INTEROPERABLE REQUEST AND STATUS MAPS FOR VGI-BASED SYSTEMS

ABDUSELAM MOHAMMED NUR
February, 2015

SUPERVISORS:
dr.ir. R.L.G. Lemmens
dr. F.O. Ostermann

INTEROPERABLE REQUEST AND STATUS MAPS FOR VGI-BASED SYSTEMS

ABDUSELAM MOHAMMED NUR
Enschede, The Netherlands, February, 2015

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.
Specialization: Geoinformatics

SUPERVISORS:
dr.ir. R.L.G. Lemmens
dr. F.O. Ostermann

THESIS ASSESSMENT BOARD:
prof.dr. M.J. Kraak (Chair)
J.J. Verplanke MSc (External Examiner, University of Twente ITC-PGM)
dr.ir. R.L.G. Lemmens
dr. F.O. Ostermann

ABSTRACT

Volunteered geographic information (VGI) comes to exist following the emergence of web 2.0 technology which transformed the role of users to create geographic data on the web besides consuming it. This type of data is becoming an alternative source of geo-information for many users in different application domains because of several factors including the free availability and timeliness of the data. Recently many Humanitarian organizations started exploiting VGI datasets to assist their activities for crisis response tasks. Several studies has been carried out how to benefit from this type of data by different interested groups. Unfortunately in most of VGI systems volunteers don't know much information about what data is required and the status of the mapping activities. The objective of this research is to design a method for an interoperable Request/Status map for VGI-based systems to increase the efficiency of VGI production by exploiting volunteers. This enables to bring uniform standards for various organizations how to request VGI dataset using request map and for volunteers how to get information about request of these organizations from their request map and also the status of the mapping process from the status map. A method was designed by evaluating three VGI systems namely OpenStreetMap, Geo-Wiki and Ushahidi selected based on topographic base mapping and crisis mapping use cases. The method was centred on evaluating and using OGC services to design interoperable R/S map systems. OGC WMS, SLD, WFS, and CSW services were identified as potential services for interoperable R/S map implementation. The implementation was carried out based on selected technologies according to the framework design. GeoNetwork and GeoServer were used as catalogue server and mapping server for the implementation respectively. To develop R/S map portal ExtJS, OpenLayers and GeoExt were used to design the user interface and incorporate mapping functionalities. Interoperability test was done by integrating sample R/S maps in Ushahidi and QGIS software. The interoperability test proved that it is promising to implement interoperable R/S maps through standards based OGC services but VGI systems should accept R/S maps concept and facilitate the integration of R/S maps with their systems by providing more improved features than they currently support. Finally the usability test indicated that still there is a need to provide more user-friendly user interfaces for R/S maps and this needs to be researched.

Keywords:
Request map, Status map, Interoperability, OGC, Geo-Web services, VGI, OSM, Ushahidi, Geo-Wiki

ACKNOWLEDGEMENTS

My utmost thanks is due to Allah (SWT), the most compassionate and most gracious granting me the chance and guidance throughout my study without which this work would not have been possible.

I thank my entire family who have comforted me through advice and prayer for their unconditional love. Most especially my mother, my wife and siblings.

I also thank my sponsor Nuffic for funding my studies in ITC. I acknowledge my employer, Hawassa University for granting me a study leave. My profound gratitude goes to my supervisors; dr.ir. R.L.G. Lemmens and dr. F.O. Ostermann for the immense support in supervising my work; reviewing, guiding and encouraging me throughout the study. This has improved my work and made it better. I therefore convey to you my warm appreciation. I thank all the staff of ITC particularly those within the Geoinformatics domain, my course mates, friends and brothers for the help given me in various ways.

My heartfelt appreciation goes to the members of the ITC Muslim community and the Ethiopian community, it has been a wonderful time being with you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Motivation and problem statement

Volunteered geographic information is a new paradigmatic shift in how geographic data is created and shared. This information is produced by novice and expert volunteers who are both users and producers of geoinformation at the same time. The basic technologies supporting the emergence of VGI are, (i) development of the web technology in allowing users to upload data besides consuming it, which is known as web 2.0 and (ii) georeferencing capability of several hand-held GPS devices and mobile phones enabled with GPS (Elwood, Goodchild, & Sui, 2012; Goodchild, 2007; Heipke, 2010). This form of geographic information production has important characteristics over the traditional way of producing geospatial information, in which national mapping agencies and corporate mapping organizations generate geospatial information by specialized mapping experts using sophisticated mapping technologies (Heipke, 2010). One of the advantages of this approach is the ability to involve a large group of citizens who know their local area more, the ability to produce timely information, and less cost investment. These VGI characteristics shifted the interest of many national mapping agencies to use this as source data to update their framework data. EuroSDR's workshop entitled "Crowd Sourcing for Updating National Database" is one of the prominent activities in this regard, in which researchers, national mapping & cadastre agencies and GIS professionals discussed how to update national database using crowd sourced geographic data (Heipke, 2010).

VGI system is defined by Fast & Rinner (2014) as a collection of users, technology infrastructures (software, hardware), which provides an environment for users to contribute geoinformation and it operates based on VGI data produced by volunteers using the facilities provided by VGI systems. There are different VGI based systems, each with different purpose and spatial scale (local, regional, global).The following are some of major VGI systems: (I) OpenStreetMap is one of the successful VGI based systems which works in producing digital geo-information throughout the world (Barron, Neis, & Zipf, 2014), (ii) Geo-Wiki is working at validating global land cover data to create more accurate global land cover map (Fritz et al., 2012), (iii) Google Map Maker is owned by Google, which allows user to create and update geoinformation online. Hence information becomes available in both Google Earth and Google Maps to be viewed by millions of users (Google, 2014), (iv) Ushahidi is developed to serve as a crisis communication platform, which facilitates reporting crisis incidents using SMS, email or website based form(Roche, Propeck-Zimmermann, & Mericskay, 2011).

Since the introduction of VGI, much research has been carried out concerning different aspects of VGI. Data quality, methods and techniques to analyse and use this new form of user generated geo-information are some of the challenging and hot research areas as highlighted by Elwood et al. (2012). Aside this prominent issues which affects reliability of VGI data, Heipke (2010) pointed out the need for production efficiency. For this purpose new methods should be devised which accelerates the way the crowd contributes geographic information. This could be by improving work flow and user interfaces. This research focuses to specific aspects of maximizing the efficiency of VGI production targeting on providing adequate information on the status of the data collection process (mapping process) and the required data details to be mapped in a particular area as well.

Unfortunately, the large group of volunteers in most of VGI systems are not contributing in an organized way. In most of VGI systems, there is no communication between the volunteers so that they can divide tasks, rather each of them produces their own contribution. They produce geo-information in the area they want to contribute, without even knowing that particular area is already done by some other volunteers. There could be more priority areas for mapping than this one only. Since there is no planned way to inform and direct participants about the progress, redundancy occurs and time and energy is wasted. Moreover, volunteers don't know much detailed information about the required data, which results in inconsistencies among the contributed VGI datasets.

A request map is a type of thematic map which gives detailed information for volunteers where VGI data is required by a particular organization and in what specification is the data needs to be produced. This specification can be determined by the requester, for instance: classification scheme for the data to be produced, the required quality level. A status map is a thematic map which informs volunteers about status of a mapping activity for a particular data request by requesters in VGI systems. Selected status variables such as completeness, accuracy defined by the requesting organization are used to determine the status. The status information helps volunteers to know the status of a certain area and plan where specifically and what type of data they should contribute within the request map extent. Whereas the request map helps them to decide for which request and where they should contribute in general and what type of data they should contribute. This has a great importance especially in case of disasters since timely and recent geo-information is required for effective disaster response tasks to effectively perform relief operation activities and helping the survivors with their need. Apparently, the integration of R/S maps helps to produce the required information in time by minimizing duplication and inconsistencies among the contributed dataset.

The need for R/S map is demonstrated in several instances; the following are some of them: (i) the US National Phenology Network (USA-NPN) receive a large number of emails from volunteers participating in the phenology observation, who want to know the priority areas where they should go and collect data. This clearly indicates that there is a lack of request information, (ii)As stated by Fritz et al. (2012), One of the future plan of Geo-Wiki is to incorporate feedback mechanisms to effectively communicate with users. This includes many things to be considered. Here the need for a status map is indirectly reflected since it plays a role as one of a feedback mechanism, (iii) due to flooding hazard in Pakistan in 2010, a request was sent to the Google Map Maker community to provide quality geo-information for the purpose of relief operation. The request was made for specific features of interest: like roads, hospitals and schools in a specific flooded area. In addition to this, the community was requested to label city and village names throughout the country (Team Google Map Makerpedia, 2014).

There are some experiences in using R/S maps in VGI systems; humanitarian OpenStreetMap uses "status map" web application to coordinate geoinformation collection by volunteers. The method employed for this application is limited to the OpenStreetMap environment; therefore, it is not compatible with other VGI systems. Being developed in the form of a web application, its accessibility is limited to the web environment; it can't be accessed from different platforms, for instance in desktop GIS applications.

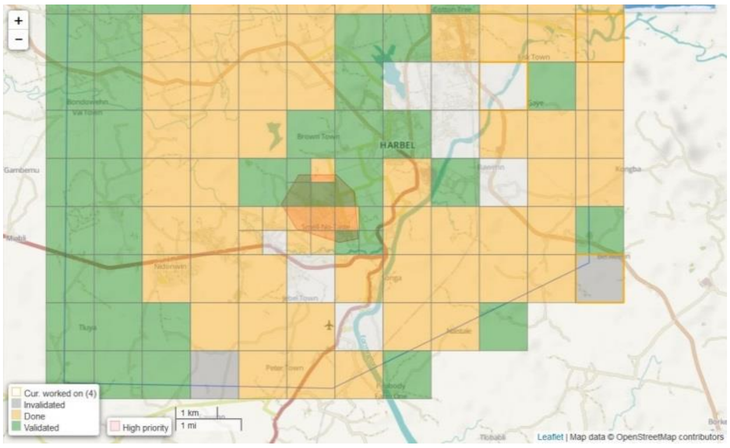

Figure 1: HOT tasking manager tool example (http://tasks.hotosm.org/project/625#task/53)

Currently there is no uniform way of handling R/S maps among different VGI systems. Using open standards based geo-web services plays an important role towards standardizing the way R/S maps are handled, and this has an advantage of interoperating R/S maps among different VGI systems and makes it accessible to other platforms as a discoverable services. This helps to have flexible access to R/S maps from the GIS environment the user prefers to work on. Being standardized also contributes towards common understanding and skill of using and producing R/S maps among participants and consumers. Therefore the objective of this study is to design a mechanism for publishing interoperable R/S maps in VGI-based systems.

1.2. Research identification

In this section the main objective of the research, its sub objectives and research questions to answer each sub objectives are identified.

1.2.1. Research Objectives

- Design and implement a method for publishing interoperable R/S maps in VGI-based systems.

1.2.2. Sub-objectives

1. Define request and status map characteristics and identify the requirements for various VGI systems' R/S maps.
2. Developing a method for user friendly and effective way of sharing and managing R/S maps for efficient management of VGI projects.
3. Identify and apply an effective way of communicating R/S maps to the intended audience.
4. Produce R/S maps which can be shareable among different VGI systems and GIS platforms.

1.2.3. Research questions

1.1. What are request maps and status maps and how do they relate to each other?
1.2. What types of R/S maps are relevant for different type of VGI-based activities?
2.1. What are appropriate data models for R/S maps?

2.2. How to handle R/S maps' multiplicity in source and time?
2.3. What are proper modes of operation for R/S maps (easy deployability Ownership, update frequency)
3.1. Who are the users of R/S map and how do they use them?
3.2. What are appropriate visualisations for R/S maps?
4.1. What are the criteria that make R/S maps interoperable?
4.2. Can the current OGC Web services qualify (support) for R/S maps handling? If not what needs to be extended?

1.3. Innovation aimed at

The novelty of this project is in three aspects, (i) a mechanism to publish a R/S maps for VGI based systems will be designed, which is not currently available in most of VGI systems, (ii) an interoperable nature of the R/S maps which can be communicated among different VGI based systems and GIS environments is also another novelty aspect, (iii) in general the R/S maps support of VGI systems contributes towards improving the meta-information on volunteered geographic information which is not given a lot of attention currently.

1.4. Related work

No research has been carried out to develop a method for interoperable R/S maps support in VGI systems so far.
In the following section we will discuss some projects, which are not published research documents, but which reflects the practise of R/S maps in VGI based systems.

