Volunteered Geographic Services

The Case of Fleet Management

Douglas Hamilton and Krzysztof Janowicz

April 2014

Table of Contents

1.0 Motivation

Volunteered Geographic Information (VGI) has been a success story so far, especially with respect to the acquisition and maintenance of data such as in the OpenStreetMap project or the tracking of wildfires in California. With the advent of mobile computing for the masses, VGI can be taken to the next level by the idea of citizens as sensors (Goodchild 2007). One of these sensors is GPS providing location data and available on most modern smartphones. In the April 2010 issue of ESRI's ArcUser magazine Goodchild notes that mobile computing will be the next big step in VGI (Goodchild 2010). The several VGI-based Haiti crisis projects have demonstrated the power of volunteered information for emergency response. Most of these projects have helped to acquire up-to date data about infrastructure such as roads or hospitals as well as locations of people in needing help. We propose to take this approach to the next level by helping volunteers to offer services in addition to data. Therefore, we intend to implement a fleet management service based on VGI principles, the Web as the communication infrastructure and GPS-enabled smartphones as the only requirement. In emergency situations such as earthquakes or flooding, having an ad-hoc infrastructure to track vehicles of all sorts is a crucial task. For instance, these vehicles may be used to transport food or equipment to the emergency zone and were used as private vehicles before.

My personal motivation for this project comes from two major sources. One, I have worked in the fleet management market as an engineer for almost 20 years. I have been a part of this market as it grew from infancy into a mature market. Second, I live in the Central Florida area that was hit hard in 2004 by three hurricanes – Charley, Francis and Ivan. Many communities in Central Florida were without power for a week or more. Nobody had any idea when, even if, the power would be restored. There was no infrastructure in place for residents to request assistance or aid. As neighbors we pooled our resources and came together to clear our street and yards of downed trees and debris. Taking that thought farther a system where the entire area could pool resources and assist each other, with focus and direction, would have been of great assistance.

Using use cases a series of research questions will be discovered. These will be discussed and boiled down to the research question that this paper will be focused on. System specifications for a fleet management service will be developed and discussed.

2.0 Use Cases

The following use cases describe how a VGI service, or VGS – a Volunteered Geographic Service, could be used to assist disaster responders in a crisis situation. First, three crisis situations are described. This will be followed by an analysis of how a Volunteered Geographic Services app could be used in each of the use cases. The analysis will include an overview and an example of the Volunteered Geographic Service app in use.

2.1 Three Crisis Situations

2.1.1 Torrential Rains Strand the Residents of a Rural Neighborhood

Torrential rains have caused flooding in a rural area of southeast Orlando, Florida. Local residents are in need of food, water, sandbags and other supplies. Travel in and out of the area is restricted by the flooding. Some roads remain passable while others have been partially or

completely washed out. In some areas residents are stranded in their community. Emergency workers need assistance to get supplies to the area. They also require assistance in getting to injured residents and, in some cases, to get them to a local hospital for more advanced treatment.

Figure 1: Photo by Carrie Kellenberger. Rainy Season in Borneo, Flash Floods, and Why You Should Always Drive A Pickup. http://www.myseveralworlds.com/2011/03/31/rainy-season-in-borneo-flash-floods-and-why-you-should-always-drive-a-pickup/

2.1.2 Tornadoes Strike the Midwest

In mid-November 2013 intense thunderstorms and tornadoes swept through the Midwest causing significant damage (Fox News, *Tornadoes, severe thunderstorms kill at least 6 in Midwest)*. With the storms moving rapidly – some as fast as 60 mph –people little time to seek shelter if they were relying on watching the sky alone. At least six people were killed and neighborhoods were leveled in Central Illinois. The community of Washington was hit particularly hard. Subdivisions of twenty to thirty homes were completely leveled leaving not a single home standing. Physicians and nurses from surrounding communities were desperately needed. Volunteers were needed to assist in the search through the tornadoes' destruction to go house to house in search of survivors. Sixty five tornadoes struck, the bulk of them in Illinois, according to the National Weather Services' website.

Figure 2: "Before and after the storm: The heartbreaking aerial photos that show the incredible scale of tornado destruction in Illinois town" by Daily Mail Reporter and Reuters Reporter. http://www.dailymail.co.uk/news/article-2509622/Washington-Illinois-tornado-Aerial-photos-incredible-scale-destruction-town.html

2.1.3 Hurricane Charley

In 2004 Hurricane Charley struck the western coast of Florida and made an unsuspected fast track towards Central Florida. Most residents in the area were caught unprepared as the central Florida area was near the edge of the hurricane's forecast cone (Wikipedia, *Hurricane Charley*).

The residents were not prepared with extra water, food and other supplies. Those that had generators lacked gasoline for them – but most did not have generators at all. Many trees were uprooted, others snapped. Power lines were down all throughout the hurricane's swath of destruction.

Orange County, which includes the Orlando metropolitan area, was particularly hard hit. Emergency services were not prepared for the level of destruction to the region's infrastructure caused as Charley quickly roared through the area.
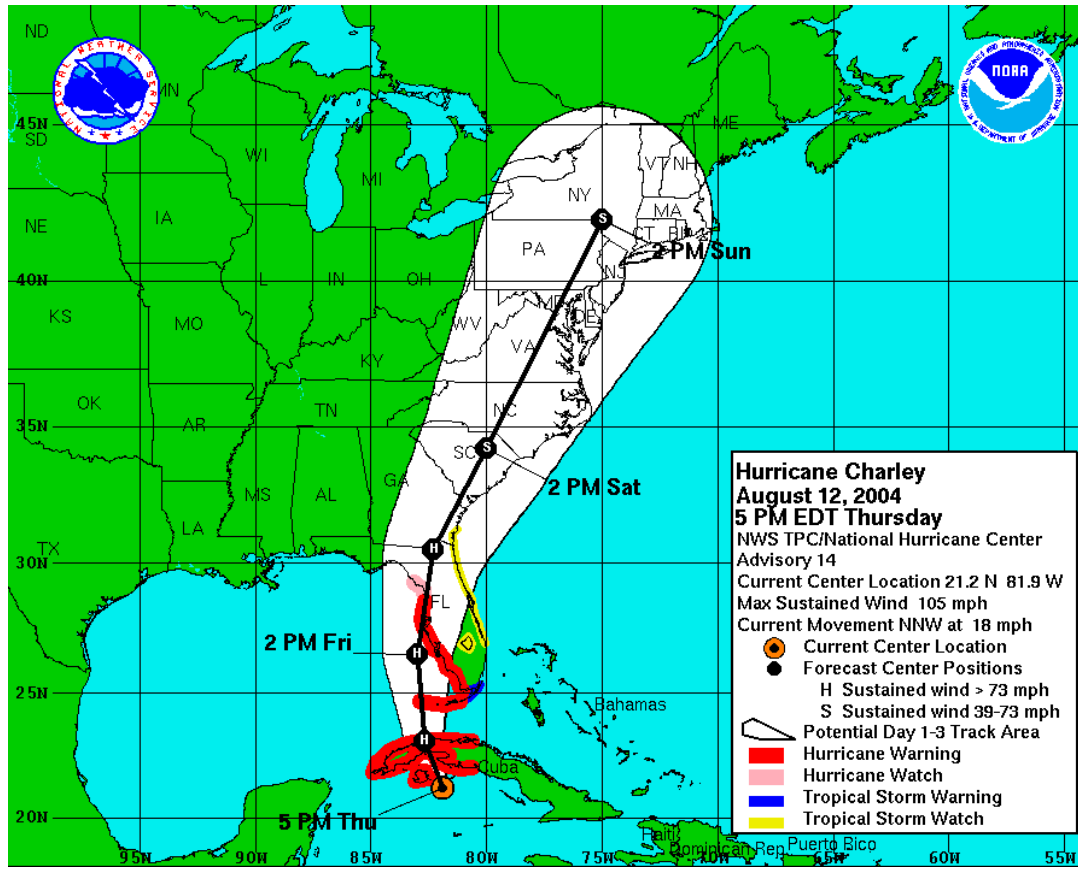
2.2   Volunteered Geographic Services App in Three Crisis Situations

