Design and Development Guidelines for Real-Time, Geospatial Mobile Applications: Lessons from 'MarineTraffic'

Dimitrios Zissis, Dimitrios Lekkas, Panayiotis Koutsabasis

University of the Aegean, Department of Product and Systems Design Engineering, Syros, Greece, GR-84100, {dzissis, dlek, kgp} @ aegean.gr

Abstract. The development of real-time, geospatial mobile applications poses particular challenges regarding their interaction design and technical implementation. In this paper, we present insights into the design and development of the mobile version of MarineTraffic (marinetraffic.com), which is an open, community-based project that provides real-time geospatial information about vessel movements and port traffic. During the technical development and deployment of MarineTraffic mobile, we identified a number of 8 principles and 25 guidelines that had to be followed in order to improve the user experience and tackle technical issues. We discuss these guidelines with respect to implementation examples and experiences from the specific mobile applications and we suggest that these guidelines have to be kept in mind for the design and development of similar mobile applications.
Keywords. Mobile application design guidelines; mobile interaction design; mobile application development; MarineTraffic; application design;

1 Introduction

Currently an overall of 10% of global website views are from mobile devices (smart phones and tablets). On Christmas day 2012, more than 50% of online activity came from mobile devices [1]. It is interesting though that, according to this study, the vast majority of mobile activity, measured at 98%, was from native applications. Only 2% of mobile activity was conducted through the mobile web. According to the NPD group survey, 37% of consumers who used to access content on their PCs switched to their tablets and smartphones, with the top two activities being web browsing and Facebook [2].

When web sites rest mostly on textual and image content, do not involve many network queries and make limited use of interactive technologies, then their conformance to web usability (e.g. Research-based Web Design and Usability Guidelines [3] and accessibility (Web Content Accessibility Guidelines [4]) guidelines can suffice for a satisfactory mobile user experience. This is not the case for more sophisticated web designs that involve: real-time geospatial information visualization, manipulation and navigation on geographic maps, layered information presentation and user content uploads and ratings. These websites are almost completely inoperable via mobile browsers, as differences in compatibility hugely deter their interactivity often making it impossible to perform even basic map functionality. These incompatibility issues are added on to typical mobile devices' limitations in CPU, memory, network availability and screen size.

The design and development of native mobile applications is complex and challenging [5]. It concerns developing for a number of different operating systems and devices with various characteristics, in a diverse set of programming languages. In such a fragmented field, designers and developers have to comprehend the restrictions of the technology (e.g. small screen, memory, CPU) and the particular technical characteristics of mobile operating systems (e.g. iOS, Android, Windows phone, etc.) [6]. In addition, mobile devices are characterized by a unique synthesis of interaction affordances that can actually transform the user experience when compared to desktop platforms including: gesture-based, multi-touch interaction with digital content; location awareness and subsequent service and content adaptation; advanced sensing capabilities (with embedded devices like: accelerometer, gyroscope, GPS, camera etc.); multimedia (photos, sound, video) capturing and sharing [7–9]. Therefore, mobile application development requires a whole new way of thinking in respect to interaction design and HCI, as well as to software development.

In this paper we present the case of the design and development of the mobile version of MarineTraffic (marinetraffic.com), which is an example of a real-time, geospatial, community-based web service that allows users to view vessel information, positions, routes and port traffic in real-time. Our work and experience with MarineTraffic mobile has yielded a number of respective principles and guidelines (8 principles and 25 guidelines). We discuss these guidelines with respect to how we implemented them in terms of the specific mobile applications and we suggest that these guidelines have to be kept in mind for the design and development of similar mobile applications.

The paper is structured as follows: Section 2 briefly presents the MarineTraffic service, outlining its basic characteristics that have to be transferred to its mobile counterparts. Section 3 presents the design and development of MarineTraffic mobile applications based on a number of principal development goals that gradually transformed into a set of design and development guidelines that were followed throughout the process; in this section we view these guidelines in terms of examples and experiences from the mobile applications developed. Finally, section 4 presents the discussion of our work with respect to related work and outlines its potential contribution for mobile design and development of similar applications.