Humanitarian OpenStreetMap (HOT) has developed the OpenStreetMap Tasking Manager Tool to coordinate geoinformation collection by volunteers, in areas with political crisis and disaster event. The objective of the tool is to coordinate volunteer mappers across the world to perform mapping activities in an organized way. This is accomplished by dividing the project area in to smaller grids (squares), which can be mapped rapidly by a volunteer. Once a grid is accessed by a volunteer it will stay locked till the volunteer completed the task and release it. The grids are labelled with different colours which indicate the status information of a particular grid: Done, Validated, Invalidated, and currently worked on (if somebody is currently working on it). This information helps volunteers to be informed about the status of the mapping activities in terms of completeness in different parts of the project area to avoid duplication and enhance productivity(HOT-OSM, 2014). The importance of this tool was well demonstrated in the Haiti earthquake where volunteers were able to produce the map of the city after the disaster within a couple of days (Ajmar, Boccardo, Tonolo, & Veloso, 2010). Some of the shortcomings of this tool are: it is designed to work with OpenStreetMap and it is not compatible with other VGI systems, for instance Geo-Wiki is a theme mapping VGI project which have its own requirements for R/S maps which cannot be completely addressed with this application. The tasking manager tool is only accessible through web browser by the URL address of the tool. We need to make it available as open standards based geo-web services so that it can be accessed from different GIS environments too besides the web. The "request maps" in the tasking manager tool assumes uniform priority of mapping within a single request boundary and the requests are defined based on requests made by organizations through emailing to HOT (Kate Chapman, personal communication, December 14, 2014). This research attempts to design a system, which allows request maps of area of interest to be uploaded as vector shape file by requesters. This enables requesters to prepare a request map partitioned in to different categories based on mapping priority attached as an attribute to each partition; which helps to prioritize their mapping request within the request map. It also provides functionalities for requesters to manage their R/S maps such as editing the geometry (extent) and priority attributes of R/S maps using R/S map application.

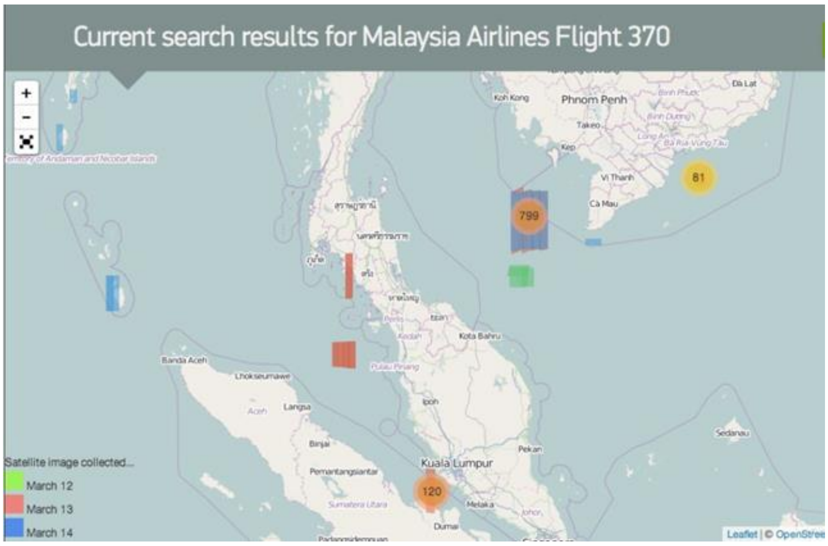

Figure 2: Tomnod Search results for flight MH370 (http://qz.com/188270/using-crowdsourcing-to-search-for-flight-mh-370-has-both-pluses-and-minuses/)

The other recent activity is a crowd-sourcing project to find the missing Malaysian airlines flight MH370 launched by Tomnod—a web application owned by Digital Globe commercial satellite Imaging Company. This project was aiming to involve the massive volunteer contribution in the search effort. Volunteers were expected to report any signs (wreck, life raft, oil slick, and other suspicious objects) of the missing airplane from high resolution satellite imagery uploaded by this company for the search area. For this project higher resolution images taken at different time were made available in different parts of the search area. The areas where there is image are represented with strips of different colour representing different date imagery availability (see Figure 2), so that the volunteers can go for that area and search. The results of the tag by volunteers was communicated by placing a point symbol of different colours labelled with the number of observations for the aggregated point observations. Besides, it also shows the extent in which this aggregated observations cover when a user clicks on the point(Meier, 2014; Tomnod, 2014). The visualization approach applied here is good in terms of aggregation of points which makes easy to get the status message. This way of representation helps in VGI systems which work on collecting point dataset (the obvious case alike base mapping VGI projects), like Geo-Wiki and US National Phenology Network (USA-NPN). This application suffers the same problems like the Tasking Manager Tool.

1.5. Method adopted

This project started by exploring VGI concepts and VGI systems, which includes: characteristic of VGI Systems and VGI data, understanding the various VGI systems working principle relevant to R/S maps. All these information helped us to understand our target system such as OpenStreetMap, Geo-Wiki and Ushahidi for which we designed interoperable R/S maps support. The VGI systems were selected based on topographic base mapping and crisis mapping use cases. In the topographic base mapping case the focus was to use VGI systems dedicated in base and theme mapping, OpenStreetMap and Geo-Wiki were the two VGI systems considered for this. In the crisis mapping case there is an immediate need for geoinformation in order to accomplish effective response tasks. Crisis mapping was perceived from two perspectives; from base and data mapping and updating and mapping crisis reports. For the first case we have considered OpenStreetMap and for the latter case Ushahidi platform was selected.

The concept of R/S maps were clearly defined; characteristics and requirements of R/S maps for the selected VGI systems were identified. Knowing all this information helped us to identify the requirements for the design.

OGC and ISO/TC211 are the two widely accepted and implemented geo-information standardization bodies. In this project we have used OGC-standards-based services to design a method for publishing interoperable R/S maps in different VGI-based systems. Being widely accepted and implemented nature of this standard contributed to facilitate interoperability. OGC services such as WMS, SLD, WFS, and CSW were selected based on the requirements to implement interoperable R/S maps.

The R/S maps are stored and managed in PostGIS spatial database as a separate feature (layer) for each request. This helps to store and manage the request data and update the dynamic nature of both the status and request maps.

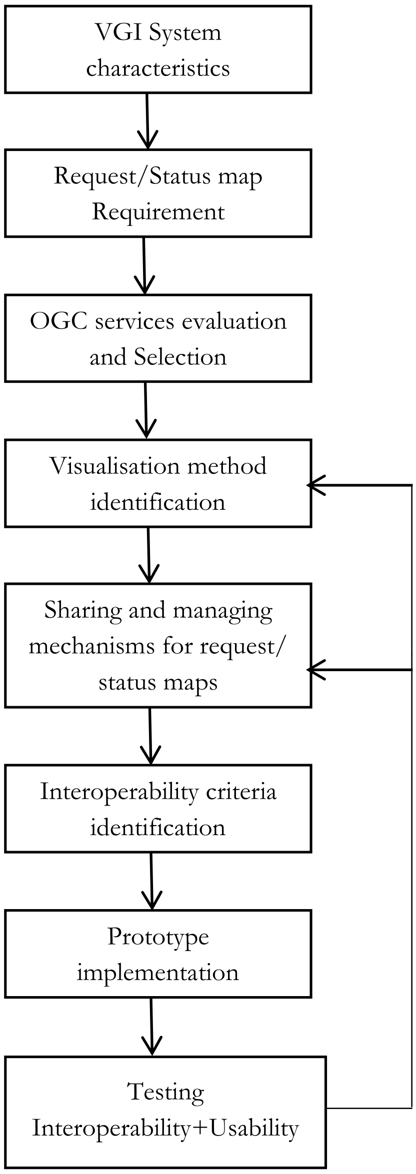

Figure 3: Flow diagram of the method

A visualization method which communicates both the request and status information in an effective way was identified. For this characteristics of the users of VGI systems, use case, the VGI system itself and the way the R/S maps going to be represented (separately or single R/S map) were considered, since we found them to be factors affecting the cartographic visualization method to use. The selected visualization techniques were applied as SLD file—created for different visualization methods—attached to the corresponding WMS representation of R/S maps to be able visualize them on the R/S map portal.

A prototype implementation was carried out based on the proposed framework in the design stage. R/S map portal prototype was implemented using various tools and technologies such as GeoNetwork and GeoServer to catalogue the metadata of R/S maps and create OGC compliant web services respectively. JavaScript libraries such as ExtJS, OpenLayers and GeoExt was used to design the portal interface and add mapping functionalities.

In the testing phase of the project interoperability and usability test was conducted for the R/S maps and the portal serving them. The R/S maps interoperability test with GIS software such as QGIS and VGI systems such as OSM and Ushahidi was accomplished based on interoperability criteria defined for this. Basic usability test was carried out. 7 random test persons were selected from ITC MSc. Students for the test. Each of them was interviewed in two rounds based on their interaction with the R/S map portal. The complete and thorough test can be carried out as a follow-up research.

2. VGI SYSTEMS AND PROCESS OF VGI PRODUCTION

2.1. What is VGI?

The term VGI is first introduced by Goodchild in 2007, which stands for Volunteered geographic information, a special case of user-generated content. It is special because the content deals specifically with geographic information produced by volunteers. The volunteers involved in the geographic information creation process are both experts and amateur citizens, often with little formal qualifications, who are mostly untrained (Goodchild, 2007).

As mentioned by Goodchild(2007) Web 2.0, Georeferencing, GPS, Geotags, and graphics are among the technologies that enabled VGI. Heipke (2010) highlighted two of them as basic technologies: Georeferencing capability of GPS and mobile phones and web 2.0 technologies including broadband communication. Web 2.0 plays an important role in the emergence of VGI. Traditionally, the user communication with web content was one-way, in which users can only consume contents (resources) available from the server. Through time, the web technology advances and it becomes possible for users to upload information (content) to the database on the server. This experience developed and transferred in to more sophisticated type of communication known as user-generated content. Websites in which the contents are fully created by volunteer users are emerged. The other important factor is the geo-referencing capability of GPS devices and mobile phones. These make users to produce and store their own geographic information of activities such as the place where they walk around, ride a bike at low cost (affordable price). Besides, web mapping interfaces, which provide high resolution satellite images, maps and tools, helped the volunteers to create geo-information for the area in which they are familiar with or interested to contribute regardless of knowing the place by tracing features on the screen using mouse clicks (Heipke, 2010).

2.2. VGI Systems

2.2.1. Ushahidi, crisis communication platform

The Ushahidi platform is an interactive map application devoted to crisis management (for example, political crisis, natural disasters and local conflicts). This tool allowed the online community to follow the progress of crisis from the reports made by those directly involved in real-time. The application covers both specific and time-bound events (for instance, the violence in South Africa, Congo, Kenya and Gaza strip; elections in India and Mexico; earthquake in Haiti, blizzard in the United States)(Roche et al., 2011).

The purpose of the Ushahidi application is twofold: at one hand it can be used for humanitarian organizations to get information on the condition of the disaster and on the other hand survivors can use it to make a report of the situation they are in to get help from others. In general, the application facilitates information sharing environment during crisis by sending any useful information anybody has by using SMS, email or web based forms, accessible from the website. Each submitted report has to be assembled, formalized, documented and checked to appear on the map so that to produce reliable information to get effective help from emergency response organizations (Roche et al., 2011).

The application integrates several pieces of technologies. It works based on the principles of mashups, varieties of web services from mapping, database, data handling tools and visual functionalities mashed up to provide better services. It is implemented as an open source API which follows the standards of both web and GeoWeb. It can be deployed for a particular purpose and adaptable to the requirements of that specific environment (Roche et al., 2011).

Haiti earthquake was one of the popular destructive crisis in which the Ushahidi platform was best used. The 4,636 Ushahidi project was deployed for the people to report near-real time information about the situation they are going through, and more than 3000 reports were received within two weeks(Roche et al., 2011).

The other crisis where the ushahidi contributed was the Christchurch earthquake in February 2011. Following the disaster the ushahidi application dedicated for this purpose was deployed to communicate useful information concerning the tragedy so that to be able to identify and report hazard areas, to request help and exchange public information (hazards and evacuation zones, infrastructures and road status) and accessible services (water, supplies, pharmacies and medical centres still open). 1,200 reports were received within 10 days (Roche et al., 2011).

These are not of course, the only experiences in which ushahidi helped in crisis response. The Haiti earth quake was a turning point for the consideration of ushahidi platform as an important tool in crisis management to assist the emergency response task (Roche et al., 2011).

2.2.2. OpenStreetMap

OpenStreetMap (OSM) is one of the successful VGI systems, which aims to create editable, free and openly accessible geographic information which covers the whole world. The contribution comes from the efforts of volunteers from different parts of the world. It has large number of diverse and passionate communities, which is growing day by day. This includes enthusiast mappers, GIS professionals, Engineers running the OSM servers, and humanitarians mapping disaster affected areas. The numbers of users is reported to be about 1.3 million In 2013 it has increased to 1.6 million in July 2014 ("OpenStreetMap," 2014, "OSM Community members," 2014).

OSM data comes from varieties of sources and various freely accessible tools facilitates the creation of geographic data from the scratch or import already existing external data using one of the freely accessible editing tools that suits the comforts of the user in terms of experience or preference. Java OpenStreetMap editor (JOSM), online flash editor, potlatch, web-based java script editor, and QGIS OSM plugin are the tools that allow loading of vector data from OpenStreetMap and edit features of interest and send the change back to the OpenStreetMap server. Spatial data can be collected using portable and GPS integrated devices in the case of classic approach. Features can also be created using Varieties of high resolution satellite images contributed from organizations such as Aerowest, Microsoft Bing or Yahoo for free provisionally and Aerial images as background layer to trace different types of features(Barron et al., 2014; "QGIS OSM Plugin," 2014).