In order to assist the emergency workers a VGS service was offered to the disaster responders that would enable volunteers to provide their location and capabilities by registering via their smartphone. The information required to register includes their capabilities (i.e. vehicles and their capabilities and capacities; any supplies they can provide; their personal capabilities) and their geographic location. The latter is automatically provided by the volunteers' GPS-enabled smartphones. No personal information is required for registration.

The VGS service is also available to residents that were affected by the crisis. They can register their location and needs. They can list of there are needs that are critical in nature. For example, if there is an immediate need for medical assistance they register this. Emergency personnel will know that the resident needs immediate assistance and can dispatch responders and volunteers to their location.

Emergency supervisors review the volunteers' information and promptly deploy them to the areas where they can offer the most assistance. They can also pair them with emergency responders, as needed.

2.2.1 VGS App and Torrential Rains Strand the Residents of a Rural Neighborhood

Overview

Due to the nature of the disaster, volunteers without four wheel drive vehicles are asked to ferry supplies from the supply depots to locations where other volunteers can access them and load up their four wheel drive or all-terrain vehicles. These vehicles are directed by the emergency supervisors to the areas where the supplies they carry are most needed. Vehicles with smaller capacities are routed to locations where the need is less. Volunteers with greater vehicle capacities are routed to the locations where the need is greater. Other volunteers are providing transportation in and out of the area to paramedics and other emergency personnel.

The progress of the volunteers is constantly monitored since their GPS-enabled smartphones periodically update their location. This allows for the supervisory staff to more efficiently route the volunteers and to track their progress towards their destinations. Thus the residents in the flooded area know that assistance is not only on the way but they know approximately when it is due to arrive and what is coming.

Example

In this example the use case will be described for a volunteer with a four-wheel drive truck that is capable of carrying 5 passengers and 3500pounds of cargo.

The crisis has happened and Warren, a volunteer, registers his capabilities and supplies with the county's VGS service. Warren was fortunate in that his residence was spared any damage from the rains and flooding. Although he is stranded like the other residents in the area, he is fortunate to have purchased extra bottled water in anticipation of the upcoming hurricane season. He also has a 5 passenger four wheel drive truck with a 3500 pound cargo capacity and a full tank of gas.

Warren had read about the new VGS service in the local newspaper that the county had implemented and registered himself as a volunteer. He provided his location, available supplies and his vehicle's capabilities. A back office worker, responsible for reviewing volunteer registrations, received Warren's registration and saw how he could be of immediate assistance to emergency responders attempting to reach the stranded residents. He sends Warren the location of the emergency responders that are having difficulty reaching the stranded residents. In addition, a notification is sent to the responder leader that a volunteer is en-route to pick them up and take them to their destination.

Warren departs his resident and proceeds to the location of the responders. He must travel a washed out road, utilizing his truck's four wheel drive capabilities. He meets the responders and with them, their equipment and supplies loaded into the truck he travels back into the neighborhood and delivers the responders to their destination.

2.2.2 VGS App and Tornadoes Strike the Midwest

Overview

In the aftermath of the tornadoes, emergency workers and volunteers rush to the areas that were destroyed by the tornadoes. As they arrive they check in with the emergency workers supervising the search for survivors. During this process they download the VGS app and register their information. They are then directed by emergency supervisors to search specific areas and to

report their findings. Any survivors found are rescued as quickly as possible since the manpower needed to extricate them from the wreckage is directed to exactly where it is needed via the location provided by the VGS app on the emergency worker's or volunteer's smartphone.

Example

Mary Ellen, a resident that was not in the path of the tornadoes, arrives at the temporary headquarters of the emergency responders and volunteers to assist with the search. She downloads the VGS app and registers her information with the VGS service. Mary Ellen is then sent, along Beth, a paramedic, to search the destruction along Oak Street, which is a half mile to the north. The team walks to the north through an area that has already been searched until the reach Oak Street. They begin to search through the remnants of houses destroyed by the tornadoes looking for residents that had become trapped. At the third house they search Mary Ellen hears faint cries for help. Together with Beth they locate an elderly man trapped in the debris. Mary Ellen and Beth update their location with the VGS app and request immediate assistance as they are not able to clear the debris that has the elderly man trapped. Other responders are dispatched to their location with tools and equipment. As this team arrives and begins the work of clearing the debris Mary Ellen and Beth proceed down the street in search of more survivors.

2.2.3 VGS App and Hurricane Charley

Overview

Hurricane Charley did an enormous amount of damage throughout the Central Florida area. Unfortunately, the damage was so great that many areas were totally cut off from all modern conveniences. Entire communities were without power. Many cellular towers were damaged and, although they had backup power, it was not meant to provide power to the tower for days on end. The cellular system was placed into "emergency mode" where the only cellphones that worked were those of emergency personnel. Thus, even if local residents happened to be lucky enough to have power, they could not access the cellular system.

In a situation such as this the only way to find out the status of the many neighborhoods was to have emergency workers and/or volunteers visit them in order to assess the situation and then report this to their assigned supervisors. Thus, the needs of the local residents were communicated verbally to supervisors who then coordinated the efforts to assist the residents. The app could have been used to track the emergency workers and volunteers as they traveled to the various communities to note their needs. Then, when the supplies were ready, they could track them back as they delivered supplies to the residents. Thus, they would know exactly what areas had received supplies. They could then concentrate on other areas.

Example

Paul, his pregnant wife Suzy and their son spent a restless night as the hurricane roared past. In the dawn they step outside to check out their house. The house appears to be undamaged but they are without power. Looking down the street and at their neighbors' houses they see many trees down, branches and other debris are scattered everywhere. It appears that only one neighbor suffered structural to their home. That neighbor was fortunate that the large pine tree that was toppled fell on their unattached garage. Leaving Suzy to watch their son, Paul joins the other men to remove the debris and trees that have blocked their street. Unbeknownst to them, this scene is repeated on many of the other streets in this rural community, including the main road

that is their only way in or out of the area. As the residents clear the streets their homes are on, they begin to assist others clear the main road so that it is passable. On the fourth day they finally have the main road cleared and the first emergency vehicles are able to reach the area and assess the damage.