2 Marine Traffic Web Application

Marine Traffic is part of an open, community-based project that provides real-time information to the public, about ship movements and port traffic, mainly across the coast-lines of many countries around the world. The initial data collection is based on the Automatic Identification System (AIS). As from December 2004, the International Maritime Organization (IMO) requires all vessels over 299GT (Gross Tonnage) to carry an AIS transponder on board, which transmits their position, speed and course, among some other static information, such as vessel's name, dimensions and voyage details. This information is broadcasted through public radio frequencies and is accessible to any interested party with the required equipment. Using an AIS receiver and dedicated software we are able to store this information in a database and then provide it to the general public through marinetraffic.com.

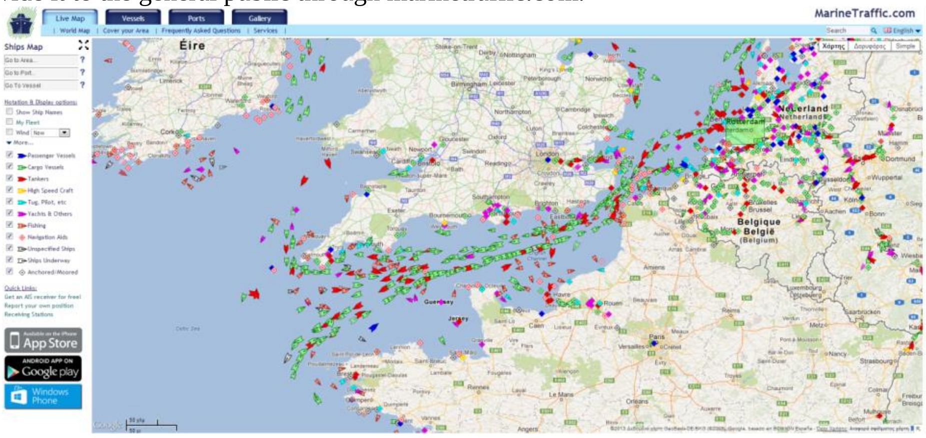

Figure 1: Aspect of Marine Traffic Web Application (http://marinetraffic.com).

The MarineTraffic Web application is an AJAX (Asynchronous JavaScript & XML) map mashup, that overlays vessel and traffic information on an interactive map. MarineTraffic makes use of a base map, a geocoder, an additional layer of presentation information and a Web interface. In general, Web mashups are mostly realized by web pages that leverage script languages such as JavaScript, which enables better user interactivity by locally executed functions and the dynamic loading of data from web services [10].

MarineTraffic provides a number of services including: real time vessel tracking, vessel search, vessel track history, direct navigation to vessel, port or area, distance calculation, vessel photographs, user ratings of these photos and additional vessel related information. MarineTraffic also provides a number of personalized services such as fleet management, vessel related notifications and specific area tracking. MarineTraffic heavily rests on network availability to display and update vessel movements and port traffic. In addition, MarineTraffic is a community-based (or crowd-sourcing) application that also rests on users who can contribute to the service infrastructure by adding nodes of data collection (according to instructions provided from the website), as well as content by uploading and rating ships' photos.

MarineTraffic is a highly popular service, with approximately 14 million visitors per month and 50 million page views. From these, approximately 1.5 million visits originate from mobile browsers (mobile phones and tablets). Before the design and development of MarineTraffic mobile applications, a large number of users had reported that viewing MarineTraffic from specific mobile web browsers offered a degraded experience (actually in many cases the user interface was completely unusable). We have to note that there are many thousands of users that access MarineTraffic at sea via their mobile devices. For example, it is typical user behavior to take photos of vessels at sea and post them to MarineTraffic, or request specific vessel information including its current coordinates, heading, speed, port arrival time and related information.

Poor user experience over mobile devices is mostly due to JavaScript compatibility issues with the AJAX map mashup. Unfortunately, websites containing AJAX map mashups are often dysfunctional when viewed via mobile browsers. Map gestures and interactions are interpreted inconsistently, mostly due to differences in JavaScript implementations)[11][12], buttons are pressed with difficulty, and the user experience is hampered by the narrow layout making it almost impossible to perform even basic map functionality[13]. Many map providers only offer a degraded version of their map for a specific subset of browsers if any. In many occasions maps have been completely unavailable on specific mobile browsers [14][13].

