, the current operation, the role of a user and the location of the device may influence the parameters. Among these parameters are the format of the map, its size and resolution as well as its styling (SLD). The resulting parameter set will be represented in the GAC. Figure 5 illustrates the procedure.

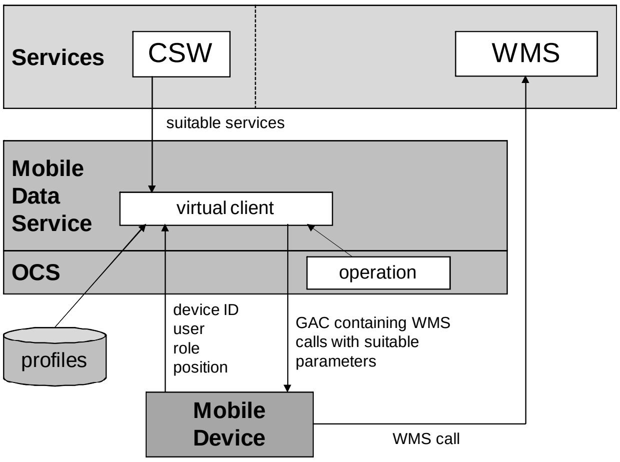

Figure 5. Adaptation by parameter set

3.2.2 Post-Processing: A post-processing will be required if a requested service is not able to produce data that fit to the capabilities of the mobile device. This refers to the syntax of the data (see next section) as well as to semantic properties. In case of the WFS, the result is an XML-document that describes the geometries by using (the simple-feature subset of) GML. A typical problem for mobile applications is that the (unnecessary high) degree of detail of the provided vector geometries does not fit to the display properties of the device. In such a case, a *generalization* of the geometries is required that takes the display properties of the requesting mobile device into account. These display properties are determined by using the profile database again. Figure 6 exemplifies this approach.

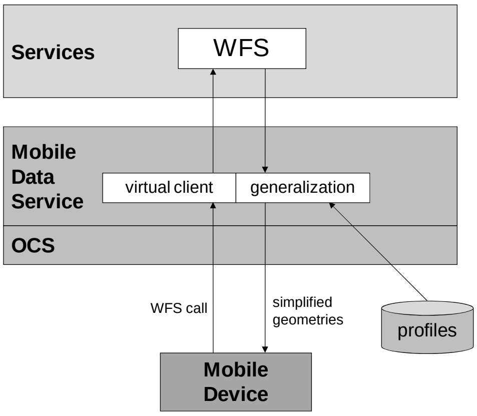

Figure 6. Example for post-processing

3.3 Format Adaptations

The WFS service delivers GML documents with geospatial features and vector geometries but not displayable maps. However, a client requires graphical primitives for displaying purposes. In case of desktop computers, two solutions may be applied: (1) to transform the WFS result into a supported graphic format by the client itself or (2) to use a portrayal WMS that calls the WFS and transforms the XML/GML-document into an image depicting the requested data as map. For mobile devices both solutions are mostly not applicable: For the first approach, the computing power and/or transmission bandwidth are typically too low. The second solution delivers a result that prevents many autonomous operations by the mobile client (e.g., the display of attributes associated to a selected geometry) and increases the need for additional communication between client and server (e.g., by performing WMS `GetFeatureInfo` requests).

Therefore, the MDS performs a post-processing for display purposes. In our current MDS implementation, the W3C standard *Scalable Vector Graphics* (SVG) (W3C, 2003) is provided as graphic format – an approved format for displaying vector maps (Neumann & Winter, 2003). The MDS produces SVG maps by applying device- and operation-specific Styled Layer Descriptors (SLD) to the GML features.

As explained in Section 3.1, a separation of GML into a document for the graphical representation and a database for alphanumerical information is desirable. Therefore, the MDS is able to extract the alphanumeric properties of features from GML and provide them by using common data formats. Among others, *DBase database files* and *SQLite database files* are supported for transmitting the alphanumerical attributes of geospatial features. Especially a SQLite database (http://www.sqlite.org) can be easily stored and handled by a mobile device. In case of data input or data modifications, inverse transformations are performed by the MDS in order to produce corresponding WFS transaction requests.