The emergency responders also provide news about the damage to other areas in Central Florida and, as hard hit as these residents were, there are other areas that suffered even more damage. They are told that it would still be days before power can be restored. The responders talk to the residents about their supplies and whether or not there are any immediate needs. The addresses of those with immediate needs are entered into the VGS app the responders have installed on their phones, as well as the nature of their needs, be it medical or food and water. Later in the day other responders arrive bringing needed supplies and medicines. Some paramedics arrive to aid those with minor injuries caused by the hurricane, or in some cases, while working to clear the debris from roads and yards.

3.0 Research Questions

The following is a list of research questions that relate to the creation of a VGS solution to assist disaster responders in a crisis situation.

1. How can one abstract from the particular hardware and protocols to develop a plug & play like infrastructure for volunteered geographic information like fleet tracking?

   In an emergency situation there is not enough time to implement a solution unless it is plug & play. There is no time to purchase specialized equipment and deliver it to the emergency responders and volunteers. What they have on hand must be utilized. Since most people now use GPS enabled smartphones these could be utilized in an emergency.

2. Can the solution efficiently provide data to track volunteers and their services?

   Utilizing GPS enabled smartphones, as most are now, for data gathering and the world wide web for data traffic the solution can be very efficient. Position and/or capability updates can be sent to the base of operations almost immediately. Thus, those in charge of the emergency situation would always have current information regarding the status and location of volunteers.

3. Is a solution created using free and open source software (FOSS) stable and reliable enough to be used in an emergency situation?

   Many open source solutions are being used now in mission critical applications. A lot of FOSS is mature and have a strong base of community support. MySQL, for example, is the second most used open source database management system in the world. MySQL is used by many popular and high profile websites such as Google, Facebook, YouTube and Twitter (Wikipedia 2014).

4. Can a VGI based fleet tracking solution be used effectively to assist official support in an emergency situation?

   A VGI based fleet tracking solution adds a tremendous capability in an emergency situation. Supervisors and other emergency workers can not only know where the official responders are but they would also know the locations of the volunteers. This would make the task of coordinating their efforts much easier and would keep them informed as

to the progress of both responders and volunteers. This would be a huge addition to the management of the crisis situation.

5. Does FOSS make the use of VGI in emergencies more viable since there are no inherent costs or licensing issues?

   Since FOSS has no inherent costs and there are no licensing issues government agencies would be more likely to adopt the solution. The agency would not need to seek budget approval of the costs. The solution could be implemented much quicker since it requires little funding.

6. Can a web based solution be created that takes advantage of FOSS to eliminate the cost and licensing issues inherent in proprietary software?

   A web based solution can certainly be developed taking advantage of FOSS. For example, OpenGTS is an open source fleet tracking system. OpenGTS works with MySQL and has been used as the starting point for many commercial fleet tracing solutions.

7. How to realize a request-offering mechanism for such a service similar to a Service Oriented Architecture (SOA)?

   A Service Oriented Architecture for the solution adds simplicity to the design and development. Developers can create the various pieces of the solution simultaneously which decreases total development time greatly. Changes and customizations can be made to one piece without affecting the other pieces. Management of the solution is simplified by this approach.

8. How to visualize and interact with fleet tracking information on mobile devices?

   Using mobile devices – smartphones, especially – is the key to enabling volunteers to share their position and capabilities. Using something they already have reduces the cost of implementing the service. The app designed for this purpose must be simple and easy to use. It should have an intuitive interface that requires little training, preferably none.

3.1   Question Research Will Be Focused On

In this paper only one of the research questions will be focused upon:

1. How can one abstract from the particular hardware and protocols to develop a plug & play like infrastructure for volunteered geographic information like fleet tracking?

This question will be the focus because if this service cannot be created then the other questions simply do not matter. It is the enabling technology.

Time is of the essence in an emergency situation. The hardware and protocols to be used must already be in place. There is not enough time to wait for budget approval for the purchase of specialized equipment. Whatever is at hand must be utilized.

In emergency scenarios volunteers use what they have. Although there may be a need for special equipment, volunteers are not typically equipped with it. The service will take advantage of the volunteers' existing GPS-enabled smartphones to provide geographic information (Arnold 2010) and their vehicular capabilities and capacities. This expands on the existing uses of VGI in crisis situations where it has been used successfully (Goodchild 2007). Using smartphones also

provides greater internet access to disaster response workers as 67% of the world's population has access to mobile phones. Thus, disaster response workers do not require high speed broadband connections and standard PCs to access data created by volunteers (Zook, et al. 2010).

4.0 Methods, Tools and Technologies

Based on previous work on VGI, we will investigate user paradigms and requirements for volunteers in an emergency situation and within the fleet management context. For example, we will expand on the application of VGI during the recent Haitian earthquake. There were a number of ways in which VGI played a part in the emergency response to the earthquake but in particular these four ways: Ushahidi, OpenStreetMap, GeoCommons and CrisisCamp Haiti. There were two crucial questions following the earthquake in Haiti that required immediate answers: Where is help needed? Who needs the help? (Zook, et al. 2010) This paper expands on this by adding: What volunteers can help? Where are they? What service(s) can they offer?

Being able to track volunteers and their assets (i.e. transportation, cargo) will assist official support as the volunteers will be able to be directed to locations that require their assistance and cargo.

As the volunteers progress toward their destinations they can be tracked and their arrival time can be estimated and forwarded to personnel at the location. Those personnel can also be made aware of what assets the volunteer(s) are bringing as well.

The visualization could be through any device that is capable of browsing the web. A website could be accessed where the information can be viewed.

We will use the notions of a micro-Spatial Data Infrastructure (SDI), Service Oriented Architecture (SOA), RESTful Services and Web 2.0 as starting points to develop an ad-hoc infrastructure for fleet management based on smartphones and free and open source software (FOSS). The solution will be created using FOSS to eliminate cost and licensing issues.

SDI, an interactively connected framework of metadata, users, tools and spatial data, will provide the capability to use the VGI data efficiently and with flexibility (Wikipedia, *Spatial data infrastructure*). As this project is a lightweight linked spatiotemporal application a micro-SDI will be used. A micro-SDI consists of lightweight and simplified Open Geospatial Consortium (OGC) services that are imbedded into web pages and applications (Janowicz et al.).

SOA will be used because the service can be created in a modular nature and the interface will be independent of the implementation. SOA also allows for the creation of new applications on top of the existing applications which provides for greater re-usability. The clients, or users of the applications in an SOA implementation, are isolated from the changes to the service (Kodali 2005).

RESTful Services will be used because it is a simpler approach than WDSL and SOAP (Rodriguez 2008). RESTful Services use HTTP methods rather than its own unique methods, as is done in SOAP (Wikipedia, *Representational State Transfer*). It is also stateless – the server responds to the client request but does not maintain any information regarding the request. Once the request is processed it returns to its RESTful state.

Web 2.0 is selected because it facilitates "interactive information sharing, interoperability, user-centered design, and collaboration on the World Wide Web" (Wikipedia, *Web 2.0*). It is also

about "creating, collaborating, editing and sharing user-generated content online" (Discovery Education). Web 2.0 has increased the capability for volunteers to assist in disaster response via spatial analysis and mapping (Zook, et al. 2010).