It was soon obvious that native mobile applications of MarineTraffic had to be built, so that mobile users would not be excluded from the service. Although time-consuming and sometimes a repetitive task, the development of native mobile applications not only makes it possible to overcome many of the interaction issues that deteriorate the user experience through the mobile web browser, but provides the development teams with the ability to program at a lower level, thus increasing application execution speed and performance, while gaining access to a wide array of device specific functionality.

3 Marine Traffic Mobile

3.1 General Design Goals

The general design goals of Marine Traffic mobile (MT mobile) were to:

1. Make use of the most popular functions of MarineTraffic web service; this selection was largely based on reviewing the navigation paths users commonly took while using the online service.
2. Get rid of traditional desktop-oriented design elements like buttons and menus. Instead, make use of established user interface design elements of mobile platforms.

3. Exploit mobile device sensing capabilities like: location awareness (GPS), accelerometer, gyroscope, and camera available on the mobile platform to offer a location-aware (context-aware) service.
4. Exploit gesture-based interaction affordances of mobile devices by making a careful selection of appropriate gestures through the conduction of simple tests for finding/remembering gestures suggested by Saffer [15].

3.2 Overview of the marine traffic mobile application

MT mobile is currently available on the Android, IOS and Windows Phone mobile platforms (Figures 2, 3, 4). These applications attempt to imitate the look and feel of the MarineTraffic web service, customized for improved interactivity and performance on each mobile platform. The basic functions available for MT mobile include:

- The ability to track live vessel positions, reported by more than 80,000 vessels per day on an interactive map. The application provides worldwide coverage of more than 3,000 ports and a significant number of open-sea areas.
- Search for current conditions in ports, vessel details, historical data and estimated time of arrivals. Port arrivals and departures are recorded in real-time.
- Search for any vessel in range but also historical data (last known positions, last port visited, track etc).
- Browse the Photo gallery with nearly 1,000,000 pictures of vessels, ports and lighthouses. Users can upload photos and rate available photos.

At the first user interaction with MT mobile, the JavaScript map is loaded and vessel information is retrieved via an asynchronous XML web request. During this first load a user is prompted to permit MT mobile to access his location so as to enable a location aware experience (load map on the user's location); however users may opt out at any time. Vessel icons are pinned onto the map at the geographic coordinates of the vessel and icon coloring is selected according to vessel type. To avoid making continuous web requests, MT mobile retrieves vessel positions for a larger map segment than visible (depending on the available device screen dimensions). Additional requests are made only when the user navigates out of this larger map segment. A user can interact with the map using most common multi-touch screen map interactions and with any element on the map by tapping on it. During interaction with the map, the user can request a "refresh" of vessel positions by taping the refresh icon at any time. MT mobile provides a clustered view of vessel positions when users selected zoom level is small, in order to avoid clutter; this also occurs in highly active areas such as popular ports.

When a user taps on a vessel icon, basic vessel information is presented with a link to additional vessel information. If a user selects to view the additional details a panorama page (a page panning more than one screen where a user can swipe left/right to view as a whole) is presented containing vessel history, vessel details but also vessel photos. A user can swipe left/right between vessel pages containing detailed, information, vessel track history and photos. At any time, a user can select to return to the original map or search for an additional vessel. A user can search for any vessel or port (XML web request) by tapping on the application bar available across the bottom of the screen. A user is provided with a list of matches for his query and can select any item from this list. This list is only fully populated when a user scrolls beyond viewable items. When a user selects to view a vessel or a port, he/she is presented with details, images and relative information and from there can select to load map on this location. The basic MT mobile functions, user gestures and technical issues are listed in Table 1.

Table 1. Overview of basic MT mobile functions, gestures and technical issues.

