Development of a Mobile Mapping Solution for Spatial Data Collection using Open-Source Technologies

C. E. de Abreu Freirea,*, Marco Painhoa

ISEGI, Universidade Nova de Lisboa, 1070-312, Lisboa

Abstract

Information technologies and sport resources aid the potential to transform governmental organizations services, and play an important role in contributing to sustainable communities, respectively. Low-cost tools combining a user-friendly and customized data interaction (forms, mapping, media support) are still in an early stage. Recognizing benefits — efficiency, effectiveness, proximity to citizens — that Mozambican Minister of Youth and Sports (MJD) can achieve employing a low-cost data collection solution, this paper describes the development and implementation of a Geospatial Information System (GIS) mapping application (app) — m-SportGIS — under Open-Source (OS) technologies, customizing an evolutionary and incremental methodology. The app development embraced the combination of mobile web technologies and Application Programming Interfaces (APIs) (e.g. Sencha Touch (ST), Apache Cordova) to deploy a native-to-the-device (Samsung Galaxy Tab 2 running Android OS 4.0) app, taking advantage of device capabilities (e.g. File system, Geolocation, Camera). In addition to the integrated Web Map Service (WMS), a tile caching procedure was carried out to produce a network-communication-independent (offline) Tile Map Service (TMS) regarding IT infrastructures limitations in several Mozambican areas.

m-SportGIS is currently being exploited by Mozambican Government staff to inventory all kind of sport facilities, which feeds a WebGIS platform to manage Mozambican sport resources.

1. Introduction

Governmental organizations produce critical data about the nation’s population, economy, services, and resources. These institutions are coming under increasing pressure, both societal and financial in nature, to develop and implement Information and Communication Technologies (ICTs), inside and surrounding their administrations, supporting a new paradigm of society and modernization, committed to electronic public service [1]. In these last years, a remarkable success in the use of hand-held mobile computers — smartphones, tablet PCs, Personal Digital Assistants, and notebooks — has been noticed for data collection in diverse fields [2]. Data collection is recognized as one of the most
time-consuming, expensive and error-given tasks in any data inventory project. Hand-held computers hold the potential to reduce the logistic burden, cost, and error rate of paper-based methods for data collection [3]; however there is a lack of appropriate, customized and technical low-cost solutions. The notable continuing growth of the Information Systems (ISs) industry creates opportunities and challenges for interesting software apps developments and implementations [4]. In this sense, Geographic Information Systems (GIS) and the electronic public administration services come into a common sphere. GIS data systems easily store and manipulate, combine, and interrelate spatial real world objects (e.g. political boundaries, roads, facilities locations) [5]. Spatial data articulated with other data sources provides efficient means for planning, decision-making, and management many aspects of socio-economic activities which take benefit of a spatial dimension [6, 7].

Mobility is a revolutionary phenomenon and constitutes the most important current trend in Information Technologies (ITs), specially, the category of low-cost solutions [8]. Mobile computing systems and hardware are changing the way mobile mapping technology is being used by moving GIS from the desktop into the user’s hands, providing flexibility in data acquisition, data accuracy and integrity — validation in real-time reducing errors and process costs — more information with much less time and effort, faster communication protocols, and high productivity, making the mobility an enticing aspect of GIS [9, 10].