FOSS will be used wherever possible. Since there is no cost to the volunteer they would be more likely to make use of the solution and participate in the VGI endeavor. Organizers would be more likely to adopt a FOSS solution because of its low cost and that there are no licensing issues.

Many FOSS applications are mature and stable enough to be used in most any solution, even an emergency. For example, MySQL, PostgreSQL and OpenOffice have been deployed in thousands of places in many mission critical applications. MySQL, PostgreSQL, MapServer and OpenStreetMap are other examples of FOSS that could be used.

5.0 Outcome

The outcome of this research will be a design of the Android app and of the website that make up the two major components of the system. The design will include a detailed description of the app and website as well as UML diagrams detailing the various components.

The components:

- users: residents, volunteers, etc.,

- tools: Android smartphones, the internet, etc. and

- spatial data: GPS position reports, graphical map display, etc.

all interactively connect in forming the Spatial Data Infrastructure of this solution.

5.1 Android App Design

The Android app is primarily a graphical user interface (GUI) and some internal mechanics to provide actions based upon a user's selections in the GUI. The app will maintain no data other the user's registration information, any messages sent to or received from the website and the location of the next destination, if applicable. All other data including but not limited to the current position will be collected as needed.
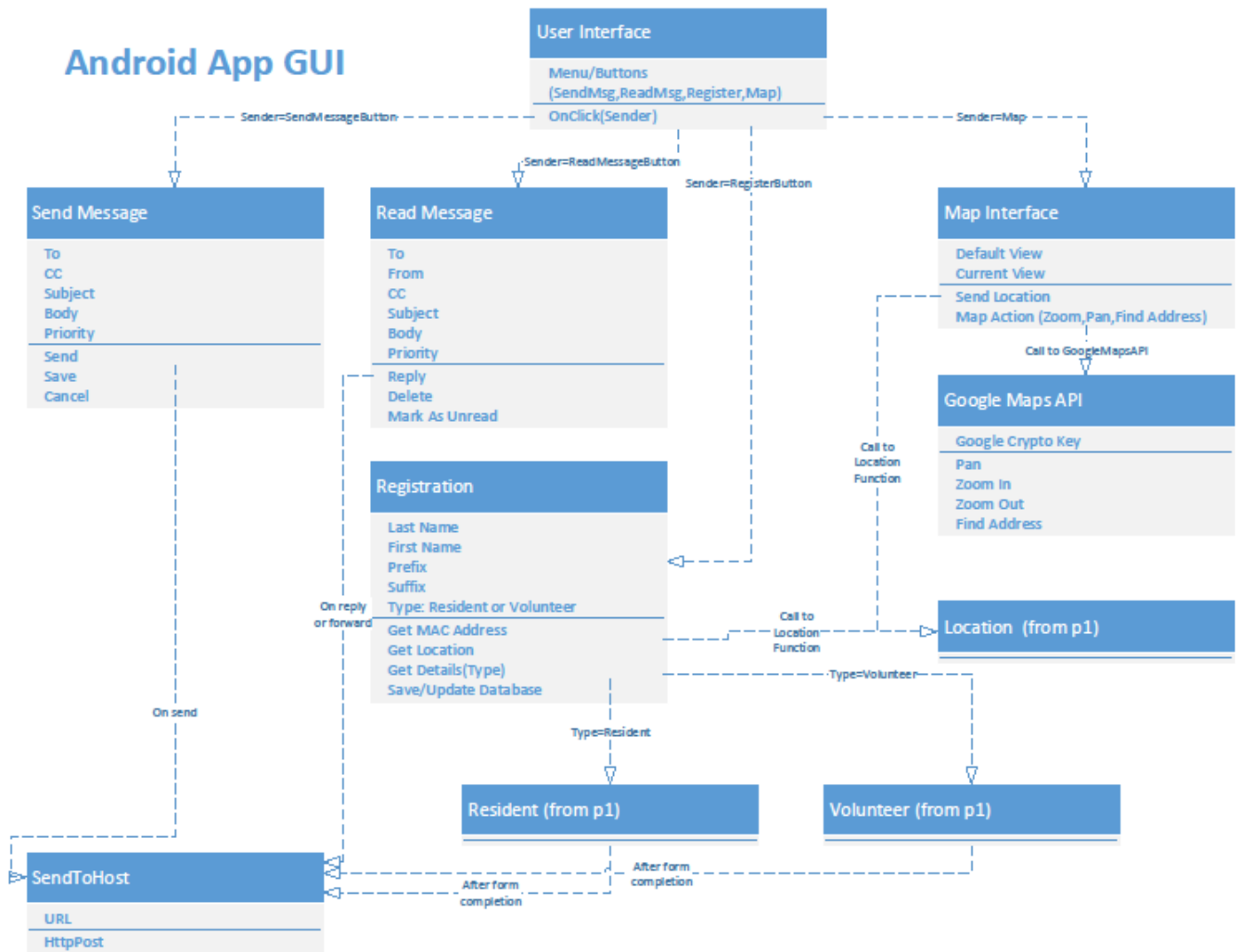
Figure 4: UML Diagram of the Graphical User Interface of the Android App

The GUI is designed as it is for a number of reasons. First, to make the app easy to use for untrained volunteers and residents a simple, uncluttered and intuitive interface is desired. Therefore, there is a simple menu system – it could be a drop down menu or even a series of small buttons – for the main actions that are available to a user. These are register, send a message and read a message. The main view will be the map and the current location of the user and their destination, if applicable, and this simple menu. To keep it intuitive the send and read messages forms should look to the common email interface. The registration form should only request information that is required to keep the interface simple and uncluttered.

5.1.1 Graphical User Interface

The GUI will be a map view showing the user's current location and the location of the next destination, if applicable. The interface will include buttons, or a menu with links, to register, send a message, or read an incoming message. The map interface will include pan, zoom and find address capabilities. All data sent to the web service are sent using HTTP Post methods. This

is because the app is RESTful and no unique methods are used. The HTTP Post is made over the web taking advantage of Web 2.0.

Registration Process: When the user selects registration the registration form will be displayed. Upon display, the MAC Address of the smartphone, phone number and the current location will be provided by the smartphone and automatically entered into the form. The MAC Address and phone number are used as unique identifiers for the volunteer or resident. These fields will not be user-editable. The registration form will include fields to select the user type (resident or volunteer) and to enter the user's name. The form will display the appropriate fields when resident or volunteer is selected. This helps to keep the interface from being cluttered with fields not needed for the task at hand.

If the user is registering as a resident, the form will include selections for common needs (i.e. food, water) as well as an area to register other needs including medical needs. In addition, the medical needs can be labeled as to their urgency.

If the user is registering as a volunteer, the form will include selections for common supplies (i.e. food, water, gas) as well as an area to register other capabilities including personal capabilities (i.e. nurse, veterinarian, construction foreman) and vehicular capabilities (i.e. type, cargo and personnel capacities, 4WD, etc.).

Upon completion of the form the user has the option to send the information to complete the registration process or to cancel it. After either selection the interface will return to the map view. If the registration was sent the web service will send a message back with confirmation that the form had been received.