Function	Gesture	Technical Issues
OnLoad	n/a	Vessel and Map Data Retrieval
Navigate	Traditional map gestures	Incremental Map and Vessel Information Retrieval
Go To Position	Tap (option on application bar)	Enable location services on application and loads map on users position(context aware)
Zoom In/ Zoom Out	Pinch Open/Close	Clustered View
Examine Vessel	Tap, Swipe	PushPin annotation containing vessel basic details. Option to visit vessel detailed page
Refresh	Tap	Incremental Vessel Postion Retrieval
Search Ves-sel/Port	Tap	Generate Response . Go to details and option-al position on map
Customize Map	Swipe (Option Page ToggleS-witch)	User can customize application (show satellite imagery, vessel positions, vessel types etc)

MT mobile displays real-time data on an interactive map, which generates a number of particular requirements for a mobile application. The service has a heavy network dependency as users tend to often load many different views requiring the constant delivery of additional information. This service needs to continuously provide users with accurate data on constantly changing vessel positions. As network delays and drops are common in mobile application Internet access and the application needs to remain responsive even during these conditions, it was necessary to find the correct balance between data freshness, network consumption, and interface responsiveness. We needed to pay special attention to map loading issues and perform all other functions on asynchronous background threads providing users with incremental results. Thus, vessel loading and other processing were performed in the background. In addition, results had to be presented in an incremental fashion and are prioritized. We also exploited local mobile storage to cache static vessel information: users tend to re-view vessel and ports they are interested in, thus this information is cached locally.

During the design of the mobile versions of MT special interest was paid to providing personalized services for mobile users. A number of services were developed, including a “what’s near me” service, that shows information of ports and vessels close to them and a “where is my ship” service, providing passengers with the ability to closely track a vessels position. Specifically for vessel navigators MT provides service for the calculation of the closest point of approach (CPA) and time to closest point of approach (TCPA). In addition, users can contribute information to these services, by taking photos and providing ratings (thus supporting crouwdsourcing functions in a similar way to [16]).

Figure 2: MarineTraffic on Iphone Platform.

4 Design and Development Guidelines

We organized the aforementioned requirements of MT mobile to the following 8 principles and 25 guidelines for design and development.

4.1 Principle 1. Promote Application Responsiveness

Users expect an application to remain responsive at all times. Although a tolerable waiting time for web users seems to be approximately 2 seconds [17], this is unacceptable delay to thin-client applications [18]. Response delay less than 1 second is noticeable to annoying and more than 1 second is potential threat to mobile user experience [18], [19]. Thus, smartphone apps strive for crisp response of less than 150 ms [20]. Responsiveness is closely related to performance: performance limitations are inevitable on mobile devices but these must not limit application responsiveness. Guidelines:

1.1. Use background threads: To the best extent possible, keep all processing that will affect the user interface on background threads e.g. MT mobile asynchronously attempts to retrieve vessel information updates.

1.2. Display results incrementally: The main idea is to show the most important pieces of information first and improve on quality as resources become available; e.g. use multi-resolution encoding and make progressive transmission and computations [6]. This is done in various aspects of MT mobile, specifically when loading long lists and image thumbnails only the "in-view" elements are originally loaded.

1.3. Make all non-UI related function calls asynchronously: An asynchronous call ensures that your active execution thread never blocks for a significant amount of time.

1.4. Use animations with care: Although animations are engaging, these can quickly make the UI unresponsive. This issue generated a lot of debate in the design phase of MT mobile, as a balance was sought between engaging animations and application responsiveness.

1.5. Identify and correct memory leaks: Mobile applications have a finite amount of memory. Memory leaks are one of the major causes of memory exhaustion problems. Take care to flush memory of unnecessary data e.g. MT disregards all information after a page is navigated from.

4.2 Principle 2. Content First

Mobile applications require a minimalistic approach about navigation that emphasizes on providing content first. This is also documented in a number of more general mobile design guidelines, for example the two out of 6 user-centered design guidelines proposed by Cerejo [21] are: '1. Minimize navigation' and '2. Prioritize content'. Guidelines:

2.1. Present the most important information from the user perspective: This requires prior knowledge of what information is important for users and requires information prioritizing. User-centered methods like card sorting [22] can be of use in this. MT mobile only loads information on user request.

2.2. Provide affordances for user comprehension with appropriate use of media for presenting content: Even when text content is well-written and understandable by users, it takes much space and is not engaging, so we need to try to provide visual designs of textual information to afford (in the sense of affordances proposed by Norman [23] for user comprehension. For example, the icon used for representing a vessel on the map, affords user comprehension about: type (passenger, cargo, tanker, etc.), size, bearing, status (moving or stopped) and projected course orientation.