Tablet PCs integrating location-based services, digital orientation, wireless communications, and sensing technologies, become capable low-cost hardware solutions for geospatial data collection directly into a digital environment, boosting field data collection in a more expeditious and operational way, and improving the data processing and availability, including its transfer and storage in databases, in a range of scientific backgrounds [11, 12, 10, 13, 14, 15]. Mobile web development relies on a collection of technologies that are opening its way into the mainstream of computing, allowing to write web apps — accessed with a browser — that are as visually rich and capable as the so-called native apps [16, 17]. One of the advantages of this recent trend of mobile devices is the availability of feature-rich and fully standards-compliant mobile web browsers (e.g. Chrome, Safari), supporting most of the current standards like HTML5, Cascading Style Sheets (CSS), Javascript (JS), and other standard technologies, such as offline storage, network connectivity, and multimedia [8]. The success of any mobile web app depends on two factors: design and performance. For a better performance, technology should benefit of offline capabilities (storage, caching), animations on the UI, and backend services that retrieve and send data via RESTful or Web Socket endpoints [18]. Every network connection can often experience limitations with online services (backend services), therefore working offline gets into consideration for the mobile development characteristics. Solutions integrating web mapping take more advantage if put up offline map services; this approach is based on caching technique which uses map servers (e.g. MapServer, GeoServer) tools to pre-render tiles in response to user requests [19, 20]. Leverage Web Storage (aka. Local Storage or DOM Storage) to store objects in the background can yield an alternative and better UX (User Experience) — twofold conception: the more that can be done on the client-side, the fewer trips across the network and the better UX provided [21, 22, 17, 23]. Client-side storage relies on data passed to the browser’s storage API, which stores it on the local device, relying on a simple structure of key-value pairs (items) like any JS object [24]. New apps depend upon the data storage in some sort of persistent action in order to be useful, handling it with either a server-side database or HTML5 storage capacity: Web Storage, Web SQL Database, Indexed Database, and File Access [25].

The client-storage security relies on the same-origin principle (Web’s security model) — the store is tied to the origin of the site that creates it (domain/sub-domain), so it cannot be accessed by any other origin. Stores and origins are associated 1 : N, where N is the number of distinct storage mechanisms represented by the current browser/origin [26].

2. Background of Enterprise Software Application Development

Information management is critical to innovation in enterprise environments, and it represents the process of creating, storing and sharing information and knowledge to improve organizational and business performance, expertise and competitive intelligence [28, 29]. Enterprise software development and its quality control assessment are potential expensive depending on the scope of project’s tasks, therefore setting up a proper software development, and testing and maintenance environment is very important to success. Software development can be very complicated without an
appropriate framework, architecture and software model. A useful implementation methodology can benefit software developers to communicate with users/stakeholders, consolidating the design before starting developing, take a better requirements analysis, optimize production costs, speed up the development, and make the code flexible/reusable [30].

Information Systems Development Methodologies (ISDMs) have been encouraged by organizations to structure, plan, and control the process of developing new software [31]. There are a variety of methods variants adopted in IS developments, and software engineers struggle to select an appropriated one to work for all development situations [32]. Nevertheless, they keep a basic structure relying on a feasibility study, system investigation, systems analysis, systems design, implementation, and review and maintenance (Figure 1). Methodologies are step-by-step procedures to carry out the development activities in different phases of a System Development Life-Cycle (SDLC), organizing coherently sets of behavioral and technical conducts, prescribing how to address development problems with the required functionality and quality [33, 34]. Prototyping and agile methodology are based on the idea that planning all requirements and contingencies in all but trivial system developments is impossible; pre-implementation system analysis (specific functionalities required of the full proposed system) may not capture all relevant requirements, and these eventually will emerge when the system is designed in detail and well implemented, conducting to higher project costs [35]. The growing adoption of these methodologies was empowered by tools that speed up the development process, allowing developers to rapidly create pseudo-working models (small-scale versions) of the full proposed system in order to gather stakeholders feedback about the system’s design, functionalities, user-interface (UI), and so on [36]. Figure 2 depicts a broad comparison between the main software developments philosophies here referred.

An Information System Development (ISD) implementation should involve a socio-technical analysis for determining which model and methodology best satisfy in effectiveness the project requirements [37].

3. GIS Mapping: Related Work

Potentials of modern mobile mapping devices have been addressed for electronic geospatial data collection in diverse fields, however it is still in an early stage, and low-cost solutions combining an intuitive customized and rich form with mapping are not yet spread.