Send Message Process: When the user selects the send message selection a form will be displayed where the user can enter the message text and then send, or cancel, the sending of the message. After either selection the interface will return to the map view. This would be similar to email except the recipient is fixed.

Read Message Process: When the app receives a message a notification will be displayed. When the user selects to read the message the message text will be displayed. The user then has these options: reply, delete and mark as unread. After either selection the interface will return to the map view. Again, this would be very similar to email.

Map Interface: The map interface will be powered by Google Maps. It will include zoom, pan and find address capabilities. Zoom will be accomplished by utilizing the common smartphone interface for zoom and pan.

- Zoom in is accomplished by fingers spreading apart.
- Zoom out is accomplished by fingers on screen closing together.
- Pan is accomplished by one finger dragging map view to a new location.
- Find address includes the display of a dialog where the user can enter an address which will then cause the map view to change to show the address's location.

5.1.2 Internal Mechanics

There are internal mechanics that must exist behind the GUI to complete the action selected by the user. These are described here for each of the GUI's options.

HTTP Post is used for data the Android app sends to the server because RESTful services are being used. The Android app is RESTful in that it sends the registration, position update or message and returns to its RESTful state. When the user selects an action or if a message is received it exits the RESTful state, handles the user action or message received and then returns to the RESTful state.

Registration Process: The initial mechanics behind the user's selection of registration is the displaying of the registration form as the user interface. If the user elects to send in the registration the internal mechanics package the registration information into the HTTP POST format and completes the action by posting it. This process sends the information to the web application. The main GUI (the map interface) is again displayed.

When the web application has processed the HTTP Post a response is returned that the new registration was successful or not. If the user is already registered the web application returns a response that the information was successfully updated or if it did not. These responses are displayed on the GUI as a notification that is displayed for a predefined period of time (i.e. 15 seconds) and then cleared.

Send Message Process: The mechanics behind the sending of a message includes the display of the send message dialog with the options to send or cancel the message. When sending the text message is formatted as an HTTP Post and includes the MAC Address of the smartphone (its unique identifier) and the message text. This is then posted to the web application. The main GUI (the map interface) is again displayed.

When the web application has processed the HTTP Post, a response is returned that the sending of the message was successful or not. These responses are displayed on the GUI as a notification that is displayed for a predefined period of time (i.e. 15 seconds) and then cleared.

Read Message Process: The mechanics behind the reading of a text message are the display of the message text and then awaiting the user selection of returning to the main GUI, to reply to the message or to mark the message as unread. If the user elects to reply to the message the send message process is called. If the user elects to mark the message is unread this is done and the user is returned to the main GUI.

Map Interface: As the map interface is powered by Google Maps all user selections will cause commands to be sent to the Google Maps interface to accomplish them. This includes all of the map interface's options of zoom, pan and find address.

5.2   Website Application Design

The web application consists of a GUI, the internal mechanics to provide actions based upon a user's selections in the GUI, a database and forms to view/edit the data contained within the database.
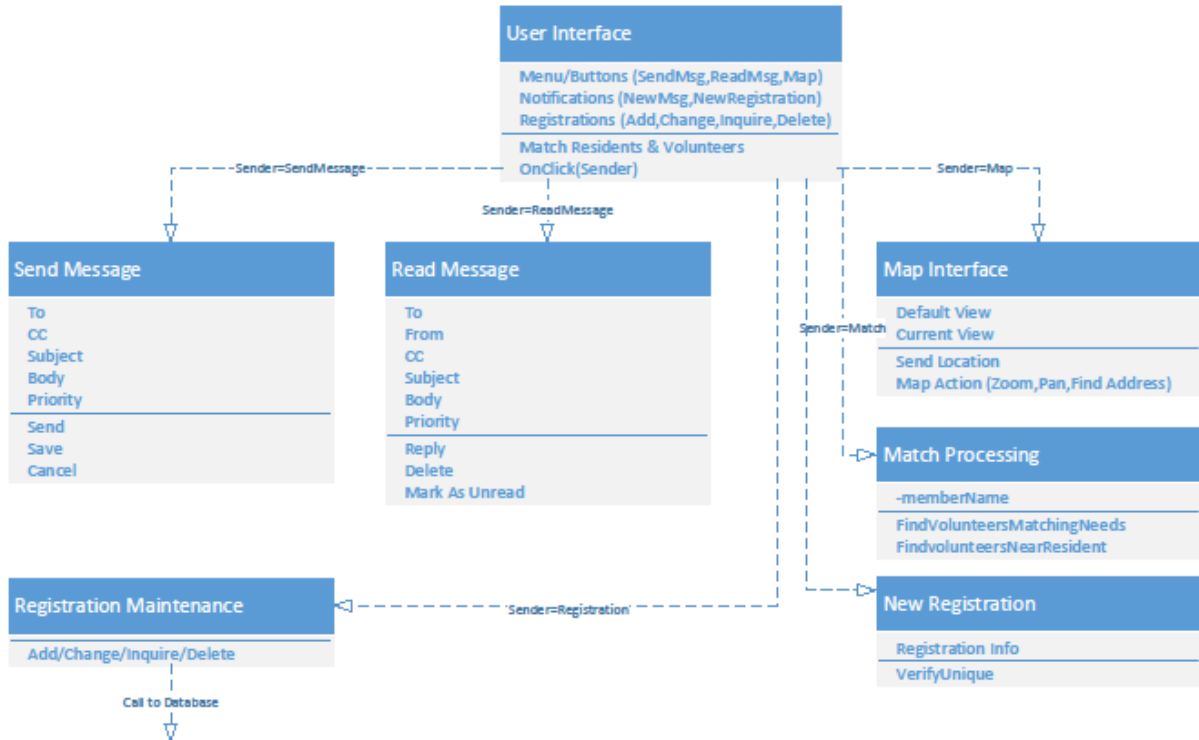
Figure 5: UML Diagram of the Website Application's Graphical User Interface

5.2.1 Graphical User Interface

The main GUI of the web application is a map view with the locations of the volunteers, residents and emergency workers displayed. The map interface includes options to zoom, pan and find an address. Each type of registrant (volunteer, resident or emergency worker) is displayed on the map with a different icon.

In addition to the main GUI there will be options to

- review incoming registrations and process them

- options to send or read text messages

- an option to request the current location of a resident, volunteer or emergency worker

- an option to access the data contained within the database

Future Enhancements: These are future enhancements that would add additional robustness to the VGS application:

- An additional future enhancement could be different icons to display immediate medical requirements for residents, different icons for the different types of emergency workers and so forth.

- Tools to assign volunteers and/or emergency workers to a resident which then sends text messages as needed to the parties involved.

5.2.1.1 OpenGTS

The FOSS application *OpenGTS™* could be used as a basis for the website application. It has all of the required map functionality including compatibility with Google Maps and OpenStreetMap. Modifications would be needed to provide messaging capabilities and registration processing. The database (MySQL) would require additions to handle the items unique to the VGS application. These include resident, volunteer and emergency worker needs and capabilities.