The contributed data are stored in OSM database in line with OSM data model, in which point features are represented as "Nodes", linear features as "Ways", and polygon features as "closed Ways". In addition to this, features can be specified by one or more tags, many tags or no tags can be associated to each feature (Barron et al., 2014).

OSM data is being served in different ways for several purposes. This is because OSM data is becoming a serious geo-information source in different application domains, that can be consumed by many GIS software. The data is consumed and made available in different alternatives starting from serving as a base layer in varieties of web mapping applications to establishing a number of geo-web services around it, such as OpenRouteService (www.openrouteservice.org), OSM 3D (www.osm-3d.org), and OpenStreetBugs. Moreover, many OGC services are also implemented to publicize the available data. OSM WMS (www.osm-wms.de/) is one of these services that provides OSM data as WMS service for global coverage (Amelunxen, 2010; Auer & Zipf, 2010; Heipke, 2010).

Haiti earthquake was a major turning point which demonstrated the potential of OpenStreetMap. In the aftermath of the disaster, OSM based collaborative mapping platform devoted to Haiti was established. And detail digital geographic information of the capital Port-au-Prince including roads, bridges, damaged buildings, functioning infrastructures were produced by OpenStreetMap community within a small number of days. This was to support emergency response organizations need for timely and up-to-date geospatial data for the effectiveness of their task (Heipke, 2010; Liu, 2014; Roche et al., 2011).

2.2.3. Geo-Wiki

Global land cover is a basic terrestrial baseline dataset which have contributions in a number of different global, regional and national scale applications such as assessing forest and agricultural land resources, and as inputs to large scale economic land use and ecosystem models. Three versions of global land-cover products have been created during the last decade, namely GLC-2000, MODIS and GlobCover. But recent studies show as there is both spatial (difference in the type of land cover) and semantic (meaning of the legend) disagreement between these land cover maps especially in forest and cropland. Several reasons mentioned for this, one of them is lack of adequate in-situ data which is required to train, validate and calibrate land cover maps. Varieties of disagreement maps are analysed and produced which needs more in situ datasets to validate this disagreement (Fritz et al., 2012).

Geo-Wiki is a web 2.0 application for global land cover validation developed by Firtz et al. (2009), which integrates openly accessible high resolution satellite imagery of Google earth and crowd-sourcing to increase the amount of information available about land cover. This information contributes in training, cross-checking the calibration, and validation of land cover products. Soft validation approach is used in this application, which is based on validating land cover products using information from high resolution images like google earth images, geo-tagged photos and local knowledge (Fritz et al., 2012).

Creating hybrid land cover products is also the other objective of Geo-Wiki. This is accomplished by combining existing land cover products and fused crowd sourced data. This provides better accuracy than the accuracy obtained from any of the single land cover products (Fritz et al., 2012).

Geo-Wiki is implemented as a geoportal and complies with the OGC standards based geospatial portal reference architecture. It is founded based on service oriented architecture (SOA), in which functionalities of the application becomes discoverable service on a network. It uses web map services (WMS) to visualize image map representation of geographic data on top of Google Earth. In addition, it implements Web Feature Service (WFS) to serve crowd sourced vector raw data (validation dataset), and Web Coverage Services (WCS) to serve the hybrid map and disagreement maps raster data in different formats. Catalogue service for web (CSW) is also another implementation which is the core service in a geoportal which serves metadata information of all available Geo-Wiki products and complies with ISO 19115 standards (Fritz et al., 2012).

The initial implementation of Geo-Wiki was as a generic tool to crowd source land cover information. Later the demand from specific domains comes to benefit from the potentials of Geo-Wiki to develop branches of modules specifically devoted to a particular land cover type (e.g. agriculture, forestry). Variants of Geo-Wiki for agriculture, biomass, urban areas and human impact are among the implemented modules (Fritz et al., 2012).

Validation of disagreement maps can be done in several ways. The quickest way is to use random point validation tool, which assigns random locations on the disagreement map to be validated by the user. This option and two more buttons are no more existing in the current version of the application. So to do land cover assessment task, the user is expected to adequately zoom in to the place in the disagreement map where he/she wants to contribute. Then, he/she can select the button with crosshair symbol and click on specific place which shows the land cover map pixels outline in different colours, and provides a form which gives information about the land cover class assigned to that area in the three land cover maps to be validated by the user (Fritz et al., 2012). Still the way one user can contribute is either based on their local area or any random area they want to work on.

2.3. Interoperability

Interoperable R/S maps facilitate an integrated environment in the effort for crowd sourced geographic data collection by allowing VGI systems to interoperate their R/S maps. It also increases accessibility of the maps in different desktop GIS platforms. This contributes to the efficiency of VGI data production. Interoperability helps to enable communication among various heterogeneous systems to exchange data and functions irrespective of their differences. Often interoperability is facilitated by open standards that serve as common specification for all systems to adhere to the standard. The standard is independent of one specific technology. This is what makes possible the exchange of geo-information produced in one application with other application programs, which are compliant to that same standard. No matter the way the data is processed, the system on which the application program is running, and the output format of the data, they can still be communicated as long as both are adhered to the same standard(Mitchell, 2007). This has advantages in creating common understanding and practice of using and producing geo-resources; Sharing of geo-information becomes flexible and interoperable; this in turn makes data integration easy and escalates the accessibility and availability of geo-information. Interoperable organizations can create economically efficient and effective use of available geo-information, software and hardware systems used to process the data. Finally it leads to an integrated approach towards solving various ecological and humanitarian problems at a global scale (Albrecht, 1999).

"Geospatial encoding and service interface standards, especially open standards, are essential elements in the success of VGI policies, initiatives and business schemes". Applications involving VGI and crowd-sourced geographical data base their foundation on the capability to transfer location data and location (geo) information service queries and responses. For this to happen the data has to be encoded in a way that can be recognized by the receiving system. At the same time the interfaces used for the geo-web services needs to be specified so that the other systems can communicate with it based on the standard for both querying (sending request) and interpreting the responses. So open standards, geospatial encoding and service interface standards in particular play an important role towards the success in VGI policies, initiatives and business schemes (Reichardt, 2014).

OGC is an international non-profit organization which works to develop open standards for geospatial information to be able to interoperate heterogeneous systems in the geospatial world. Different organizations: companies, universities, governmental agencies participate in a consensus process(OGC, 2015). OGC plays a specific role in making geospatial information open and seamless on the web(OGC, 2011).

OGC has developed different Web service standards each with specific defined goals to address various interoperability challenges in different application domains. Web Map Service interface standard (WMS), Styled Layer Descriptor (SLD), Web Feature Service interface standard (WFS) and Catalogue Services for the web (CSW) plays an important role in a Geoportal—is a gateway to access geographic data and GI services using internet— implementations and other web mapping sites to visualize map, access raw geographic data, and catalogue geographic data and geoinformation services metadata (OGC, 2004; Rautenbach, Coetzee, & Iwaniak, 2013) . WMS provides open standards based HTTP interface for the exchange of geospatial information as georeferenced map images from one or more geospatial databases in a distributed computing environment over the web. A request—formulated based on the standard—can be send from a web client or a desktop applications and the response is a map image which can be displayed on the requesting application's interface (OGC, 2006). SLD is a profile of WMS that defines an encoding that extends the capability of WMS to support user-defined symbolization and colouring to display geographic features. This enables us to control the way how features are drawn on web mapping applications. It uses symbol encoding (SE) as a styling language—developed by OGC— to define presentational rules for applying styles for WMS layers (OGC, 2007b). WFS provides open standards based HTTP interface for requesting geospatial features (raw vector data) from distributed geospatial databases across the web. The response is in GML format, that is XML based data format developed by OGC for exchanging geographic features. The basic WFS service allows querying and retrieval of geospatial features from distributed databases whereas when editing such as creating, deleting and updating of features is required WFS-Transactional is used (OGC, 2010). CSW provides common interfaces to publish and discover metadata of geographic data, services and other important resources. Metadata of resources stored in catalogue represents characteristics of resources that can be consumed by a user or software. CSW has a number of profiles in which all of its implementations are required to give support for Catalogue Services Implementation Standard (CAT) (OGC, 2007a).

The importance of interoperability was demonstrated during Haiti earthquake and Philippines Typhoon Haiyan storm disaster to answer the demand for geospatial data access for producers to access the satellite images in their own GIS application environment to trace features, and for consumers to use the produced VGI data and other existing data to plan their relief activities. OGC WMS and WFS services are mainly used services which helped to increase the accessibility of the data for varieties of users and producers in crisis response (Ajmar et al., 2010; Harrison, 2010).

Standards based OGC services can be integrated with VGI systems to use the additional OGC service (mostly WMS) as a background or as an overlay layer which helps during editing. Open standards based services help to interoperate this background and overlay layers between VGI systems. It also integrates them in different platforms that can be used to access and contribute in VGI systems (both on the web and desktop). In this project R/S maps will be published as OGC standards based services which can be integrated to any VGI systems which are compliant to the standard so that it can be consumed by any other VGI systems and GIS software compliant to OGC services. This makes the R/S maps reusable. Once created, it can be served for different systems without the need to create different versions of the R/S maps for different systems.

Currently it is possible to incorporate WMS services in various editors of OpenStreetMap such as JOSM, online editor and potlatch. iD editor and Potlatch don't support WMS directly, instead they require custom URL provided by some applications such as MapWarper which converts the WMS URL to tile format URL so that it can be accessed in applications which supports tile based maps (OpenStreetMap Forum, 2013, 2015). In JOSM there is a direct support for WMS layers as a background layer to be used for editing(OpenStreetMap Wiki, 2014). Ushahidi's WMS plugin helps to incorporate WMS layer as an overlay in Ushahidi platform. Since Geo-wiki's design is based on OGC services (WMS, WFS, and CSW) it becomes easy to incorporate WMS services as layers.

2.4. VGI Process

As mentioned above, the basic technology for the emergence of VGI is web 2.0 technology. This technology allows accessibility of databases to be linked with websites through an API support, and this facilitates for users to create and upload their data to be accessible to the online community. In addition, interactive information sharing is also made possible by combining information from different sources, often with geospatial content like maps, images or videos. This facilitates collaborative mapping on the web. All these advances were founded on the principle of interoperability, which includes open standards in data formats and services, creating an independent, yet linked patchwork of a geodata infrastructure (Heipke, 2010).

The U.S. mapping science committee of National Research delivered a report concerning the definition of spatial data infrastructure. One of the main ideas introduced in the report was the concept of patchwork, which states that "national mapping agencies should no longer attempt to provide uniform coverage of the entire extent of the country, but instead should provide the standards and protocols under which numerous groups and individuals might create a composite coverage that would vary in scale and currency depending on need"(Goodchild, 2007).

Goodchild (2007) indicated the relationship of the national spatial data infrastructures (NSDI) model with VGI. He elaborated as individuals working in collaboration, yet performing independently towards a common goal to answer the requirements of the local community this together forms different pieces of a patchwork. The local need determines the updating frequency needed and the accuracy of the patchwork.

VGI data can be contributed by using different ways in which the VGI system provides: web based VGI system like OpenStreetMap, mobile applications like Geo-Wiki pictures and desktop applications like OpenStreetMap's QGIS OSM Plugin. The data contributed through all these means are sent to the server and stored in a central or federated database or the storage can be on the cloud based on principles of cloud computing. More information can be generated by integrating and processing several pieces of independent contribution of the crowd by the help of automated processing tools (Heipke, 2010).

Use of VGI data as an alternative spatial data source is becoming more popular in different domains. A number of researches are being done to investigate the potential use of VGI in a particular filed such as National mapping agencies. For instance the case of EuroSDR's workshop to investigate how to use VGI dataset to update national framework database is the most prominent one. This is because world mapping has been in decline for several decades. For example the U.S. Geological survey and national mapping agencies of many developing countries are no longer capable of accomplishing their regular-bases map updating task. Several reasons are mentioned for this, and one of them is lack of readiness by governments to invest on this high cost demanding project (Ariffin, Solemon, Anwar, Din, & Azmi, 2014; Goodchild, 2007; Heipke, 2010).

The Haiti earthquake was one of the foremost incident which changed the attitude of many organizations towards the potential of crowd sourced geospatial data. Red Cross organization became aware of benefits of VGI data for cost minimization to accomplish projects, and starts to use them globally in different projects, and promised any data to be collected by the Red Cross afterwards will be released under open data license (Meyer, 2013).

When organizations with diverse interest who wants to benefit from the VGI domain increases, better management mechanisms to facilitate the VGI systems to produce data that fits the demands of varieties of domains by exploiting volunteers needs to be devised.

The humanitarian tasking manager tool is developed by Humanitarian OpenStreetMap (HOT), for the purpose of solving coordination problems of volunteers contributing from different parts of the world. It allocates tasks by partitioning the whole project area in to several pieces of small grids which can be quickly accomplished by volunteers. Of course, this is one step ahead of solving the problem but the approach is confined with OpenStreetMap application and is not interoperable with other VGI systems. Its focus is also on mapping (political crisis, natural disasters, violence) under situations which deals with specific type of users (requesters) who needs geoinformation for their activities. In general, it is all about coordination to facilitate the mapping activity to map a particular area quickly based on requests made by humanitarian organizations ("HOT Tasking Manager," 2014, "OSM Tasking Manager," 2014).