Examples of mobile GIS are noticed in biology (e.g. epidemiology, ecology) and environment monitoring (e.g. EpiCollect¹, Magpi²); researchers and public health physicians employed handsets to inventory and manage data such as soil contaminated sites, plant and animal species, being submitted directly into databases or statistical packages through Internet service, which further could be linked to geographic information, performing spatial analysis and planning [13]. The IBM Almaden Research Center has designed an iPhone app — Creek Watch³ — to monitor watersheds; users with the phone’s GPS enabled take a photo and submit crucial pieces of data based on observations [40]. The Center for Embedded Networked Sensing (CENS) at the University of California, Los Angeles, have also developed an Android app to locate invasive species in Santa Monica mountains, by making geo-tagged observations and taking photos, to study the spread of habitat-destroying invasive plants and animals (What’s Invasive) [41].

¹ epicollect.net
² magpi.com/login/auth
³ creekwatch.researchlabs.ibm.com
Computer scientists at the University of Washington have launched an Android app — Ultra Mobile Field GIS — to acquire georeferenced data of public facilities (e.g. bicycle stands and their capacity, car parking lots, and illegal garbage dumps) to assist administrators in maintaining the campus lands cape and managing the student facilities [42]. Docking special sensors to handsets have also been conducted to run environment-centric apps (e.g. Haze Watch4) in Australian urban spaces; these “mobile stations” afford air quality monitoring collecting environmental parameters as geodata related to air quality or noise pollution. Mobile stations collect less accurate measurements than settled stations, but are able to recognize mobile unpredictable events [43]. Field workers in coastal zone management studies in England took spatial data acquisition including media information — digital photograph to provide visual record of the study area — to determine shoreline movements (erosion) over time [11, 44]. Air quality management systems, regarding emissions of air pollutants and greenhouse gases, have been built thanks to air pollution software apps, combining GIS field tools and Internet, by the United States Environmental Protection Agency, and California Environmental Protection Agency, to produce GIS-based emission inventories [5, 45]. Attending to accuracy of raw data collected, mobile GIS is adopted in geological field mapping (information of stratum, rock, and structure) by for geologists and geoscience users (petroleum and mining industries) to provide geological materials for territorial planning, mineral exploration, and engineering geology [46, 12, 15]. In forest management inventory, where conventional data surveying and processing methods need high-intensity labor force since large amounts of data are implicated, forestry science workers paid close attention to mobile mapping combined with remote sensing images, vector maps and grid maps. Mobile GIS supports a modern electronic governmental forestry service — e-forestry — integrating automatic mapping and warehousing [47, 48]. In the context of participatory planning projects (e.g. National Spatial Data Infrastructure (SDI)) US citizens employed handsets to register, mapping, and tracking almost anything that was conceivable with spatial analysis relevance; mobile devices were perceived as “networked mobile personal measurement instruments” [49].

4. Development of m-SportGIS

Regarding the nature of mobile web-based project characteristics, such as evolving requirements, both in size and functionality, evolutionary and incremental software development approaches were considered an affordable solution to the m-SportGIS development; yet, based on [50] considerations, due to governmental organization behavior, it was assumed that a certain amount of method tailoring was required, in order to make use of any methodology, or indeed a combination of several methodologies. As well, [51] identified that the majority of the companies implement hybrid or customized approaches as the most adopted methodology in systems design, followed by traditional SDLC, and Rapid App Development (RAD), or agile development methods.

In order to pursue the deployment of a customized and low-cost mobile GIS solution to the MJD, an app was developed under OS and hybrid (native-to-the-device) technologies, offering a user-friendly layout which integrates a customized data form addressed to the sports arena (150 form fields broken-down into field technician, sport facility, sport activities, technical specifications, and general characteristics information) and a map interface (coordinates and online and offline map service). Technicians input data directly in the client platform, which is validated by models, and stored (primarily) locally (HTML5 Web Storage: LocalStorage). Saved data is suitable to be sent to a central spatial database (server-side) using communication technologies (Asynchronous JavaScript and XML (Ajax)). The spatial database implemented in the server-side communicates with a WebGIS platform — “Atlas Desportivo Nacional de Moçambique”5 — that attempts to visualize, query and analyze data, and to create statistics and reports of sports facilities and services. The WebGIS was performed in a post-processing project, which conducted to the conception of a Sport SDI. The full integrated system is broken-down as shown in Figure 3.