5.2.2 Graphical User Interface's Internal Mechanics

There are internal mechanics that must exist behind the GUI to complete the action selected by the user as well as to process posts received by the Android apps in use. These are described here for each of the GUI's options. Also described will be the actions caused by HTTP Posts from connected Android apps.

Since RESTful services are being used the website application maintains no information regarding the user request or data received from the connected Android apps. It processes the request and then returns to its RESTful state.

Map Interface: On the map interface the user has options to zoom, pan and to find an address. These options are accomplished by calls to the OpenStreetMap (or other map engine in use, i.e. Google Maps).

Zoom and pan are typically accomplished by changes to the mouse pointer to initiate the map action (zoom in, zoom out and pan tools). Zoom options could also be accessed via a zoom level slider component. If provided by the map engine, the mouse wheel could also be used to change the zoom level.

If find address is selected a dialog will be displayed where the user can enter the address they wish to locate.

5.2.3 Registration Process

When an Android app user completes the registration process the app sends the information to the web application via an HTTP Post. When this post is received by the web application it notifies its users that a new registration has been received. A back office emergency responder can then take over the registration process by selecting the new registration, verifying its information and completing the addition of the data to the database. The verification process includes the user organizing the registrants' needs or capabilities as required.

5.2.4 Send Message Process

The send message process begins with the entry of text to send or when emergency workers and/or volunteers are paired with residents' locations that they are to assist.

Text Messages: Text messages can be sent from the application to emergency workers, volunteers and registered residents. A menu item is selected or a button or other link is clicked on. The send message form appears. The user can then enter the desired text or copy text from a template. When the text is complete the user can then send it or cancel.

Pairing Emergency Workers, Volunteers and Residents: When a member of the emergency worker office staff pairs emergency workers, volunteers and residents a text message can be automatically generated informing all involved parties of it. This process lets the residents in need know that assistance has been assigned for them and who will be bringing the assistance. It also lets the emergency workers and volunteers know where they will be going and who they will be working with.

5.2.5 Read Message Process

When a message is received from a resident in need, emergency worker or volunteer a notification can be posted to the toolbar, a small notification could be displayed for a short while or some other method of making the operators aware that a new message has arrived. The operator can then click on the notification to read the message immediately or they can choose to complete their current task before reading the message. When an operator reads the message a note is made in the database that the message was read by the operator. The date/time will also be noted.

5.2.6 Request Location Process

If an operator or supervisor requires an update on the location of an emergency worker or volunteer they can select the request location menu item, link or button. A list of registered workers and volunteers will be displayed. The desired worker or volunteer is selected and then the request is sent to the Android app on that person's smartphone which responds automatically with its current position.

An additional method to request an updated position is to right click the worker's or volunteer's vehicle on the map which would cause a small menu to popup. This menu could include links to send a message, request position update, and so forth. The request is then sent as before.

5.2.7 Database Design

The design of the database is generic so that it could be used with most databases although the target database is an open source database such as MySQL. Inheritance is used in the database design as it avoids the duplication of data in multiple tables. Inheritance also makes the building of a new service on top of this web service an easier task. This is a part of the SOA of the system.

The details of the database design follow. Please see the UML diagram of the database design on the following page.

Tables

Person: Person is the base table for all types of persons including volunteers and residents. Person contains all of the common elements of a person such as their name (first and last),

preferred prefixes and suffixes, address, current location and phone number. GetLocation is a method in Person. It returns the current location of the person.
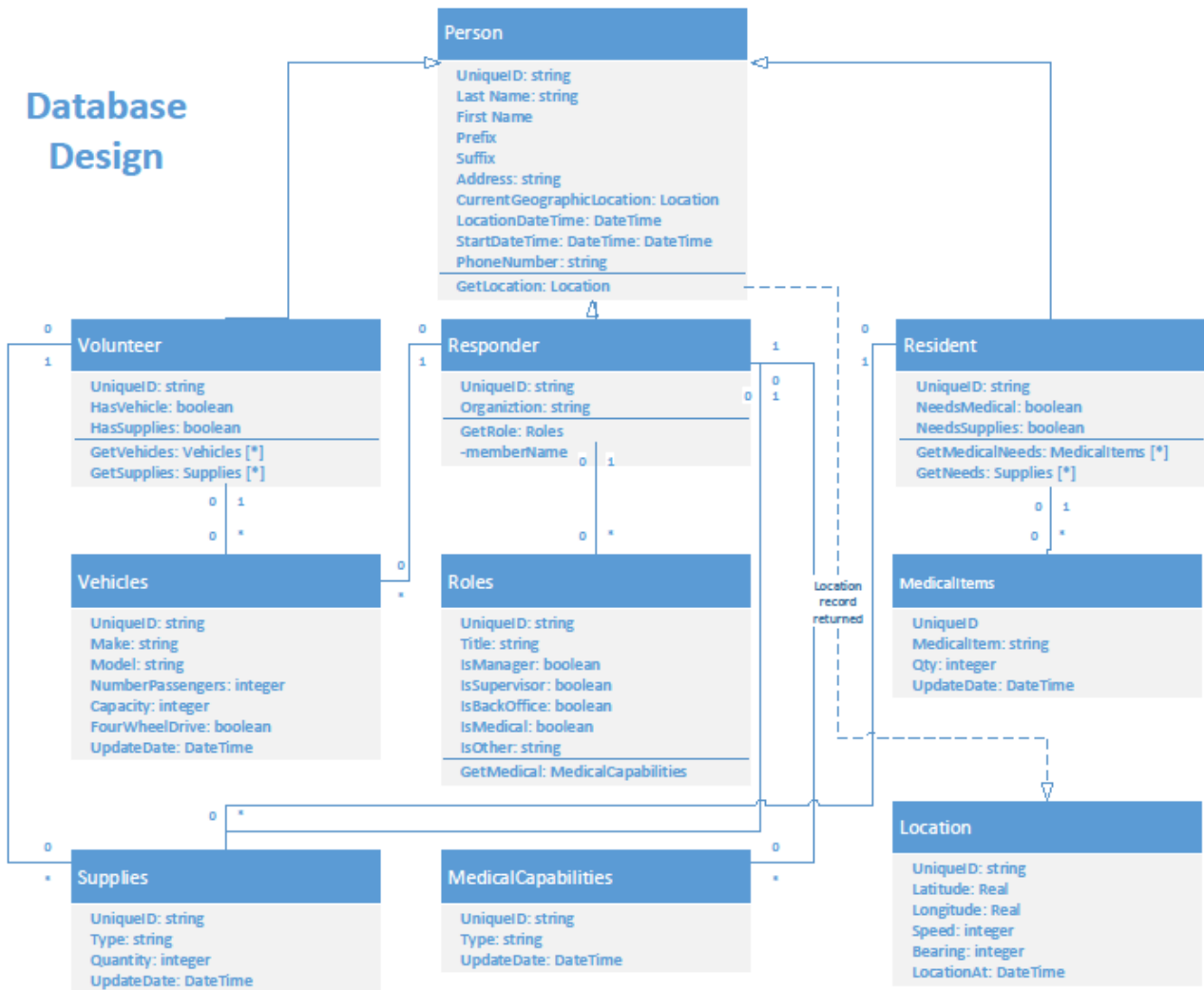