In Geo-wiki there are different types of layers for a user to choose from, and one of this layer is a layer which shows the amount of contribution of validation dataset by volunteers in different areas which is visualized as pieces of grid cells of red, yellow and green colours covering the whole area. Each colour represents different information, red for no data, yellow for few data and green for some data contribution (Fritz et al., 2012). Perhaps this is good to give information for users about the overview of the contribution process, but it may not tell priority areas for contribution. Since areas with no contribution don't necessarily mean priority area, there might be an area in which more information is required by end users. In this case the end-user's requirement needs to be known to help the volunteers to focus for a particular area with clearly defined goal (requirement).

As explained above by Goodchild on his view of NSDI model fits VGI, one important point is highlighted, which is "local need" which determines the accuracy and the updating frequency needed for the patchwork. The question here is how to communicate this need. It is apparent that some means of communicating this requirement which can inform more volunteers about the need is essential.

This research looks the purpose of R/S maps beyond coordination to complete the unmapped areas as quick as possible. It will investigate various possibilities of R/S maps in meeting the increasing demands of VGI dataset for the diverse community specifically for the selected use cases. Therefore it designs a method for interoperable R/S maps for VGI systems.

3. REQUEST/STATUS MAPS

3.1. R/S map concepts

Request maps are thematic maps that shows an area in which data is required to be produced by volunteer mappers in VGI systems based on a particular specification determined by a requester. Request maps contain simple geographic features (such as polygons or grid cells), representing the areas to be mapped. These geographic features carry attributes on priority, classification needed, etc. It gives volunteers the required information about what is needed and how it should be produced. This in turn helps to produce data which meets the requirement of a certain application domain for VGI datasets.

Request maps can have different kinds of attribute information depending on the type of VGI system and the application domain (use case) for which the data is required. This attribute information influences characteristics of the required data. For instance, a request to update the base map of a particular area under crisis condition will have more emphasis for availability of data than accuracy; quality will be compromised to some extent, because in this case 'any information is better than no information' (Ajmar et al., 2010). In addition the priority level of this kind of mapping activity is high, so that timely data can be produced. Whereas a request made by a mapping agency of for instance country "x" might focus on accuracy and completeness variables to be addressed by mappers. The priority level of this type of request is relatively low as compared to crisis mapping requests.

A status map is a thematic map that shows status of the mapping activity within the requested extent in terms of the status attribute which is defined and described in the request map. Each particular place in the map will have associated status information that reflects the situation of the mapping activity. For instance, the status map can show the accuracy of the mapped data at different places within the request map extent. Status for a particular area is determined based on the situation of the VGI data within each status boundary. So each status information is associated to some particular place. Each status map can show one attribute at a time, for example completeness or accuracy. Yet it can have as many attributes as possible depending on the characteristics of the request to address different issues to be associated with the data. The user can select a status map with one of these attributes.

Besides, a status map is not only limited in informing the user to the current status of the mapping activity. It also provides the user (consumer) with the required information to consume VGI datasets. For instance accuracy and completeness information in different parts of the area helps to decide which part of the dataset fits ones context of use for a particular project (activity).

From the perspective of a consumer, the status map can provide an information that helps to know the status in terms of completeness (data availability) of an area. Meanwhile, requesters can use this information to plan where to request data in case where there is inadequate data for a particular area they want to get data. It also helps to plan what data is already available in the VGI systems and what they need to produce in case of integrating VGI datasets with their local owned dataset.

Request and status maps complement one another, when the status of a particular area changes it affects the request attribute, for instance priority attribute of the request map will be affected. The status boundary which falls within the high priority area of the request map will have high mapping priority but through time when those areas (status boundary) in that priority area becomes completely mapped the priority attribute changes again.

3.2. R/S map variables

3.2.1. Spatial data quality elements

Data quality issue is one of the important factors of geographic data, the case becomes critical, especially in the process of production, assessment and exchange of geographic data. In VGI systems there is no restriction applied on the contribution process of a user to control the quality of the data. This in turn affects the quality of the data. Quality in the context of VGI can be defined as "fitness for purpose" since the quality depends on the use case the data is going to be used. Four types of spatial data quality elements defined by ISO are mentioned in the work of Barron et al., (2014), which are "completeness", "logical consistency", "positional accuracy" and "thematic accuracy".

Several scientific studies have been carried out to assess the quality of VGI data and different methods are proposed. It can be categorized in general as extrinsic approach, which is based on comparing the VGI data with authoritative reference datasets. The work by Girres & Touya (2010) to assess the quality of the French OpenStreetMap dataset is based on such an approach. The second approach which is recently proposed by Barron et al. (2014) is the intrinsic approach, which is basically depending solely on the history of the VGI dataset without the need to use reference dataset. This makes it possible to make a relative statement about the quality of the data.

Quality can be one of the attributes in a status map, which provides the user with the current quality level of the dataset in a particular area. This information has a twofold advantage, firstly it informs the user the overall quality level of the dataset, and secondly it gives information about which particular feature in the VGI dataset is more accurate relatively to the other area, so that the user can prioritize their contribution.

Completeness is one of the elements of spatial data quality as mentioned in ISO standard. It tells us about the coverage of a particular area for VGI dataset both at spatial and thematic level, how much completed is a particular area in terms of its spatial coverage (the mapped features in the area) and attribute coverage, how many of the mapped features attribute is completely recorded. Completeness has an advantage both for producers and consumers in a VGI system. In case of producers (contributors) it gives information about the status of completeness: about completed and uncompleted areas, so that they can focus and work on the area where there is less or no data. Its importance becomes immense especially during crisis mapping where immediate access for geo-information is needed by speeding up the contribution process by avoiding redundancy of contributions for the same area. From the perspectives of a consumer it gives information to inform a consumer about the completeness status of an area.

The positional and thematic accuracy of a dataset directly affects its usability for a particular application, for instance routing and navigation application rely on accurate road network dataset so that they can give reliable service for their customers. Besides, many location based services rely on buildings attribute such as house number. The accuracy of this attribute affects providing accurate geocoding services at house number level precision.

This project follows an approach that tries to determine the value of status map's attributes to be automatically calculated. Unlike HOT (humanitarian OpenStreetMap) tasking manger tool which defines the status of each pieces of tasks (grids) manually by the volunteers assessment and confirmation of the contributed data (Kate Chapman, personal communication, December 14, 2014). The fact that many researches has been done to automate the spatial data quality assessment in a VGI systems can contribute to this effort. The intrinsic approach for data quality assessment proposed by Barron et al. (2014) is the one which fits this approach amongst others. But this is not the scope of this project, but it can be considered for future improvement of the R/S maps.

3.2.2. Classification scheme

Currently satellite images, specifically high resolution satellite images are becoming the source for most of land use/land cover products. In the land use/land cover map production process, the identification and delineation of the different features is guided by the selected land cover classification systems, that defines the characteristics of each land cover category of the classification scheme. Most of VGI systems are operating based on high resolution satellite imageries that needs to be interpreted by volunteer mappers, and this needs to define and inform volunteers the classification system that needs to be considered for a given requested area. The difference in classification systems followed by volunteers in order to interpret high resolution satellite imagery in Geo-Wiki could affect the usability of the final validation data set outcome by end users. This is because most of volunteers are unaware of the semantic of land cover types in different land cover classification systems (LCCS), which is not also mentioned in VGI systems so that the users become aware of the characteristics of the land cover types he/she wants to contribute. In order Different organizations to benefit from this kind of VGI dataset, the data needs to fit with their classification system of interest that can be mentioned as an attribute in their request map.

There are various land cover classification systems standards developed by different organizations such as Anderson Classification System, CORINE Land Cover, IGBP DisCover, UMd Legend (Herold, Hubald, & Di Gregorio, 2009). Each with different and some common characteristics in their hierarchy level, detail of each level and definition of categories in each hierarchy. The scope of the classification system can be global, regional, and countrywide. A number of harmonization tasks between different classification systems are done at different level (Anderson, Hardy, Roach, Witmer, & Peck, 1976; Herold et al., 2009). Anderson et al. (1976) revised the U.S Geological Survey's classification work to develop National level classification systems which helps federal and state agencies to have consistent land cover datasets throughout the country. The system defined two hierarchy levels: level 1 and 2, and left the remaining hierarchy which is level 3 and 4 to be defined in the way which keeps the specific requirements of local agencies. The system can also be adopted by several other countries. UN came up with a global land cover classification systems which helps to harmonize global level land cover datasets and has an advantage when working at large scale projects to integrate and even compare between different land cover datasets. It developed a translation from the aforementioned classification systems to UN Global Land cover classification systems. Nevertheless there are also state-wide and local level systems as mentioned in the case of U.S Geological Survey revised work above, which better addresses the requirement of that particular area (Herold et al., 2009).

According to this research a number of request maps can be requested by different organizations each with its own attributes and descriptions of the request. A classification scheme (system) is one of the attributes of request maps especially in case of land use/land cover mapping, which is required in different application domains at different governmental level (Federal, State, Local) for specific purposes like water resource inventory, environmental impact assessment, national policy formulation (Anderson et al., 1976). So, producing geo-information (land use/land cover map) based on a particular classification system requested by a specific organization enables organizations to get consistent dataset from VGI projects which meets the organization's classification scheme standard. Currently this option is not implemented in most of VGI projects.

3.3. Types of R/S maps

There are different types of R/S maps based on the type of VGI system they are going to be used. Besides the VGI system, the use case type can also determines what type of R/S maps to use. There can be different varieties of R/S maps for a single VGI system depending on the use case characteristics in addition to the VGI system. The difference can be in various aspects such as the R/S map variables required, representation of the R/S map, updating frequency, etc. Table 1 (see Section 6.1) illustrates the characteristics of R/S maps required in three different VGI systems namely OpenStreetMap, Geo-Wiki and Ushahidi considering two different use cases. OSM was considered two times since it can be used in both use case scenarios in reality under the topographic base mapping and crisis mapping use case. For instance in Ushahidi we can use two different varieties of R/S maps taking the following two scenarios in to consideration. The first scenario is that, when we want damage assessment information from the volunteer community in ushahidi, we can use a R/S map which is represented as vector grid map in which each grid indicates the status of damage assessment level (no assessment, less assessment, medium assessment, and high assessment) so that volunteers contribute in another place where there is no assessment yet. The other scenario is that when we need crisis information (reports) to be reported by volunteers. In this case a simple polygon map which shows the extent of where information is required is enough to guide volunteers where they should contribute. The clustered representation of reports in ushahidi can be used as a status map to know the hotspot areas for prioritizing help for organizations who need this information for relief operation. So considering both the VGI system and the use case characteristics helps to decide the type of R/S map that fits to the situation.

A request map can be of two types based on request map attribute and this in turn affects what type of R/S maps is required. The first type is a request map with different attribute values in different parts of the request boundary, this is the case when the whole request area have not the same priority attribute value. Different areas within the request map will be associated with specific type of priority value, be it high, medium or low. This type of request map can be represented with a polygon divided in to different categories of priority areas, and each priority area (polygon) will have its own priority attribute value. The second type of request map is a request in which all places within the requested area are equally important, no place is more important than the other. In this case a polygon map without partition (priority area categories) is enough to represent the request area and there is no need of attribute also here since all have the same value of priority.

The first type of request map fits more the case of crisis mapping scenario, for instance request for topographic base mapping in the aftermath of earthquake disaster, the organizations who requests VGI dataset needs data in some particular place more quickly than other places within the request boundary, so they will have to delineate the request map in to different partitions and assign an attribute value to each that tells the priority of each partition. The second type of request map can fit the case of normal topographic base mapping in which a national mapping agency requested to map buildings and roads of a particular place but with all places within the request boundary have equal priority value. But there are also times where the first type of request map will be more important for requests by national mapping agencies. For instance the agency might want to make a request to update the map of a particular place but each with different degrees of update requirement: areas with high, medium, or low update level requirement. In this case apparently the first type of request map comes to fit the purpose.