2.3. Promote highly user-rated content. This is particularly important for community-based applications that rest on crown-sourcing and can dynamically select content based on user ratings. When the user is viewing a vessel, MT mobile dynamically loads the most popular vessel photo as a background image for the information page; this is also done on the port page (Figure 3). Also, users can browse all vessel photos sorted by their ratings.

4.3 Principle 3. Promote Robustness.

Robustness is important for user experience and has been referred to as important usability principles in popular HCI textbooks at the early days of Graphical User Interfaces (e.g. by Dix et al [24]). Incorrect memory & CPU usage, network and bandwidth changes can often occur and crash a mobile application. Develop your application so that it can deal with all kind of exceptions. Mobile application crashes must be avoided at all costs. Guidelines:

3.1. Ensure system recovery from errors. This is achieved through exhaustive exception handling: uncaught exceptions can make your application to crash abruptly.

3.2. Provide meaningful user feedback in cases of errors. Ensure that the user is informed with understandable messages in cases of errors/exceptions and that he is provided with appropriate options.

3.3. Monitor network and bandwidth changes: To adjust processing accordingly. Before any network related query MT has to pass a number of validations so that no web exception is thrown that could potentially crash the application.

4.4 Principle 4. Respect user expectations.

Many mobile users are familiar to some extent to typical user interface / interaction conventions and responses, before making use of any particular mobile application. Therefore, a user expects from any mobile application to respond to ways he or she is accustomed to.

4.1. Make use of device (hard) buttons consistently: for example, a back or home button on the device must respond in the way that the user is accustomed to.

4.2. Provide timely feedback: for example progress indicators and appropriate messages when the application delays to respond.

4.3. Maintain the flow of the application throughout the interaction. For example, avoid breaking the application interaction experience by requesting the user to visit external webpages, to download and view files, etc.

4.4. A user has expectations about the look and feel of a native application for the particular device and OS. In respect to this principle MT mobile has followed a separate design approach for each platform (iOS, Android, Windows Phone) making use of particular design guidelines provided by each vendor. For example, MT mobile for Windows phone makes use of windows metro design and Bing maps in contrary to MT mobile for android that is built on Google maps.

4.5 Principle 5. Expect and design for latency.

Latency is the time it takes for a single data transaction to occur and it is measured in milliseconds (ms). In a CNET article, Marissa Mayer, Vice President of Search Product and User Experience, Google, said that when the Google Maps home page was “put on a diet” (shrunk from 100K to about 70K to 80K), traffic was up 10% the first week and grew 25% more in the following three weeks [25] [26] Network delays and drops are common in mobile application Internet access. Guidelines:

5.1. Isolate UI operations from the network to your best effort: To avoid delays or crashes due to network loading data.

5.2. Only load image thumbnails and let user select to view larger images: this requires design provisions at the web service side to provide mobile apps with appropriate images. MT mobile only loads thumbnail version of user uploaded photos until a larger size is specifically requested by the user. To minize network dependency and data download size, MT mobile makes use of lazy load algorithms [27].

4.6 Principle 6. Balance energy consumption

Energy consumption is one of the most important concerns of mobile application development and can result to poor user experience. Guidelines:

6.1. Perform CPU power intensive tasks, only when necessary.

6.2. Use orientation changes with care: Orientation changes put a heavy performance burden on the device processor; if they are not necessary, do not support them.

6.3. Exploit local storage: in order to minimize network connections. MT exploits local storage to cache static vessel information; thus minimizing unnecessary information retrieval

4.7 Principle 7. Exploit mobile interaction techniques.

Mobile users have different behavior patterns than desktop or laptop users, which are to some extent cultivated by their interaction with the mobile device. Understand and design for these. Each version of MT mobile (android, iPhone & windows phone) takes a platform specific approach to designing application interactions e.g. windows phone supports panorama and metro UI interactions. Guidelines:

7.1 Web design conventions can be replaced with mobile applications' counterparts: For example, replace zoom, menu bars and buttons with pinch, swipe and tap respectively (Table 1).