In some cases, a mobile client needs the graphical representation of the features as well as the alphanumerical database. In other cases, only one representation is required. Therefore, the requests are separated (see also the exemplary requests of the GAC depicted in Figure 2). However, both requests are typically based on the same WFS call. Therefore, the MDS caches the results of WFS `GetFeature` requests in order to avoid duplicate (remote) service calls. Figure 7 illustrates the format adaptation.

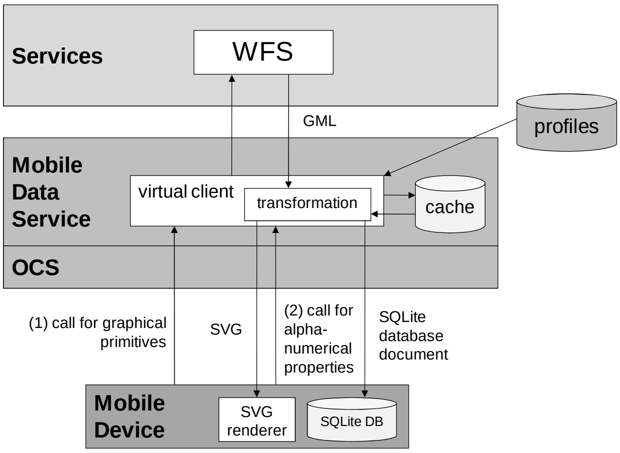

Figure 7. Example for format adaptation

3.4 Offline Support

Devices may be offline for certain periods of time. For instance, a mobile device may be temporarily switched off or the network connection may be interrupted by the user (for saving battery power) or because of technical reasons. Within these periods, data may change on the server side as well as on the client side. It is the task of the MDS to resolve such pending changes. The mobile clients must support offline periods by offering map operations like offline zooming and offline panning. Furthermore, the changes performed must be logged by the mobile client. A more detailed discussion of offline support and synchronization issues is given in (Brinkhoff et al., 2008b).

4. IMPLEMENTATION ISSUES

The MDS is implemented in Java 5. Its operations are offered as Java servlets. The classes are organized in three main packages:

- The `base` package includes all sub-packages that are not specific for the MDS. Among these are packages for the representation of OGC-related specifications like simple features, features, SLD, filter conditions and services. Furthermore, utility classes for the support for SQL, XML, logging and servlets are provided.

- The `mds` package contains all application-independent functionality of the MDS. This covers the representation of requests and their general processing. Sub-packages for registration (with managers for devices, clients, and users), catalogue access, profile management and SLD management are implemented. Finally, a representation of GAC documents is provided.

- The `ocs` package implements all classes that are specific for disaster management. This includes the representation of operations and the computation of actual GACs.

As external libraries are used:

- the JTS Topology Suite (http://sourceforge.net/projects/jts-topo-suite/)
- GeoTools 2 (http://www.geotools.org)
- the Apache Commons HTTP Client (http://hc.apache.org/httpclient-3.x/index.html)
- JDOM (http://www.jdom.org)
- the SQLite JDBC driver (http://www.zentus.com/sqlitejdbc/)

The PostgreSQL DBMS (http://www.postgresql.org) with the PostGIS extension (http://www.postgis.org) is used as spatial database system and Apache Tomcat (http://tomcat.apache.org) as servlet container.

The *OKGIS Viewer* has been developed (http://www.fh-oow.de/institute/iapg/projekte/ok_gis/download.php) as mobile client. It is based on a mobile SVG viewer that originated from a previous project (Brinkhoff & Weitkämper, 2005). The prototype is targeted at Windows CE based operating systems like PocketPC, Windows Mobile, and WinCE.NET. A reasonably good performance was achieved by using C++ as implementation language. Figure 8 gives an overview of its main components. A more detailed description of the OKGIS Viewer is presented in (Brinkhoff et al., 2008b).

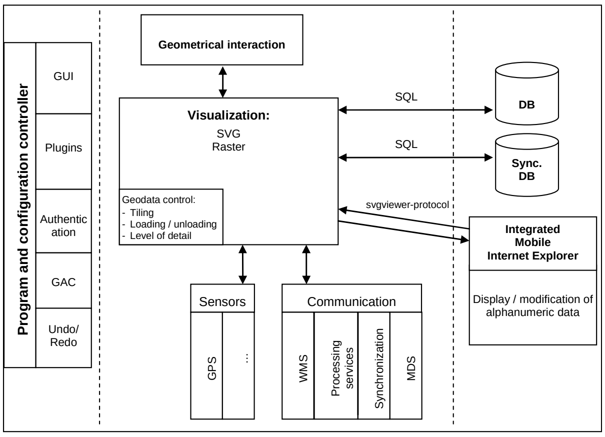

Figure 8. Architecture of the OKGIS Viewer (Brinkhoff et al., 2008b)

In 2008, the complete software developed by the OK-GIS project will be published as free open-source software.