Table 1: R/S maps Use case comparison

<table>
<tr><th colspan="8">Use cases & VGI systems</th></tr>
<tr><td colspan="4">Topographic base and theme mapping</td><td colspan="4">Crisis Mapping</td></tr>
<tr><td colspan="2">OSM</td><td colspan="2">Geo-Wiki</td><td colspan="2">Ushahidi (Crisis reporting and damage assessment)</td><td colspan="2">OSM (topographic base and data mapping)</td></tr>
<tr><th colspan="8">Attribute</th></tr>
<tr><td>Request</td><td>Status</td><td>Request</td><td>Status</td><td>Request</td><td>Status</td><td>Request</td><td>Status</td></tr>
<tr><td>Priority-level</td><td>Completeness (spatial and thematic)
Attribute-Accuracy
Positional-accuracy</td><td>Priority level
Classification-scheme</td><td>completeness
Thematic-accuracy</td><td>Priority-level</td><td>Hot spot areas (clustering)
Completeness</td><td>Priority-level</td><td>Completeness</td></tr>
<tr><th colspan="8">Request description (metadata elements)</th></tr>
<tr><td colspan="2">Title
Description
Contact-address
Owner
Date
Purpose
Data request</td><td colspan="2">Title
Description
Contact address
Owner
Date
Purpose
Data request</td><td colspan="2">Title
Description
Contact-address
Owner
Date
Purpose
Data request</td><td colspan="2">Title
Description
Contact-address
Owner
Date
Purpose
Data request</td></tr>
<tr><th colspan="8">RS map Data type</th></tr>
<tr><td colspan="2">Vector</td><td colspan="2">Vector</td><td colspan="2">Vector</td><td colspan="2">Vector</td></tr>
<tr><th colspan="8">Representation (visualization)</th></tr>
<tr><td colspan="2">Vector grid map.</td><td colspan="2">Point density vector grid map</td><td colspan="2">Request map boundary (polygon) for the request and clustered (aggregated) point representation for the status map.</td><td colspan="2">Vector grid map.</td></tr>
<tr><th colspan="8">Management</th></tr>
<tr><td colspan="2">Catalogue server with search capability and listing R/S maps. Both options provided on the interface of VGI system for the user to choose from.</td><td colspan="2">Catalogue server with search capability and listing R/S maps. Both options provided on the interface of VGI system for the user to choose from.</td><td colspan="2">Catalogue server with search capability and listing R/S maps. Both options provided on the interface of VGI system for the user to choose from.</td><td colspan="2">Catalogue server with search capability and listing R/S maps. Both options provided on the interface of VGI system for the user to choose from.</td></tr>
<tr><th colspan="8">VGI data duplication level</th></tr>
<tr><td colspan="2">Medium, to be determined by the number of editing applied on the features.</td><td colspan="2">Low it depends on the confidence level supplied by the user while supplying.</td><td colspan="2">No duplication is required for timely GI creation</td><td colspan="2">no duplication is required for timely GI creation</td></tr>
<tr><th colspan="8">Updating frequency</th></tr>
<tr><td colspan="2">Low, because of the process of data collection is relatively low.</td><td colspan="2">Medium</td><td colspan="2">High, in order to communicate the frequent information change</td><td colspan="2">High, in order to communicate the frequent information change</td></tr>
</table>

The visualization for both cases will also be different. The first type will be represented with different colours for each priority type in which higher priority areas will be represented with more outstanding colours since they are more important than others. Within each category we will have a colour that ranges from light to dark to show the status of each grid (status boundary) in the status map within each priority area, but this colour will be the same as the one which is assigned to the priority boundary (see Figure 7, Figure 8, Figure 9).

For a request map without any priority area which affects the priority of the mapping process, can be represented with single colour for the whole request area and the status maps grid will be represented with that same colour that ranges from light to dark.

Through time the request priority attribute can change. The change can be for a priority area boundary as a whole within the request map or for some of the grid cells within different priority boundary. This could be affected by the status of the mapping process in different parts of the request map. An area which was high priority becomes low priority since the majority of the mapping work was done in that area, the medium becomes high and the low becomes high priority at the same time. This is going to be managed by the requesters when they believe the priority of a particular place should be changed they can update the attribute of the request map, from that moment on the area will have a different priority value.

We can have request and status maps represented separately or together, it depends on the use case and manageability (easy to manage and use). The situation where request and status maps needs to be separately represented can be affected by the use case and also the VGI system characteristics. For instance in the above case where we have different priority areas within the request map, we can represent the request map as a separate map in which each priority polygon is visualized with different colour that shows the priority difference, and the status map will be another separate map which shows the status of the mapping activity within the request area. But this type of separation between the request and the status maps has also its own disadvantage. The user has to manage which status map is for which request map. This can be solved by representing them as a single map, by combining both maps in to one, so the user don't need to search for the matching status map for a certain request map. The second case where both maps becomes separate depends on the VGI system characteristics. One of the VGI system characteristics is its visualization approach. When the visualization approach of the VGI system itself can reflect the status map information both maps becomes separately represented. This is true for example in Ushahidi, but it is not always true it depends on the use case. If the purpose is to guide volunteers where they should report crisis incidents in Ushahidi, we can publish a request map which shows the request boundary and describes what is needed, but for the status map the clustered representation of the VGI system itself can reflect the status map information. In this case the user is expected to manage only the request map because the status map is always there by the VGI system.

3.4. Users of R/S maps

Generally users of R/S maps can be classified in to two broad categories: users (requesters) who make requests for VGI dataset production in some particular places and volunteer users who uses the R/S map to decide where they should contribute. Specific users under the category of requesters differ from use case to use case, according to our use case in this project WFP, Red Cross, and national mapping agencies fall under this category. The second type of users are volunteers who are producing VGI data in different VGI systems. This category contains varieties of users with different levels of expertise which can be classified in to novice and expert level volunteer users. Both requesters and volunteer types of users use the R/S map differently. Requesters need simple ways of publishing and managing their request maps to publish their request with convenience. Volunteers need also simple user interfaces plus easy to understand representation of R/S maps so that they can easily get R/S maps of their interest and also get the correct perception of what is needed from the portrayal of the R/S maps to do the correct contribution type requested by the requesters.

3.5. R/S map use case description

3.5.1. Crisis mapping

Geographic information has an important role in all phases of crisis management cycle: mitigation and prevention, preparedness, response, recovery). The characteristics and demand of geographic information differs in each phase of crisis management. In the response phase there is an immediate need for geographic information by different bodies (emergency respondents, NGO, etc.) in order to support the response activities in a more effective way (Roche et al., 2011).

The power of Geoweb 2.0— that facilitates geographic data and applications to be used over the World Wide Web which is based on features of web 2.0 technology— helps to crowdsource geographic information and processing tools required together with open source software and standards leads to achieve different crisis communication platforms. This enables different bodies to be able to report about the crisis situation and also create and update base maps and data to answer the geo-information demand by different stakeholders in the crisis response (Roche et al., 2011).

We can see crisis mapping from two perspectives: first when we consider it as a contributing platform, which is the case in Ushahidi, where people report real-time information about the ongoing crisis situation; this helps emergency respondents and NGOs to effectively contribute to the response activity. The second case is from the perspectives of collaborative platform, which is the case of OSM, in which geo-information is crowdsourced by collaborating between different people around the world to create and update base map and data for timely provision of geo-information to facilitate the response activity.

The status map characteristics for OSM under crisis mapping and normal topographic base mapping situation will be different, since both serves different purpose (see). For instance, in creating and updating a base map and data under a crisis situation, completeness is more important attribute than quality, whereas in normal topographic base mapping for national mapping agencies, completeness, positional accuracy, temporal accuracy and other spatial data quality elements are more important.

3.5.2. Topographic base mapping

Topographic base map can be defined as detailed, and accurate diagrammatic representation of features on the surface of the earth. These features include: cultural, hydrography, relief and vegetation. It has a number of uses in different application domains such as by government and industry for urban planning, mining, emergency management and recreational purposes like traveling (Wikipedia contributors, 2015).

In the past decades the source of this kind of maps were merely produced by professional mapping organizations (authoritative data) in which spatial data were produced by professional mapping experts using GPS and survey measurements. But due to the very high cost involved in this method many national mapping and cadastral agencies are not capable of updating their base map data regularly and also producing detail geographic information in some other places. The budget for mapping and updating process is shrinking and governments are not interested to fund huge costs on such activities(Ariffin et al., 2014; Goodchild, 2007).

The emergence of web 2.0 technologies contributed a big role for the introduction of VGI systems which provides free geographic information produced by volunteers as an alternative. OpenStreetMap is one of the VGI systems among others which creates world base map for free produced by efforts of volunteers worldwide (Goodchild, 2007).

VGI is now getting acceptance worldwide as a reasonable means to collect and update geographic information using local expert's geographic knowledge about their immediate surroundings. With VGI there is a possibility for organizations to exploit volunteers at low cost in addition to using volunteer's contribution to improve the geoinformation production and updating process. Although quality of VGI data is still main concern in using VGI datasets, volunteers are most of the time contributing geoinformation about their immediate surrounding which may contribute to get good quality information (Ariffin et al., 2014).

Both governmental and commercial organizations are shifting from conventional mapping to this new alternative sources for geoinformation. The dataset can be used to update their map and enrich the attributes of selected features. OSM, TomTom, and NAVTEQ all benefits from VGI to keep their databases up-to-date. In Australia, Victoria State the updating process of state-level mapping of features and attributes is open for Volunteers—registered government employees. Google, TomTom, and NAVTEQ discovered that Government mapping organizations can exploit the potential that exists in web 2.0, new media and voluntarism for their change detection and geospatial data updating process (Coleman, 2013).

Humanitarian organizations such as Red Cross, WFP have started using VGI datasets to get timely, up-to-date and detail geoinformation of the affected area to facilitate their works effectively. Highly detailed base map which shows individual buildings, roads and other features were produced in OSM by volunteers to support the humanitarian organizations relief operation effort, to plan their activities and effectively support the rescue and humanitarian assistance activities during the Haiti earthquake and typhoon Haiyan storm — which struck Philippines (Ajmar et al., 2010; Meyer, 2013; Roche et al., 2011).

3.6. R/S map Use cases

3.6.1. OSM topographic base mapping

Many national mapping agencies are no longer able to update their framework database and started to see VGI as potential alternative for updating base map data and elaboration of features. It is obvious that different organizations have their own specific requirements to get from VGI data, such as the area where they need data and also what type of data is required in that particular area are different from organization to organization. This requires a way how these national mapping agencies can exploit volunteers to benefit the most out of VGI data contribution by letting volunteers know their needs.

The following describes a fictive but realistic use case, which is used to represent the relevant context of base mapping in this research. A mapping agency of Ethiopia is in a desperate need of updating its framework database. The organization decided to use VGI dataset for this. But it is necessary to decide what and where data is needed and to let volunteers know this requirement. So the organization needs to prepare a request map that covers the extent where data is required and its associated metadata, which describes who requested the data, what is required and how the data should be produced. Then this request map will be submitted to a dedicated R/S map server and it becomes processed and stored in a spatial database of R/S maps. Then this request map stored in the database becomes published as OGC WMS and WFS service in a web map server. Then the metadata (service metadata plus the metadata submitted along with the request map by the ministry) of this WMS service will be harvested and stored in CSW catalogue server.

The status variable will be determined based on the requested status attribute by the requester such as completeness (spatial and attribute) and positional accuracy, from the contributed data in that particular area.

Local volunteers who are interested to map buildings of their locality area and other volunteers will evaluate the list of R/S maps provided in the R/S map portal interface and select the one in which they are interested to contribute based on the description of each R/S maps. The selected R/S map by the user will be shown on top of the base map for the user to see the geographic location, extent, and the status of the R/S map. The users can also search for request maps using place name of the locality and building as keywords from within the R/S map portal. Then the search result responds list of R/S maps which satisfies the search keyword criteria and the user can select the one which fits his/her interest. Once selected it becomes displayed and the description of that particular R/S map will also be displayed. The user can read its description to know what is required besides where. Since the R/S maps are interoperable, a user can also have access to the R/S maps from within GIS software he/she prefers to work on. In this case the user searches for R/S maps using the CSW-client integrated in Desktop GIS software (ArcGIS, QGIS) and will get list of R/S maps with the possibility of viewing their description (metadata) and can add the WMS map of the selected R/S map as a layer.

3.6.2. Geo-Wiki— theme mapping

The following describes a fictive but realistic use case, which is used to represent the relevant context of land use/land cover mapping in this research.

Ministry of Agriculture of Ethiopia needs to produce land cover map at countrywide coverage scale which shows the proportion of cropland and other cover type distribution throughout the country. The map is going to be used as one input for Agricultural policy making. So accurate land cover map of the area is required to achieve this goal. Nonetheless it is time consuming and costly to get ground truth datasets at countrywide scale that helps during image classification and validating classification result. Consequently, the ministry takes crowdsourced ground truth dataset as best alternative. Geo-Wiki is identified as a crowdsourcing tool to exploit volunteer's validation dataset contribution in this VGI system.

Volunteers should be informed about this requirement through the R/S map portal interface. The ministry is expected to prepare a request map together with its metadata. Then this request map will be submitted to a dedicated R/S map server and it becomes processed and stored in a spatial database of R/S maps. The request map stored in the database becomes published as OGC WMS and WFS service in a web map server. The metadata of this WMS service will be harvested and stored in CSW catalogue server. The status variable attribute value will be determined based on the requested status attribute by the requester such as completeness and attribute accuracy, from the contributed data in that particular area.