7.2 Make use of device sensing affordances to provide contextual interactions: E.g. make use of the camera to add a photo, or make use of the GPS to present information based on the user location.

7.3 Avoid using heavy input methods: Use techniques to make text entry more efficient, and require it only when necessary. When possible, use smart algorithms to help text input.

4.8 Principle 8. Carefully design application navigability

When designing websites we develop sitemaps and allow user to move using links between various webpage of our websites. For each such page our browser makes a new request and old pages are discarded. In mobile applications such pages remain dormant in memory as these as stateful devices. Guidelines:

8.1. Do not create a new mobile page when you can simply return to the previous one that is dormant in memory. This is one of the principal reasons of memory leaks.

8.2. Design navigation paths so that the creation of new pages is minimized.

Figure 3: MarineTraffic on Windows Phone. Port Information Page.

5 Discussion

5.1 Related work

A number of guidelines have been proposed to assist practitioners in mobile application design and development. Most mobile OS vendors have issued design guidelines for mobile application design and development: Apple iOS Humane Interface Guidelines [28], Android Design Guidelines [29], MS Windows Phone User Experience Guidelines[30]. These guidelines refer to a large number of issues specific to the software platforms and are certainly the starting point of study for any designer/developer. These sets of guidelines continue to be updated mainly to provide conformance to new versions of the operating systems.

These guidelines mainly refer to general design principles and to specific user interface elements and functions like icons, widgets and notifications. They also tend to emphasize design over development: this is natural since their aim is not only to provide help to potential designers and programmers but also to show off good and appealing design practices, and if possible characterizing each platform. However, proposed guidelines do not refer to specific types of mobile applications – with the exception of recent guidelines published by Apple for routing apps (on December 17, 2012); this may be the beginning of issuing guidelines for specific types of mobile apps from industrial vendors.

A number of mobile app designers have also proposed relevant guidelines. Weevers [31] proposes seven guidelines for designing high-performance mobile user experiences: (1) Define UI brand signatures; (2) Focus the portfolio of products; (3) Identify the core user stories; (4) Optimize UI flows and elements; (5) Define UI scaling rules; (6) Use a performance dashboard; (7) Champion dedicated UI engineering skills. We find these guidelines really useful especially for organizing team work with a focus on the design aspects of the development lifecycle. In addition Weevers's guidelines are general and relevant to any type of mobile app that offers some kind of business service.

Matzner [32] presents 10 mistakes to be avoided during the design and development of mobile apps. All are important to consider, however we distinguish the following: '1. Don't begin wireframes or designs without a flowmap', '6. Don't leave users hanging', and '7. Don't blindly copy style from other operating systems', which are relevant to our proposed principles: '1. Promote Application Responsiveness', '2. Content first', '8. Carefully design application navigability'.

Cerejo [21] discusses a user-centred approach to Web design for mobile devices. He outlines an iterative lifecycle process and presents a number of important guidelines for design: 'Simplify Navigation', 'Prioritize content', 'Minimize user input', 'Design for intermittent connectivity', 'Offer cross-channel consistency and integration'. From all available sets of guidelines this is probably one that is most relevant to our work, since that all these guidelines are relevant to our proposed set.

There are too few works on design and development guidelines for geospatial mobile apps with the characteristics of marine traffic. Samet et al [33] discuss the porting of the Web-based application of NewsStand, a mapping application that enables searching for spatially-referenced news information, to a native App. They identify the issues of: how to compensate for the absence of a hovering action; the implementation of an intuitive mechanism for undo; how to integrate an interaction restriction to one hand coupled with use of the thumb as the pointing mechanism; and the formulation of navigation shortcuts to avoid excessive network traffic. In our work we have not met the first two issues, and we have provided guidelines for the latter two in our best attempt to exploit mobile device affordances.

The focus of our approach is quite different with respect to these works since that: (a) we refer to a specific type of mobile apps: real-time, geospatial mobile applications; and (b) we attempt to connect design and development issues, since that any product design process requires an appreciation of the way the product will be built. Our approach is largely based on our experience with the design and development of the mobile applications of MarineTraffic. The characteristics of MarineTraffic called for a demanding approach to application design and development that sought to find the fine balance between application responsiveness, data freshness while providing a valued mobile experience. Our proposal consists of a number of 8 principles and 25 guidelines that collectively attempt to define an overarching approach to designing such applications. We also provide examples of applying these principles and guidelines for MT mobile.