Database Design

Person
UniqueID: string
Last Name: string
First Name
Prefix
Suffix
Address: string
CurrentGeographicLocation: Location
LocationDateTime: DateTime
StartDateTime: DateTime: DateTime
PhoneNumber: string
GetLocation: Location

Volunteer
UniqueID: string
HasVehicle: boolean
HasSupplies: boolean
GetVehicles: Vehicles [*]
GetSupplies: Supplies [*]

Responder
UniqueID: string
Organiztion: string
GetRole: Roles
-memberName

Resident
UniqueID: string
NeedsMedical: boolean
NeedsSupplies: boolean
GetMedicalNeeds: MedicalItems [*]
GetNeeds: Supplies [*]

Vehicles
UniqueID: string
Make: string
Model: string
NumberPassengers: integer
Capacity: integer
FourWheelDrive: boolean
UpdateDate: DateTime

Roles
UniqueID: string
Title: string
IsManager: boolean
IsSupervisor: boolean
IsBackOffice: boolean
IsMedical: boolean
IsOther: string
GetMedical: MedicalCapabilities

MedicalItems
UniqueID
MedicalItem: string
Qty: integer
UpdateDate: DateTime

Location record returned

Supplies
UniqueID: string
Type: string
Quantity: integer
UpdateDate: DateTime

MedicalCapabilities
UniqueID: string
Type: string
UpdateDate: DateTime

Location
UniqueID: string
Latitude: Real
Longitude: Real
Speed: integer
Bearing: integer
LocationAt: DateTime

Figure 6: UML Diagram of the Database Design

Volunteer: Volunteer inherits all of the attributes and methods of Person. It extends Person with the addition of the attributes of HasVehicle and HasSupplies. It further extends Person with the additional methods of GetVehicle and GetSupplies. GetVehicle returns details about the Volunteer's vehicles. GetSupplies returns the details of the supplies that the Volunteer has. Both GetVehicle and GetSupplies can return multiple records just as a Volunteer may have more than one vehicle and more than one type of supply.

Responder: Responder inherits all of the attributes and methods of Person. It extends Person with the addition of the attribute of Organization and the method GetRoles. GetRoles returns the roles of the Responder. More than one role can be returned since Responders can hold different roles.

Resident: Resident inherits all of the attributes and methods of Person. It extends Person with the addition of the attributes of NeedsMedical and NeedsSupplies. Resident also adds the methods of GetMedicalNeeds and GetNeeds. GetMedicalNeeds returns the medical needs of the Resident. GetNeeds returns the supplies needed by the Resident. Both GetMedicalNeeds and GetNeeds can return multiple records just as a Resident may have more than one medical need and need more than one type of supply.

Vehicle: Vehicle is the record type returned from the method GetVehicles, in the Volunteer table. The Vehicle record contains the make, model, passenger capacity, cargo capacity, and four wheel drive capabilities.

Roles: Roles is the record type returned from a call to the Responder method GetRoles. The Roles record includes the Title of the Role, and Boolean values for IsManager, IsSupervisor, IsBackOffice, IsMedical and IsOther. It has a method of GetMedical which returns the medical capabilities.

MedicalItems: MedicalItems records are returned when a call to a Resident's GetMedicalNeeds method is made. MedicalItems includes the MedicalItem needed and the Quantity needed.

Supplies: The Supplies table contains a record of each supply item that a Volunteer has or that a Resident is in need of. The Supplies record has the Type and Quantity of the supply. When the Volunteer method of GetSupplies is called multiple SupplyItems records may be returned. When the Resident method of GetNeeds is invoked multiple SupplyItems records may be returned.

MedicalCapabilities: MedicalCapabilties is a table containing records of the medical capabilities of Responders. When a call is made to the Responder's GetMedical method multiple MedicalCapabilties may be returned.

Location: The Location record contains the latitude, longitude, speed, bearing and date/time of the location update. Location is returned when a call is made to a Person's GetLocation method.

5.2.8 Database Access Process

Since inheritance is designed into the database, multiple tables are accessed when some records are retrieved. For example, a retrieval of a Volunteer's info also invokes the reading of a record in the Person table. If HasVehicle is true it could also include a call to the Vehicles table. The same goes for HasSupplies, which may cause records from the Supplies table to be fetched.

6.0   Related VGI Projects

The largest number of projects that have used VGI are in the map creation arena. This can be in the form of placing points of interest, be they placemarks, incidents of violence or areas effected in a disaster situation, on an existing map or the actual creation of the map itself. There are some significant projects in this category such as OpenStreetMap and Ushahidi. Both of these were used during the Haitian earthquake to improve the quality of the map data available and to locate who needed help and where (Zook, et al.. 2010).

The map data created in the aftermath of the Haitian earthquake had these issues: duplication of efforts and problems encountered in the combining of data created within different software packages. These issues resulted in the creation of maps of Haiti that had varying degrees of coverage (Zook, et al. 2010).

6.1   OpenStreetMap

OpenStreetMap is a free and potentially current source of map data, in theory at least. Universally, the data is not up to date and it is not always free. The quality of the data is also suspect because the accuracy and completeness of the data provided and the credibility of the contributors is unknown (Koukoletsos).

6.2   Ushahidi

Ushahidi, created to map incidents of violence and peace efforts in Kenya (www.Ushahidi.com), was also used to create locations of various incidents during the Haitian earthquake. Incident information is submitted via SMS, MMS, e-mail, Twitter, or an online interface. These reports are then geo-tagged so their location is known (Zook, et al. 2010). This extra step was required because the incident information did not include its geographic location.

6.3   WikiMapia

WikiMapia is an online resource that is a combination of Google Maps and a wiki system for describing features (Wikipedia). Where OpenStreetMap has road networks as its centerpiece WikiMapia has polygonal features identified by users. For example, lakes, streams, neighborhoods, cities and mountains are all identified on the map. WikiMapia is privately owned and is not creative commons – the owners reserve the right to impose usage restrictions or fees at any time although it is currently free (Wikipedia).

7.0   Conclusion

In conclusion, a fleet management service that would allow for residents to make their locations and needs known and for volunteers to register their vehicles, capabilities and any surplus material they may have would be of a great benefit in an emergency. Emergency responders could bring order to the chaos that exists in an emergency situation. Volunteers could be marshalled and directed to assist others and to aid the emergency responders in reaching out to residents in need. Volunteers could assist other residents in need by delivering their surplus items to those that had great need of them. Residents would be more aware of when aid would be coming their way.

Since this solution requires the use of smartphones and the cellular system there is the possibility that this service could not be launched. The cellular infrastructure could also have been damaged in an emergency. However, if that is not the case, this service would be a blessing to the residents impacted by the crisis. They would *know* that their needs are registered and assistance is being directed to them. They would *know* when to expect this assistance. Volunteers would *know* where to go to best assist the emergency responders and fellow residents in the community. *Knowing* when assistance will be coming your way is much better than wondering when, even if, it ever will.

Using Free and Open Source Software for a fleet management service is important as it keeps the cost of the service low and there are no licensing issues. Volunteers will more readily take advantage of the service since there is no cost to them.