Local agricultural officers, researchers on the area or any other volunteer who is interested to contribute to this request will browse to the R/S map portal and evaluate the list of available R/S maps published in the portal based on their descriptions. When they get the one which fits their interest they can select and add it on top of the base map to see its geographic location, extent, and status of the request. From the visualized map they can decide where to contribute specifically for that particular R/S map. The users can also search for R/S maps using search keywords such as Ethiopia, agriculture from within the R/S map portal. List of R/S maps which satisfies the search criteria will be retrieved back from the catalogue server and presented to the user to select the one which fits his/her interest. The selected maps becomes displayed and the description of that particular R/S map will also be displayed. The user can read description of the R/S map to know what is the required accuracy level and classification scheme to be used for the selected R/S map besides where. Based on an information obtained from the request map and the status map— that shows the progress of the land cover assessment in a particular area (legend that shows different quantitative classes based on quantity of the contribution) the user will go for an area with relatively less data contribution (completeness) and less accurate contribution (thematic accuracy) and starts assessing the land cover in that particular area.

3.6.3. Ushahidi and OSM damage assessment map

The following describes a fictive but realistic use case, which is used to represent the relevant context of crisis mapping in this research.

Following the Typhoon Haiyan storm that struck Philippines, the Red Cross arrived on the affected area to give relief support for victims in the affected region. To accomplish their task they have to plan their support based on evidence. This in turn enables them to make critical decisions about where to send food, water, and supplies and where to conduct rescues. Detail and up-to-date Geospatial information of the affected region specifically on the situations of roads and buildings was required for Red Cross Workers and volunteers to effectively carry out their task. Geospatial data before the storm struck the area that shows where buildings and roads are and after the storm that shows where actually they are is needed. So a detailed damage assessment map was required. This can be done in two ways: 1) by comparing pre-disaster geospatial data of the area with high resolution post-disaster satellite images and 2) based on on-site damage assessment data report from field.

To get all these information timely, VGI data was identified as best alternative. For this, Red Cross needs to make a request for volunteer collaboration in producing the damage assessment map quickly in both OpenStreetMap and Ushahidi to benefit from the contribution of both systems. For this, request maps, that show the affected area extent and the type of data needed should be produced and communicated to volunteers. The affected area will have different levels of priority that can be indicated in the request map for volunteers to prioritize their contribution.

Once the request map is submitted to a dedicated R/S map server, the data becomes processed and stored in a spatial database of R/S maps together with its metadata. Then this request map stored in the database becomes published as OGC WMS and WFS service in a web map server. The metadata of the request maps WMS service will be harvested and stored in a Catalogue server.

Disaster management expert volunteers and other volunteers will browse to the R/S map portal and evaluate the list of available R/S maps published in the portal based on their description. When they get the one which fits their interest they can select and add it on top of the base map to see its geographic location, extent, and status of the request. From the visualized map they can decide where to contribute specifically for that particular R/S map. The users can also search for R/S maps using keywords (such as damage assessment, Red Cross, Typhoon Haiyan) through the R/S map portal. list of R/S maps which satisfies the search criteria will be retrieved back from the catalogue server and presented to the user to select the one which fits his/her interest. Once selected it becomes displayed and its description will also be displayed. The user can read the description of the R/S map to know what is the required data besides where. Then they will contribute for their selected R/S map in the process of producing the damage assessment map based on the priority of the area as specified in the request map. Since the R/S map is interoperable it can be accessed in both OpenStreetMap and Ushahidi systems, this bridges the gap in managing the information in both systems. It helps to communicate where damage assessment map is needed and what is needed in both systems. So volunteers (off-site users) who are working using the satellite imagery in OpenStreetMap and volunteers (on-site users) who are working on-site damage assessment and reporting to Ushahidi can have the ability to know where damage assessment is done and where not. This helps to get the best out of the two systems to produce the damage assessment map that gives quick and more accurate result. The R/S map can also be accessed from within a platform which can support OGC services such as Quantum GIS and ArcGIS through their CSW-client. This capability helps volunteers to access the R/S maps in their favourite GIS environment they preferred to work on. The status map will be determined and updated based on the characteristics of the collected VGI data in both systems. Based on the information in the status map each particular area will have its own status information indicating the level of assessment in that particular area (no assessment, medium, high). The requesters can also update their request map—both the description and the geometry, through the WFS-Transaction service and the catalogue server metadata editor.

3.6.4. OSM Flooding use case

The following describes a fictive but realistic use case, which is used to represent the relevant context of crisis mapping in this research.

Following flooding hazards in Pakistan, WFP was in need of up-to-date geospatial data of the inundated areas to deliver the relief operation effectively. The organization looked-for data specifically on roads, schools and hospitals within the flooded area, and location and names of cities and villages across the country. For this, the organization required help from volunteer communities contributing in VGI systems to get the data timely. The organization requested the map which shows the extent of the flooded area, created by the GIS department at the urban unit of Government of Punjab. A request was made to inform volunteers the features of interest WFP needs for its operation. The map obtained from the department was used as a request map together with additional information (metadata) such as what is needed to be mapped. It was then submitted to the server dedicated for the management of R/S maps. Then the request map was processed and stored in a spatial database and corresponding OGC compliant WMS and WFS service of the request map was published in a web mapping server. Then the metadata of this WMS service harvested and stored in CSW catalogue server.

Local volunteers and an international community mappers who can create map in VGI systems will browse to the R/S map portal and evaluate the list of available R/S maps published in the portal together with their descriptions. When they get the one which fits their interest they can select and add it on top of the base map to see the geographic location, extent, and the status of the R/S map. From the visualized map they can decide where to contribute specifically for that particular R/S map. Alternatively the users can also search for R/S maps using keywords such as Pakistan, flood, WFP through the R/S map portal. list of R/S maps which satisfies the search criteria will be retrieved back from the catalogue server and presented to the user to select the one which fits his/her interest. Once selected it becomes displayed and the description of that particular R/S map maps will also be displayed. The user can read the description of the R/S map to know what type of data is required and other information for the selected R/S map besides where. From this the user can go to the VGI system where the selected request map is published for.

Since the request map is interoperable, besides OpenStreetMap it can also be accessed from within GIS application programs such as Quantum GIS and ArcGIS through their CSW-Client and this speeds up the contribution process by increasing the accessibility of request maps. Based on the information from the request map and the status map, which shows the progress of the mapping activity in a particular area (no data, good, completed) the user will go for an area with relatively less data contribution and starts mapping in that particular area.

4. INTEROPERABLE R/S MAPS FRAMEWORK

The whole process of interoperable R/S map framework is divided in to three basic sections listed below, each of them are explained in a separate section along with explanatory diagram that shows different components and workflow.

- Request map publishing
- R/S map creation (generation) and Updating
- R/S map consumption.

4.1. Publishing request maps

This is the first stage in the whole process of R/S map production (see Figure 4). Various organizations who are in need of getting geo-information for a particular area take the first step by submitting (uploading) their request maps that represent an area where the requester needs VGI data to be produced by volunteers. The publishing process is going to be done through the R/S maps portal interface.

Depending on the type of request maps submitted by the user some processing is required before it comes out and exposed to volunteers as R/S map. The request map area coverage and description of the request needs to be analysed to know what type of R/S maps to setup. As mentioned above in the types of R/S map (see Section 3.3), the request might have priority attributes or not which affects the type of R/S map to create. Once the specification of what type of map is known it will pass to the next step of producing the R/S map. The request map and the status map can be represented together as one map or separately as individual maps and this affects the type of R/S map to produce (see section 3.3). Then Each R/S maps will be loaded in the spatial database of R/S maps as separate spatially-enabled table that can benefit from the features of database management system for better management of R/S maps.

A web mapping server plays a role of exposing R/S maps available in the database as OGC web services independent of technology or system differences the request for R/S maps comes from. It exposes (publishes) each R/S maps as a web Mapping service (WMS) and Web Feature Services (WFS).

Besides publishing R/S maps as a service their metadata will be documented and stored in a catalogue server, which helps to store the metadata of R/S maps which will be presented to the user when they search for R/S maps from this catalogue server through the R/S map portal interface. CSW catalogue server will be configured to expose R/S maps through standards based open interface, to access R/S maps from within different systems which implements this open standard web service. This can be implemented in a full-fledged catalogue server such as Geonetwork to effectively manage and give services in a production environment which can handle large number of metadata records of R/S maps.

4.2. R/S map creation (generation) and Updating

4.2.1. Status map creation and updating

After the request maps are published by users and the maps has been processed and stored in the database, the next step is to calculate the value of the status variables attribute for the status map.

In this part the first step is to have access for the VGI dataset from which the status variable attribute value is going to be determined. For those areas where there is request the VGI dataset can be downloaded from the respective VGI data server and stored as structured data (as opposed to unstructured data format nature of VGI) in the spatial database dedicated for storing VGI dataset. Once the data is stored in the database it becomes convenient to be accessed by other programs and SQL queries (both spatial and non-spatial) can also be executed to extract part of the data which fits certain criteria.

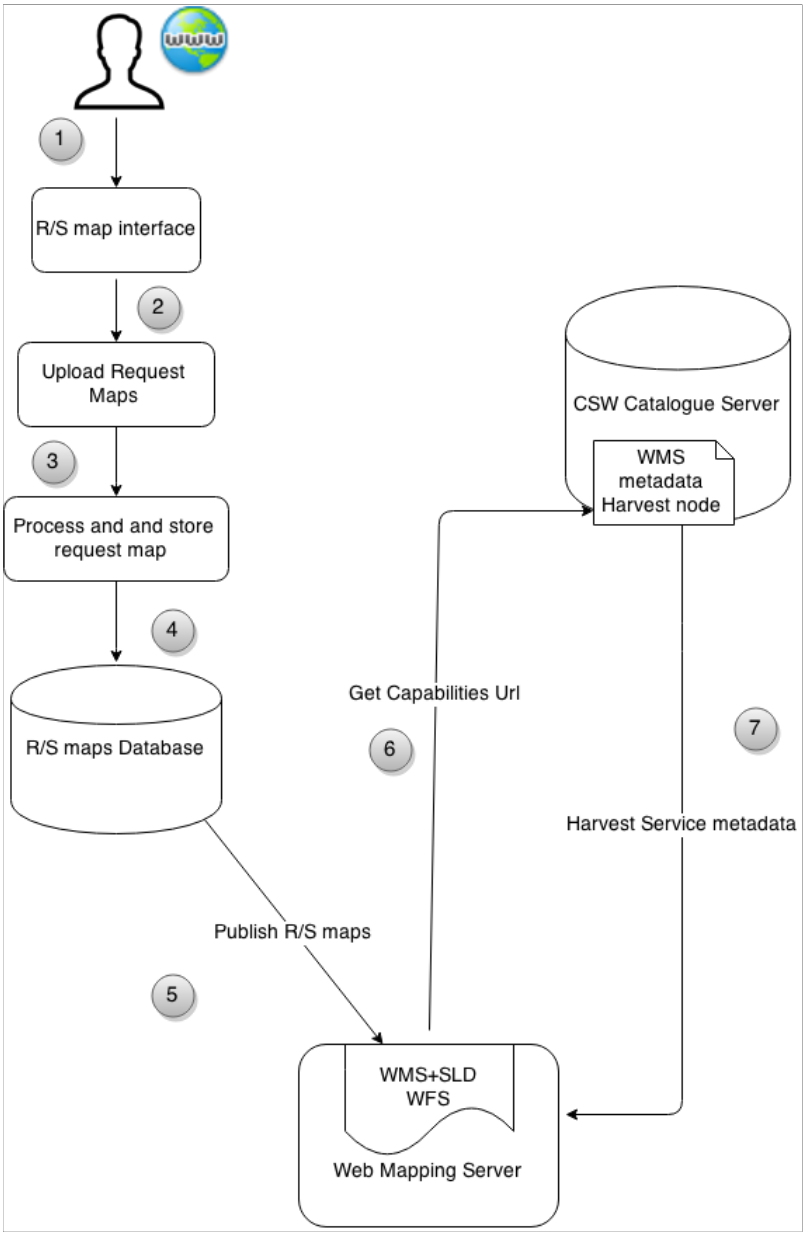

Figure 4: Request map publishing process

The python script (see **Error! Reference source not found.**) is dedicated to calculate the initial values of the status variables and also update the value changes of the status variables through time for the status map. The script accesses the data (VGI data and R/S maps) stored in the spatial database using database connection libraries to be able to get connected to the database. First, it establishes connection and then requests the R/S map for which the status variable value is to be calculated/updated and then its corresponding VGI data will also be requested based on some SQL queries formulated according to the data need. Based on the spatial relationship of the R/S map and the VGI dataset the script evaluates the VGI dataset characteristics for each grid cell (status boundary) which helps to determine the status variable attribute value in that area. Then the script writes the initial value of the status variable to its corresponding R/S map's attribute. When the script runs other than the first execution (to update the initial status variable value) for a particular status variable of the R/S map, the new value will be updated to the R/S map's status variable attribute.