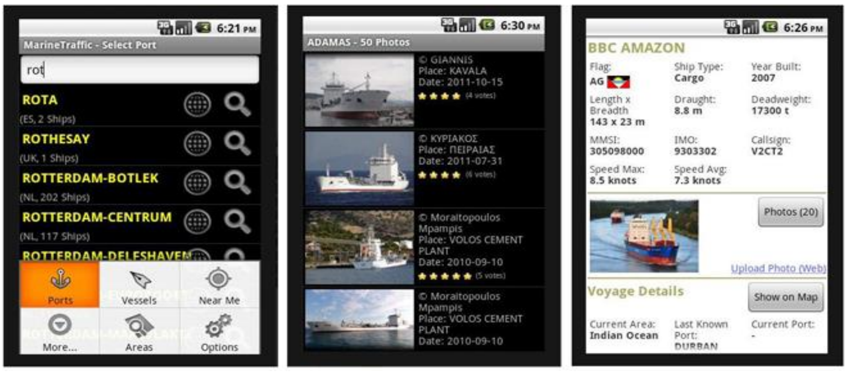

Figure 4: MarineTraffic android version.

5.2 Outline and Potential contribution

Mobile devices present a whole new ballgame for which web designers, but even traditional programmers, are not well prepared. Designers and developers from a web background need to conceive the differences in design created by network delays, network costs and screen size. The interface has to remain responsive regardless of the information retrieval or network latency. But most importantly mobile devices present an array of functions that can increase the user experience in methods not available on traditional PCS. Even for traditional application developers, mobile application development is a challenging task. Traditional software developers need to apprehend the complexity of memory management and CPU restrictions to maintain application responsiveness [6]. Without special programming considerations, such applications will exhibit unacceptable user responsiveness when resources are slow.

The design and development of real-time, geospatial, community-based mobile applications has to cope with particular challenges: (a) the ineffectiveness of current web design guidelines for usability and accessibility for this particular type of applications; (b) the technical issues for mobile application development that stem out of the requirement of constant geospatial information update and support for user-generated content over the Internet; (c) the exploitation of interaction affordances of current mobile devices to maximize the user experience.

The paper makes two distinct contributions to the field. First, we provide an overview of the complexities and perplexities of developing interactive real-time, geospatial, community-based mobile applications, as documented for the case of MT mobile. Second, we propose a number of design principles and development guidelines that stem out of the experience of MT mobile. Our work strongly couples design and implementation issues and we envisage that this orientation can be useful for other practitioners.

References

[1] MixPanel, “Celebrating Christmas with Data,” 2012.

[2] N. Group, “Connected Intelligence Application and Convergence Report,” 2013.

[3] U.S. Department of Health & Human Services, “Research-Based Web Design and Usability Guidelines,” 2006. [Online]. Available: http://www.usability.gov/guidelines/index.html. [Accessed: 15-Feb-2013].

[4] W3C, “Web Content Accessibility Guidelines (WCAG) 2.0. W3C Recommendation 11 December 2008,” 2008.

[5] B. Charland, A. Leroux, “Mobile Application Development: Web vs. Native,” *Communications of the ACM*, vol. 5, no. 54, pp. 49–53, 2011.

[6] G. H. Forman, “Wanted: Programming Support for Ensuring Responsiveness Despite Resource Variability and Volatility,” 1998.

[7] D. Ferreira, V. Kostakos, and A. K. Dey, “Lessons Learned from Large-Scale User Studies: Using Android Market as a Source of Data,”, *IJMHCI*, vol. 4, no. 3, p. 16, 2012.

[8] G. . Corey, “Nine Ways To Murder Your Battery (These Are Only Some Of The Ways),” in *Battcon’10*, 2010.

[9] M. Reddy, S., Mun, M., Burke, J., Estrin, D., Hansen, M., Srivastava, “Using Mobile Phones to Determine Transportation Modes,” *ACM Transactions on Sensor Networks 6(2)*, 2010.