Creating the fleet management service using a Service Oriented Architecture makes development of the fleet management service a more manageable process as it can be built in pieces by

different developers. SOA also makes any possible customization of the service much easier. Only the portion that needs to be customized requires change so maintenance of the service is simpler.

Utilizing Web 2.0 the sharing of user-generated data is facilitated. After all, this is what Web 2.0 was made for.

RESTful services are used because they provide a simpler approach to the fleet management service. Since RESTful services use HTTP Post methods there are no unique methods to be concerned with. This makes the development of the fleet management service a simpler process. It make the service more cross-compatible to all of the different computing platforms that are available.

Appendix A: References

Goodchild, M. *Citizens as Sensors: the World of Volunteered Geography*, GeoJournal, 2007, 69, 211-221

Zook, Matthew; Graham, Mark; Shelton, Taylor; Gorman, Sean. *Volunteered Geographic Information and Crowdsourcing Disaster Relief: A Case Study of the Haitian Earthquake*. World Medical & Health Policy, 2010. Volume 2, Issue 2, Article 2.

Doodle. http://doodle.com/. Accessed on October 25, 2010.

Service-oriented architecture (SOA) definition. http://www.service-architecture.com/web-services/articles/service-oriented_architecture_soa_definition.html. Accessed on October 25, 2010.

Kodali , Raghu R., JavaWorld.com. What is service-oriented architecture? June 13, 2005. http://www.javaworld.com/javaworld/jw-06-2005/jw-0613-soa.html. Accessed on October 25, 2010.

Discovery Education. Web 2.0 Tools. http://web2010.discoveryeducation.com/web20tools.cfm. Accessed on October 30, 2010.

MySQL. http://www.mysql.com/. Retrieved on October 9, 2010.

PostgreSQL, http://www.postgresql.org/. Retrieved on October 9, 2010.

PostGIS, What is PostGIS?, http://postgis.refractions.net/. Retrieved on October 9, 2010.

OpenGIS Implementation Specification for Geographic information, http://www.opengeospatial.org/standards/sfs. Retrieved on October 9, 2010.

MapServer: Open Source Web Mapping, http://www.mapserver.org/. Retrieved on October 9, 2010.

OpenStreetMap. http://www.openstreetmap.org/. Accessed on October 30, 2010.

Raubal, Martin; Winter, Stephan. *A Spatio-Temporal Model Towards Ad-Hoc Collaborative Decision-Making*.

ESRI ArcUser Online, *The Role of Volunteered Geographic Information in a Postmodern GIS World: A conversation with Michael Goodchild*, April 2010. http://www.esri.com/news/arcuser/0410/vgi.html. Accessed on November 13, 2010.

Ushahidi, http://www.ushahidi.com/. Accessed on November 13, 2010.

Bregt, Arnold. *WORKSHOP: VGI for SDI? Voluntary geographic information for spatial data infrastructures?* April 16, 2010. http://www.grs.wur.nl/UK/Workshops/VGI_SDI/. Accessed on November 13, 2010.

Pultar, Edward; Raubal, Martin; Goodchild, Michael. Dynamic GIS Case Studies: Wildfire Evacuation and Volunteered Geographic Information. http://www.geog.ucsb.edu/~raubal/Publications/RefJournals/Pultar,Raubal,Cova,Goodchild-Dynamic GIS Case Studies-FinalPreprint.pdf. Accessed on November 13, 2010.

ESRI ArcUser Online, *The Next Step Extending the role of GIS*, June 2010.

http://www.esri.com/news/arcuser/0610/findanduse.html. Accessed on November 14, 2010.

Koukoletsos, Thomas. *Position Paper for the GIST Workshop on 'The Role of VGI in Advancing Science': VGI data quality: a dynamic way of assessing OpenStreetMap's accuracy.* http://www.ornl.gov/sci/gist/workshops/papers/Koukoletsos.pdf. Accessed on November 15, 2010.

WikiMapia. http://wikimapia.org. Accessed on November 17, 2010.

Janowicz, Krzysztof; Schade, Sven; Bröring , Arne; Keßler, Carsten; Maué, Patrick; Stasch, Christoph. *Semantic Enablement for Spatial Data Infrastructures.* http://www.personal.psu.edu/kuj13/janowicz_etal_tgis_semantic_enablement.pdf. Accessed on November 17, 2010.

Wikipedia, The Free Encyclopedia (2010). *Service Oriented Architecture.* http://en.wikipedia.org/wiki/Service-oriented_architecture. Accessed on October 25, 2010.

Wikipedia, The Free Encyclopedia (2010). *Representational State Transfer.* http://en.wikipedia.org/wiki/Representational_State_Transfer. Accessed on October 30, 2010.

Rodriguez, Alex. *RESTful Web services: The basics.* November 6, 2008. https://www.ibm.com/developerworks/webservices/library/ws-restful/. Accessed on October 30, 2010.

Wikipedia, The Free Encyclopedia (2010). *Uniform Resource Identifier.* http://en.wikipedia.org/wiki/Uniform_Resource_Identifier. Accessed on October 30, 2010.

Wikipedia, The Free Encyclopedia (2010). *Web 2.0.* http://en.wikipedia.org/wiki/Web_2.0. Accessed on October 30, 2010.

Wikipedia, The Free Encyclopedia (2010). *Spatial data infrastructure.* http://en.wikipedia.org/wiki/Spatial_data_infrastructure. Accessed on October 31, 2010.

Wikipedia, The Free Encyclopedia (2010). *WikiMapia.* http://en.wikipedia.org/wiki/WikiMapia. Accessed on November 17, 2010.

Fox News (2013). *Tornadoes, severe thunderstorms kill at least 6 in Midwest.* http://www.foxnews.com/weather/2013/11/17/midwest-under-high-storm-threat/. Accessed on November 27, 2013.

Wikipedia, The Free Encyclopedia (2010). *Hurricane Charley.* http://en.wikipedia.org/wiki/Hurricane_Charley. Accessed on December 13, 2013.

The OpenGTS™ Project (2013). Open GPS Tracking System. http://opengts.sourceforge.net/. Accessed on December 13, 2013.

Rainy Season in Borneo, Flash Floods, and Why You Should Always Drive A Pickup (2011). Kellenberger, Carrie. http://www.myseveralworlds.com/2011/03/31/rainy-season-in-borneo-flash-floods-and-why-you-should-always-drive-a-pickup/. Accessed on March 15, 2014.

Before and after the storm: The heartbreaking aerial photos that show the incredible scale of tornado destruction in Illinois town (2013). Daily Mail Reporter and Reuters Reporter. http://www.dailymail.co.uk/news/article-2509622/Washington-Illinois-tornado-Aerial-photos-incredible-scale-destruction-town.html. Accessed on April 18, 2014.

CHARLEY Graphics Archive, NOAA/ National Weather Service. Image is in the public domain. http://www.nhc.noaa.gov/archive/2004/CHARLEY_graphics.shtml. Accessed on April 18, 2014.

Wikipedia, The Free Encyclopedia (2014). *MySQL*. https://en.wikipedia.org/wiki/MySQL. Accessed on April 19, 2014.