The first large project phase took into account principles of Structured System Analysis and Design Method (SSADM) (a waterfall-based method) methodology [52], contemplating a basis software implementation analysis to set up requirements and plan an IT strategy. This analysis considers the hard and soft skills required to implement the software solution, the type of end-users, costs estimation, and the risk involved within the app development life-cycle. Meetings between stakeholders (Mozambican Government) and software developers and managers (ISEGI) took place to discuss and plan main project guidelines. Further, a technological analysis was conducted, based on outcomes from stakeholders feedback, and current low-cost ICT (mobile and web technologies) available in order to fulfill the

---

4 pollution.ee.unsw.edu.au
5 atlasdesportivo.mjd.gov.mz
objectives effectively (Figure 4). A second important phase contemplated the design of an app model attending to the specifications defined, based on the prototyping model (RAD and agile methods), in order to get low-cost and fast product functionalities regarding the client feedback. It was launched an iterative operation consisting in the construction of a mock-up of the proposed app (model-driven) (using OS Pencil Project), presenting a suitable outlook of the final product with its general functionalities, minimizing risks of incomplete or inadequate issues. The next significant phase was focused on the creation of the full app, driven by the last model-driven solution, using OS resources, based on web and mobile technologies, and web mapping services. An app beta version was built, tested against faults, and refined in the laboratory and then in-situ, considering the infrastructure and environments that users can encounter during its usage (device: Samsung Galaxy Tab 2 7” and 10.1”; Operation system: Android version 4.0). The technical documentation (manual) was also written, and users training were performed in Mozambique. Later, a refined official app version (Figure 5) and its documentation were launched in the Mozambique governmental platform. Currently, a maintenance period is being taken, with users support, technical corrections, updates (e.g. new functionality), and improvements, to commit incremental value to the product.

Fig. 3. Integrated system architecture; Fig. 4. Software used in m-SportGIS production.

4.1. Technologies for the m-SportGIS Deployment

4.1.1. Coding Approach

A mobile device such as an Android tablet PC constitute today a convenient client computing device with mobility and context-sensing capability. Designing for the mobile domain forces developers to embrace those constraints (e.g. small screen sizes, light memory, changes in networking conditions), leavening the creation of innovative mobile experiences supported by new capabilities on the web browsers [27, 53]. The mobile channel leads us questions such as which code approach, i.e. native vs. web vs. hybrid, can be appropriated, which testing platforms, libraries, and resources support have we available to a proper development [8]. The deciding factors are ultimately user’s requirements and expectations from the mobile solution, developer’s capabilities, project specifications, available tools, administrative and security concerns, budget, time-lines, and long-term goals. Native code-based apps run “natively” granting the best access to the capabilities of the device (e.g. Camera, File system) assuring the best UX. Nevertheless, developers cannot reuse the UX from one device when they are building an app for another; they require the skills of specialists in specific mobile vendor SKDs and IDEs to deal with its own conventions which its users expect (e.g. Apple’s iOS demands Objective-C; Google’s Android relies on Java) [54, 55]. The mobile web, from a user’s perspective, falls essentially in web content accessed from a mobile device; from a developer’s perspective, the mobile web it is a group of best practices, design patterns, and new code. [56]. Modern handsets support a WebKit-based rendering machine engine browser (e.g. Chrome, Safari, Opera), supporting HTML5, and adapting the latest W3C specifications [18]. An advanced mobile web app can play a really interesting role for a portable platform. Device manufacturers do the work of carrying on the web platform to their device, ensuring a good UX that is consistent with other web apps on that device. Using JS APIs web developers can today access to mobile native APIs within the browser [55]. Hybrid mobile apps are a type of mobile apps developed using both web and mobile technologies. A Hybrid app is part native app, part web app (aka. native web app). It is effectively a web app, but hosted inside a native-to-the-device app container. It is installed, launched and used like any native app, and can access a variety of device APIs, but it is written with web technologies HTML5, CSS and JS [56, 57].