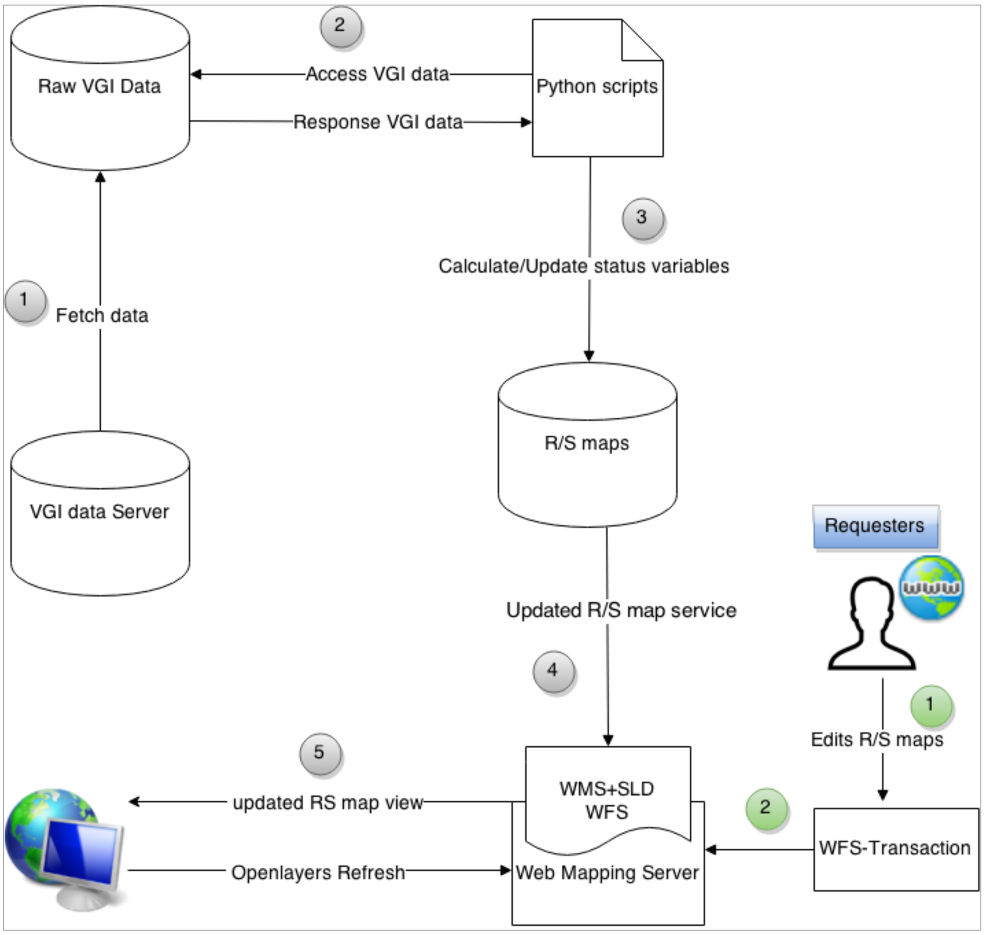

Figure 5: The status map generation and updating process

Future requests of R/S maps from the web mapping server reflects the updated value of the status variable from the database. For those users who are already viewing the maps on their interface the updating process of their current view will be taken care of by OpenLayers refreshing capability, which refreshes the browser by sending request to the map service without the knowledge of the user running behind the scene at a time interval adjusted for that particular R/S maps.

4.2.2. Updating R/S maps by requesters

Each R/S maps belongs to a particular owner who submitted the request and will be in charge of updating it in the future. R/S maps have its own associated information which describes purpose of the request including what is needed and how data should be produced. There is a possibility that once R/S maps are submitted some updates might be required by the requesters. These includes modifying the description, status variables required, type of features requested to be produced such as roads, buildings, modifying the extent of the request boundary. Through time they need to update their request map's attribute value such as the priority of mapping for different priority boundaries within the request map boundary. These same organizations are also responsible in tracking the status of their R/S maps and to remove those maps in which the requested data is completed.

The updating process can be done in two ways depending on what the requester is interested to do. When the owner wants to modify priority attribute of the request map, extent of the R/S map or delete completed R/S maps, first the owner needs to log on with their credentials which qualifies them to access only R/S maps that belongs to them and request for a particular R/S maps which they want to update. This gives them access to the raw R/S map published as WFS in a web mapping server. Then they can edit the WFS feature and its attributes (priority attribute) according to their need. The WFS transaction service allows them to perform the transaction type they want to do with the R/S map. When the transaction is committed the change will also be reflected on the database. If the update is required to modify the associated metadata information of the R/S maps, this will be done on the catalogue server. The owner needs to log on also here to access the metadata of R/S maps in which they have editing privilege. Then once the metadata is modified it will be saved and volunteers will get the new description.

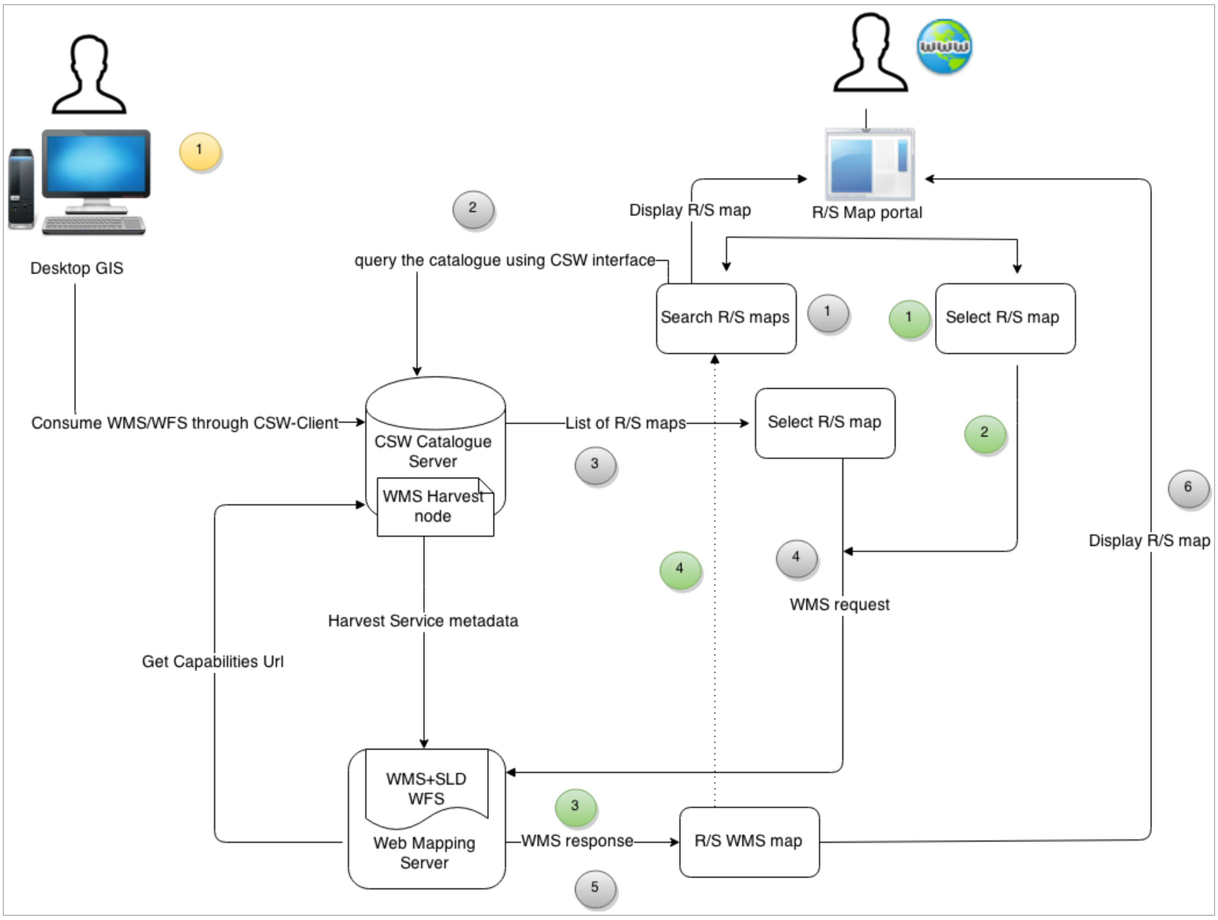

Figure 6: The process of consuming R/S maps

4.3. Consuming R/S maps

The last stage in the R/S map process is to consume the R/S maps published and catalogued in the catalogue server by volunteers using the R/S map portal interface. At this stage the user interacts with the R/S map portal interface (See Figure 6) as before in the publishing process, but here to consume R/S maps and obtain an information which helps to decide where to contribute to make contributions in a certain VGI system. This can be done in two alternative ways; either to select from the list of R/S maps based on their description or search for R/S maps using the search interface provided on the R/S map portal interface. When the user uses the later alternative he/she searches for R/S maps stored on the catalogue server using the search interface. The user's search keyword are used to retrieve the R/S map the user wants to get from the catalogue server. CSW interface based request of the user search criteria will be send to the catalogue server which also recognizes CSW requests. List of R/S maps obtained from the response will be send back and presented to the user. The selected R/S map's metadata will be presented to the user and WMS map will be requested from the web mapping server using the online resource address obtained from the metadata, and the map will be shown to the user on the map viewer. If the user uses the first alternative, then they can select from the list of R/S maps with their descriptions published on the portal for each R/S map layer. WMS Get Map request will be constructed for the selected layer and the response map becomes displayed on top of the base map. When working in desktop GIS programs such as ArcGIS and Quantum GIS, CSW-clients of the application will be used to get the required R/S map. The client program handles the communication process between the catalogue server and CSW-client to provide the requested R/S map by the user. The CSW client provides search interface and the search result lists the layers (R/S maps) and their metadata description that meets the search criteria. When the user selects to add one it will be displayed in the program's map display window.

5. PROTOTYPE IMPLEMENTATION AND TESTING

5.1. Technology and tools for prototype implementation

After evaluating the requirements to implement interoperable R/S maps the following technologies and tools were identified to facilitate the task at different stages in the R/S map workflow.

- ExtJS, a java script library to build user-friendly graphical interface for web applications (Sencha, 2015)

- OpenLayers, a java script library which provides mapping functionalities for web mapping applications, it serves as web mapping client for different OGC services such as WMS, WFS (OpenLayers, 2015).

- GeoExt, a JavaScript library which combines the user interface functionalities of ExtJS and mapping functionalities OpenLayers to build desktop style web mapping applications (GeoExt, 2015).

- PostgreSQL/PostGIS, spatial extension of PostgreSQL database management system software which enables to store geographic features as a table with its geometry stored as an attribute together with other attributes (PostgreSQL, 2014).

- GeoServer, a web mapping server which has user friendly interface to configure various OGC compliant geo web services. It provides easy to use interface through its web admin interface to configure geo web-services and manage the system. Once a layer is published it can be exposed as three different geo-services (WMS, WFS, WCS), which is quite important to save the configuration time needed in other web mapping servers such as MapServer. The layer preview mechanism helps to check whether the published layer is working and how it looks like before the final publishing(GeoServer, 2014).

- Geonetwork, is a full-fledged catalogue server which provides different kinds of functionalities, basic to advanced level of geospatial metadata management functionality. It also supports large number of metadata catalogue storage for production environment (GeoNetwork, 2014).

- WampServer, is windows based web development environment which provides all the tools required for web applications as preconfigured installations (WampServer, 2014).

- ArcGIS, is commercial GIS software for working with maps and geographic information (ESRI, 2015). It is used for creating vector grid map using the fishnet tool and to produce single R/S map using its multiple attribute visualization tool.

- Quantum GIS, is open source GIS software for working with maps and geographic information (QGIS DevelopmentTeam, 2015). It is used for SLD file generation and interoperability test.

5.2. Prototype Implementation

The prototype was implemented based on the proposed framework above and for a specific use cases such as Red Cross and WFP use cases based on OpenStreetMap and Ushahidi. The implementation of the framework for status map creation and updating (see Section 4.2.1) is out of the scope of this research as mentioned in section 3.2.1.