[10] J. Bailey, D. Maier, K.-D. Schewe, B. Thalheim, and X. S. Wang, Eds., *Web Information Systems Engineering - WISE 2008*, vol. 5175. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[11] Google, “Google Maps API,” *Google Developers*, 2012. [Online]. Available: https://developers.google.com/maps/faq. [Accessed: 18-Jan-2013].

[12] Sensa Developer Forum, “PR3: Google Map does not respond on Map control click events in mobile Browsers,” *Sencha Touch 2.x Forums / Sencha Touch 2.x: Bugs /*, 2012. [Online]. Available: http://www.sencha.com [Accessed: 18-Jan-2013].

[13] J. Blagdon, “Google enabling Maps access for Windows Phone after uproar,” *TheVerge*, 2013. [Online]. Available: http://www.theverge.com/. [Accessed: 18-Jan-2013].

[14] G. KUNITO, “Issues for mobile Ajax for cellular users,” Hikari-no-oka, Yokosuka, Kanagawa, 239-0807 JAPAN, 2012.

[15] D. Saffer, *Designing Gestural Interfaces*. O’Reilly Media, 2008.

[16] S. Shah, F. Bao, C.-T. Lu, and C. I.-R., *CROWDSAFE : Crowd Sourcing of Crime Incidents and Safe Routing on Mobile Devices*. 2011, pp. 521 – 524.

[17] Forrester Research, Inc, “eCommerce Website performace Today, An Updated Look At Consumer Reaction To A Poor Online Shopping Experience,” 2009.

[18] N. Tolia, D. G. Andersen, and M. Satyanarayanan, “Quantifying Interactive User Experience on Thin Clients,” *Computer*, vol. 39, no. 3, pp. 46–52, Mar. 2006.

[19] S. Abolfazli, Z. Sanaei, M. Shiraz, H. Keshavarz, and A. Gani, “An Evolutionary Study of Rich Mobile Applications,” *CoRR*, 2012.

[20] A. G. Saeid Abolfazli, Zohreh Sanaei, Muhammad Shiraz, Hassan Keshavarz, “An Evolutionary Study of Rich Mobile Applications,” *CoRR*, vol. 1206.6222, 2012.

[21] L. Cerejo, “A User-Centered Approach To Web Design For Mobile Devices,” *Smashing Magazine*, 2012. [Online]. Available: http://mobile.smashingmagazine.com. [Accessed: 15-Mar-2012].

[22] W. Hudson, “Card Sorting,” in *The Encyclopedia of Human-Computer Interaction*, 2nd ed., Interaction Design Foundation, 2012.

[23] D. Norman, *The Design of Everyday Things*. 1988.

[24] A. Dix, J. Finlay, G. D. Abowd, and R. Beale, *Human Computer Interaction*. Prentice Hall, 2003, p. 832.

[25] Nokia Siemens Networks, “Latency- The impact of latency on application performance,” 2009.

[26] D. Farber, “Google´s Marissa Mayer: Speed wins,” *CNET News.com*, 2006. [Online]. Available: http://www.zdnet.com/. [Accessed: 09-Feb-2013].

[27] R. Cameron, *Pro Windows Phone 7 Development*. Berkeley, CA: Apress, 2011.

[28] Apple, “iOS Humane Interface Guidelines,” 2008.

[29] Google Inc, “Android Design Guidelines.”

[30] Microsoft Corporation, “Microsoft Windows Phone,” 2012.

[31] I. Weevers, “Seven Guidelines For Designing High-Performance Mobile User Experiences,” *Smashing Magazine*, 2011. [Online]. Available: http://uxdesign.smashingmagazine.com/. [Accessed: 15-Mar-2013].

[32] R. Matzner, “Designing a Mobile App? Don’t Make These 10 Mistakes,” *Mashable.com*, 2012. [Online]. Available: http://mashable.com/. [Accessed: 15-Mar-2013].

[33] H. Samet, M. D. Adelfio, B. C. Fruin, M. D. Lieberman, and B. E. Teitler, “Porting a web-based mapping application to a smartphone app,” in *Proc. of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS ’11*, 2011, p. 525.