The m-SportGIS development was proceeded under this approach, taking advantage of a mobile app platform (Apache Cordova API (aka. PhoneGap API), acting as a web-to-native wrapper, providing a JS-based API (cordova-
JS) to interact with native device capabilities (e.g. File system, GPS/Geolocation, Accelerometer, Media/Camera); it works in parallel with a Software Development Kit (SDK) of the target platform (Android SDK). This feature is very important since when the app leads with multimedia capture (through PhoneGap Camera), HTML5 offline storage capacity is limited to 5 MB [58] (enough for alphanumeric data), and digital data requires extra physical storage (device or external memory card); thus, with PhoneGap we can manage data storage and retrieval. Indexing a record to photography — destinationType: Camera.DestinationType.FILE_URI — guarantee that multimedia data physically stored is explicitly called when that record is requested.

4.1.2. MVC Architecture
The Model-View-Controller (MVC) is an architecture pattern and describes a way to organize an app code through a break-down of concerns, i.e. the separation of business data (state of the app) (Models), from user interface/presentation (Views), functionality/business logic handling user actions (Controllers) [59, 60], interacting between then according to user actions. In a client-side web-based app, the MVC architecture is implemented with the HTML and CSS shaping the view, the JS becomes the controller, and the objects (containing the state of the UI) make the model [59]. A typical MVC structure is depicted in 6. Proxies are used by stores to handle the loading and saving of the model data (LocalStorageProxy in client-side); data objects transfers are done based on JavaScript Object Notation (Json) data standard format ({"attribute" : value}). Readers are used to load data into a model or store [61]. Every object-oriented framework or toolkit (e.g. ST, Backbone, AngularJS) has conveyed its own sort of MVC, showing the infinitive flexibility of software, and the adaptability of design patterns to different situations and environments [8]. According to experiences reported by [62, 55, 63, 56], ST and jQuery Mobile (jQM), both free solutions, constitute the most serious client-side framework options to mobile web development, noticing that jQM seeks to work on a broad range of mobile devices, and ST targets the most operative ones — iOS, Android 2.3+, and BlackBerry 6 — holding a robust MVC code base, enhanced support for touch events, and great API documentation.

m-SportGIS was developed using ST (version 2.0.2) JS framework (WebKit-based). Ext JS (version 4) library is the basis of ST. However, ST including several key features, such as a set of Graphical user interfaces (GUIs) components and a particular gesture library, highly optimized for touchscreen devices. ST holds a large UI library widget (design and functionality) inspired by iOS (Apple mobile Operation System), hardware-accelerated engine based on CSS, and complex customizations [64, 8]. Figure 7 shows the Ext.app.Application file which defines the set of Models, Controllers, Profiles, Stores and Views that an m-SportGIS consists of.

4.1.3. Location Services
Location-based services (LBS) and mapping make up very useful categories of modern mobile apps. The mobile device’s geographical location can be gathered using the W3C Geolocation API (or using third-party APIs from carriers (e.g. OneAPI, BlueVia)) [56]. The Geolocation API represents an object able to programmatically obtain position information of a device, since it has external location data (GPS), cell tower triangulation, and/or Wi-Fi data available [67]. Its implementation in a WebView (mobile browser) takes the navigator object in JS and passes us the geolocation property (window.navigator.geolocation), allowing us to interact with that interface through its methods (e.g. getCurrentPosition(), watchPosition()) [68]. This methods, together with OpenLayers API for map visualization, were implemented in functions of the Map Controller of the m-SportGIS, benefiting of the Cordova Geolocation API, allowing location aware on the WMS.
4.1.3.1. Tile Map Service (TMS)