5. CONCLUSIONS

In this paper, an architecture for integrating mobile geospatial applications into an OGC-compliant SDI was proposed. This was achieved by a so-called Mobile Data Service (MDS). The MDS supplies context documents – a so-called Geoapplication Context (GAC) – that inform mobile clients about relevant data services. The GAC also provides information about suitable tessellations of maps and about processing services Furthermore, the MDS generates adequate maps and geospatial datasets by adapting their content and format according to the requirements of mobile devices.

Future work will be the extension of the Mobile Data Service for providing further data formats like OGC KML (OGC, 2008) and for supporting mobile devices that use Google's Android platform (http://code.google.com/android). Another task is the support of OGC SWE (Sensor Web Enabling) services (Botts et al., 2007) by the MDS in order to enable mobile clients to visualize the state of sensors and to control them. Finally, further application areas like tourism should be supported by the MDS.

REFERENCES

Botts, M., G. Percivall, C. Reed and J. Davidson, 2007: *OGC Sensor Web Enablement: Overview And High Level Architecture, Version 3*. OGC White Paper. Document 07-165.

Brinkhoff, T. and J. Weitkämper, 2005. Mobile Viewers based on SVG$^{\pm geo}$ and XFormsGI. In: *Proceedings 8th AGILE International Conference on Geographic Information Science*, Estoril, Portugal, pp. 599-604.

Brinkhoff, T., C. Lindenbeck and J. Weitkämper, 2008b. Interoperable Data Processing by Mobile Geospatial Applications. In: *Proceedings GI Days 2008*, Münster, Germany.

Brinkhoff, T., M. Bertling, J. Biermann, T. Gervens, R. König, D. Kümper, P. Neis, B. Stollberg, C. Rolfs, A. Weiser, J. Weitkämper and A. Zipf, 2008a. Offenes Katastrophen-management mit freiem GIS zur interoperablen Kopplung von Leitstellensystem, mobilen Klienten und GDI mit Prozessierungsdiensten. In: *Proceedings AGIT 2008*, Salzburg, Austria.

Brisboa, N.R., M.R. Luaces, J.R. Parama and J.R. Viqueira, 2007. Managing a Geographic Database from Mobile Devices Through OGC Web Services. In: *Proceedings APWeb/WAIM 2007 International Workshops DBMAN 2007, WebETrends 2007, PAIS 2007 and ASWAN 2007 (Springer Lecture Notes in Computer Science 4537),* Huang Shan, China, pp. 174-179.

Neumann, A. and A. Winter, 2003: Vector-based Web Cartography: Enabler SVG, Version 2.01. *carto:net*, http://www.carto.net/papers/svg/index_e.shtml (accessed 29 April 2008)

OGC, 2005a. OpenGIS Styled Layer Descriptor Profile of the Web Map Service Implementation Specification, Version 1.1.0. Document 05-078r4.

OGC, 2005b. Web Map Context Implementation Specification, Version 1.1. Document 05-005.

OGC, 2005c. OpenGIS Location Service (OpenLS) Implementation Specification: Core Services, Version 1.1. Document 05-016.

OGC, 2007. OpenGIS Catalogue Service Implementation Specification, Version 2.0.2. Document 07-006r1.

OGC, 2008. *OGC KML, Version 2.2.0*. Document 07-147r2.

Predic B., D. Stojanovic and S. Djordjevic-Kajan, 2006. Developing Context Aware Support in Mobile GIS Framework. In: *Proceedings 9th AGILE International Conference on Geographic Information Science*, Visegrád, Hungrary, pp. 90-97.

W3C, 2003. *Scalable Vector Graphics (SVG) 1.1 Specification*. W3C Recommendation 14 January 2003.

ACKNOWLEDGEMENTS

Most of the presented work was developed within the project “OK-GIS: Offenes Katastrophenmanagement mit freiem GIS”. That project is funded by the German Ministry for Education and Research (BMBF). This support is gratefully acknowledged.