Online WMS provides us dynamic (and updated) services. m-SportGIS presents a Google Maps and an Open Street Maps layer (WMS). Nevertheless, network communication can occasionally not be available, depending on the IT infrastructural environment. In several Mozambique areas there are important network limitations. In order to guarantee that m-SportGIS can always afford a mapping service (statically at least), creating cached tiles (aka. slippy maps) as a parallel map service solution (offline layer) was fundamental. Server side caching for web map services stores and manages map image tiles (square images) produced by map servers (e.g. TileCache, MapTiler, GeoWebCache, MapBox), responding to client requests directly with cached tiles. TileCache is an implementation of a Tile Map Service (TMS) (or Web Mapping Service-Cached (WMS-C) (Figure 8). The caching mechanism is based on file systems — each tile is saved in the cache as a separate operating system file — and depending on layers extensions and number of zoom levels requested, the information needed for a file system to locate and open those files (and the number of correspondent tiles) can be tremendous, and impossible to exist in the device main memory [19]. In order to prepare data to carry out a WMS-cached of Mozambique, a project in Quantum GIS desktop was created to organize and edit data (shapefiles) downloaded from Centro Nacional de Cartografia e Teledetecção (CENACARTA) of Mozambique (e.g. administrative divisions, roads, rivers, points of interest — Spatial Reference System SRS : Spherical Mercator EPSG3857). As outputs, Mapfiles (file.map) were conceived for each layer class. TileCache holds various versions (wsgi, cgi, mod_python) [69], capable to draw “meta-tiles” where a large tile is rendered, and then sliced into smaller tiles using the Python Imaging library [70]. Based on this references, tiles (256x256 pixels) were created under the CGI mode (since configuration procedure is similar to MapServer) running on Apache, setting up the TileCache (version 2.11) configuration file as show in Figure 9, and running a Python mapscript (MapServer) — tilecache_seed.py [options] <layer> [<zoom start> <zoom stop>]. Iterative resolution analysis were undertaken to established the number of zoom levels needed to cover all the country with adequate detail, and in the available disk storage space (10 zoom levels set). Since resulting hierarchical structure of tiles took a very large size on disk, it was necessary store it in a Secure Digital Memory Card (8GB), and a path to display tiles tree was defined in a offlineLayer instance (inside Map controller) (Figure 10).

5. Results
m-SportGIS has been deployed presenting the following main interfaces:

- A main/initial view allowing users to create a new record (“Criar Novo Registo”) or look up to saved records (“Ver Registos”) (Figure 11).
- A data collection view comprising a data form broken-down into five information tabs and a map interface with a geolocation request functionality (Figure 12).
- A data collected view consisting of a list of saved records and a map window showing their location (highlighting a record on the map when it is selected in the list) (Figure 13).
- A data edition view similar to the data collection view, but restricting the geographical data (GPS request) modification (Figure 14).

![Fig. 11. m-SportGIS start view; Fig. 12. m-SportGIS data collection view.](image1)

![Fig. 13. m-SportGIS saved data view; Fig. 14. m-SportGIS data edition view (no GPS).](image2)

### 6. Conclusions

An OS mobile software tool for geospatial data collection has been successfully developed under mobile web technologies, based on a prototyping ISDM. Mobile mapping tools represent very convenient vehicles to overcome the substantial bottleneck of available spatial and attribute data in developing countries. m-SportGIS constitutes a low-cost solution to produce rich data, and is currently available and being exploited by Mozambican Government staff. That brings improvements to the e-government usage, since politicians, sport planners, and citizens are being empowered to regulate and bestow better national sport resources and services. The app holds a very complete and customized form interface, a web mapping service online and offline capable, and a multimedia-record-indexed functionality. Android devices proved to be very suitable (and low-cost) tools for GIS mapping. Current JS APIs also demonstrated to be useful means to rich device (native) capabilities, such as File system, Geolocation, and Camera. Tile caching proved to be a valuable mechanism to improve client experience and build a TMS network-communication-independent.

Being a mobile web-based app, it is a tool able to be adapted and expanded to new functionalities or reformulations. Future work can include research of other OS programming techniques. Regarding the cache mechanism other features and scales and details can be built. A plausible feature extension discerned would be the spatial geometry edition capability to polygons and lines in-situ, requiring also the implementation of a client spatial database capable of receive data from a server database, providing advanced storage and capacity.