R/S map creation process

Semi-dummy R/S maps were produced for different countries sample administrative boundaries (the source for the administrative boundary vector dataset was obtained from DIVA-GIS free geographic data (DIVA-GIS, 2014)) using ArcGIS software. Different procedures were followed to produce different types of R/S maps identified above in section 3.3. For the case of separate request and status maps, a vector grid map which fits the extent of the request administrative boundary was created to represent the status map and the grid map was dissolved and a single polygon map was produced to represent the request map. An exception to this procedure is when the request map has attributes such as priority which creates different priority areas within the request boundary as opposed to the previous case without attributes which results in no partition inside the request map. In this case first the administrative request boundary was divided in to three priority areas (low, medium, high) to represent the request map. Since the status map is vector grid map and the administrative area boundary doesn’t fit nicely the status map’s grid boundary, so the vector grid map was assigned priority attribute value of the request map based on the location of each status grid in relation to the request map’s priority areas. Then the resulting grid map with priority attribute was dissolved based on the priority attributes assigned to each grid to produce a polygon partitioned in to different priority areas and fits the status map grid boundary. The status variable (completeness and damage assessment) attribute value was determined manually using OSM as a background to guide to assign near real (approximate) value to each status grid of the whole vector grid map. In the case of representing the request and status map as a single map (see Figure 7, Figure 8, Figure 9), multiple attribute visualization tool

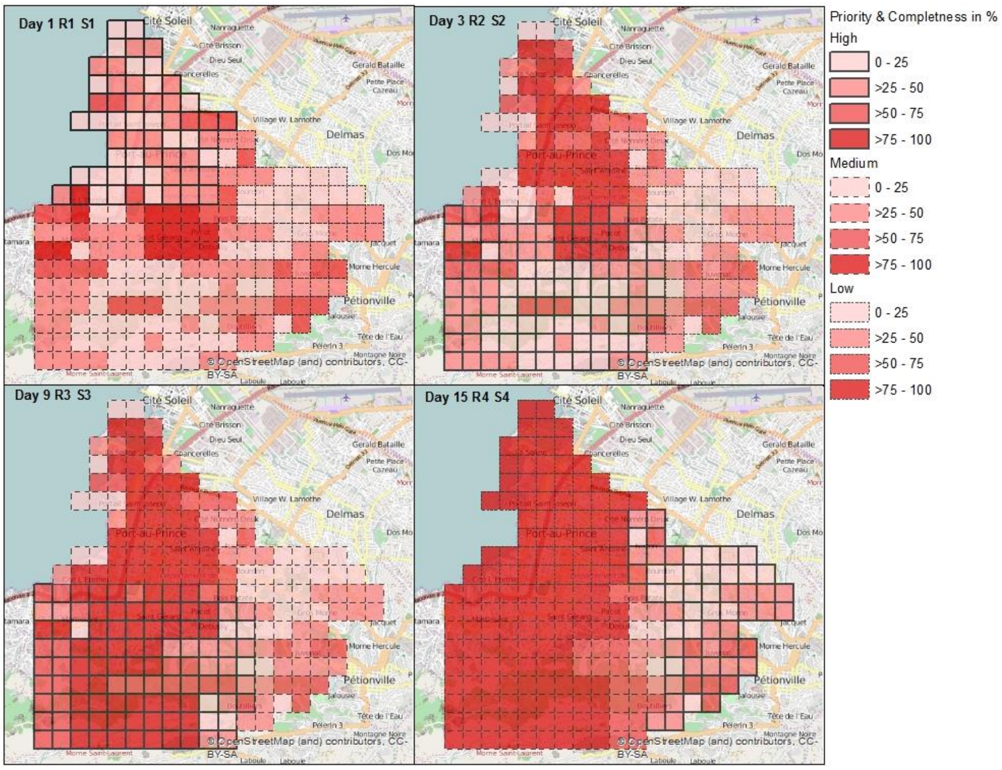

Figure 7: Combined representation of request and status map based on grain/texture visual variable

of ArcGIS was applied using two attributes of the vector grid map: priority as request attribute and completeness as status attribute.

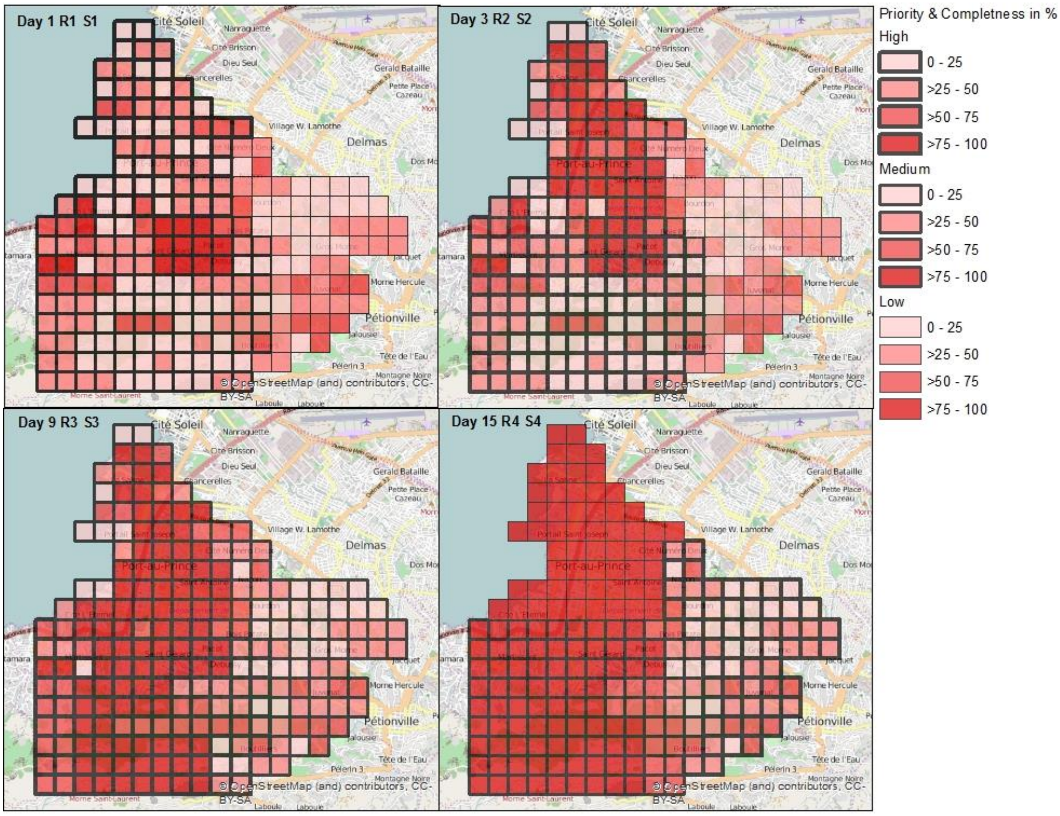

Figure 8: Combined representation of request and status map based on size visual variable

Publishing R/S maps

Both the request and status maps produced in ArcGIS software were uploaded in to the R/S map spatial database configured in PostgreSQL/PostGIS database. PostGIS's Shapefile and DBF loader plugin was used to import the R/S map shapefiles to the database.

GeoServer was setup in apache tomcat 6.0 as a mapping server to publish the R/S maps as OGC compliant web services. A store which reads data from the PostGIS database was configured in GeoServer using the connection parameters of the database such as the host, port and database name. Once the connection was successfully established the list of available R/S maps in the database are accessible to be published as a service. The layers were published as OGC web services. Once they are published GeoServer can expose a single layer as WMS and WFS services whenever a request comes. To render the R/S maps in a style other than the default style of GeoServer it needs to create SLD style file which helps to provide style information to visualize WMS layers. Different SLD rules were generated using QuantumGIS's functionality to convert layer's style to SLD file (see Figure 10), then the file was edited to correct colour issues and transparency of the layer. The SLD file was linked to its corresponding R/S map WMS layer.

GeoNetwork version 2.10.3 was deployed/installed on apache tomcat 6.0. The catalogue server was configured to serve metadata based on CSW interface so that the R/S maps can be requested using CSW interface from different clients, such as web client, and desktop clients of Quantum GIS and ArcGIS software through their CSW client. To store the metadata of the R/S maps represented as WMS in the GeoNetwork node, the harvesting capabilities of the catalogue server was used. It was configured to harvest the metadata of the WMS layers of the R/S maps using the url of the WMS service provided as part of the configuration. Then it harvests at a time interval adjusted during the configuration. Every time a new harvest is produced it replaces the previous metadata with the new one, this helps to keep up-to-date the metadata catalogue when new WMS layers are published in the web mapping server (GeoServer), and it also helps to avoid redundancy. The harvested metadata can also be searched with in the catalogue server.

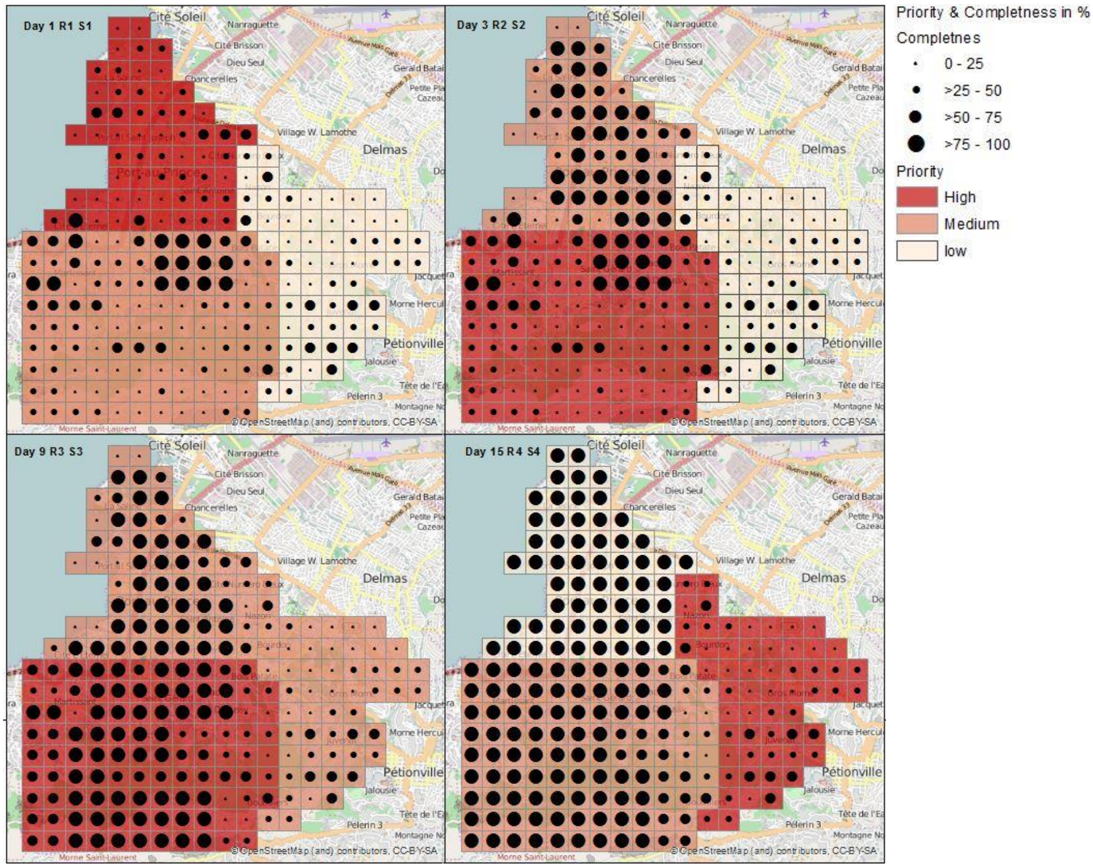

Figure 9: Combined representation of request and status map based on proportional symbol + Value

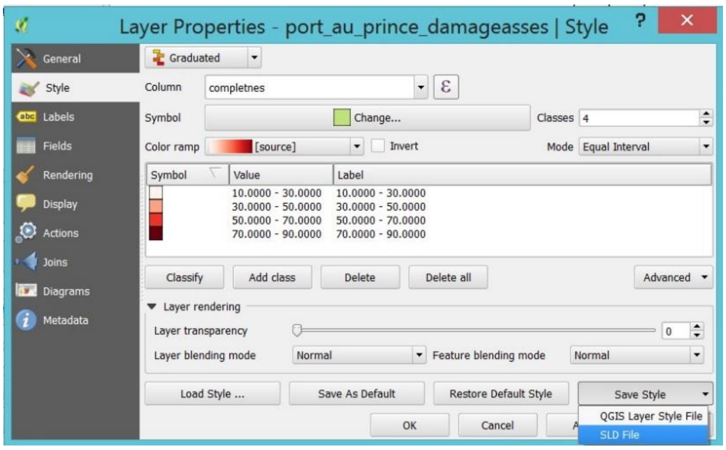

Figure 10: QGIS layer style to SLD conversion

Ushahidi

The Ushahidi platform was installed on WampServer and configured to implement the use case about using Ushahidi to produce damage assessment map. OpenStreetMap was configured as base map for R/S map layers using the map setting of the platform. In order to integrate the R/S maps to ushahidi, WMS support in Ushahidi is required since the R/S maps were published as WMS. The WMS plugin of Ushahidi was configured on the ushahidi platform. Then using this plugin's interface WMS layers of the R/S maps were configured as separate overlay layers to see them on top of the base map. But currently the plugin has no support for showing the legend of the WMS layers, but this can be upgraded in the future by extending the capability of the plugin to display the legend. Therefore the legend on the figure is not the real implementation representation, it is only to demonstrate how it should look like so that it can give information about what the different colours and symbols represent in the R/S maps for volunteers.

5.2.1. R/S map portal

R/S map portal was implemented to serve as a gateway for requesters to publish their requests and for volunteer users to access R/S maps published by varieties of organizations. The implementation was based on selected technologies for different components of the framework. This includes GeoServer as a web mapping server to publish R/S maps as OGC web services, GeoNetwork as a catalogue server to catalogue the metadata of the R/S maps and expose them through CSW interface, and JavaScript libraries such as ExtJS, OpenLayers, and GeoExt, these libraries are used to design the interface of the portal and incorporate mapping functionalities.

The portal has two basic categories represented by two corresponding tabs on the interface; the first category represented with 'view-R/S Maps' tab is dedicated to serve volunteers by providing functionalities specific to them, such as selecting from available list of R/S maps and explore. Under this category the user interface (see **Error! Reference source not found.**) has different components divided in to three basic functionality areas, the upper part of the component provides list of R/S maps in a grid. The grid is populated from the layers records of WMSCapabilitiesStore created through its WMSCapabliitiesReader by parsing the available

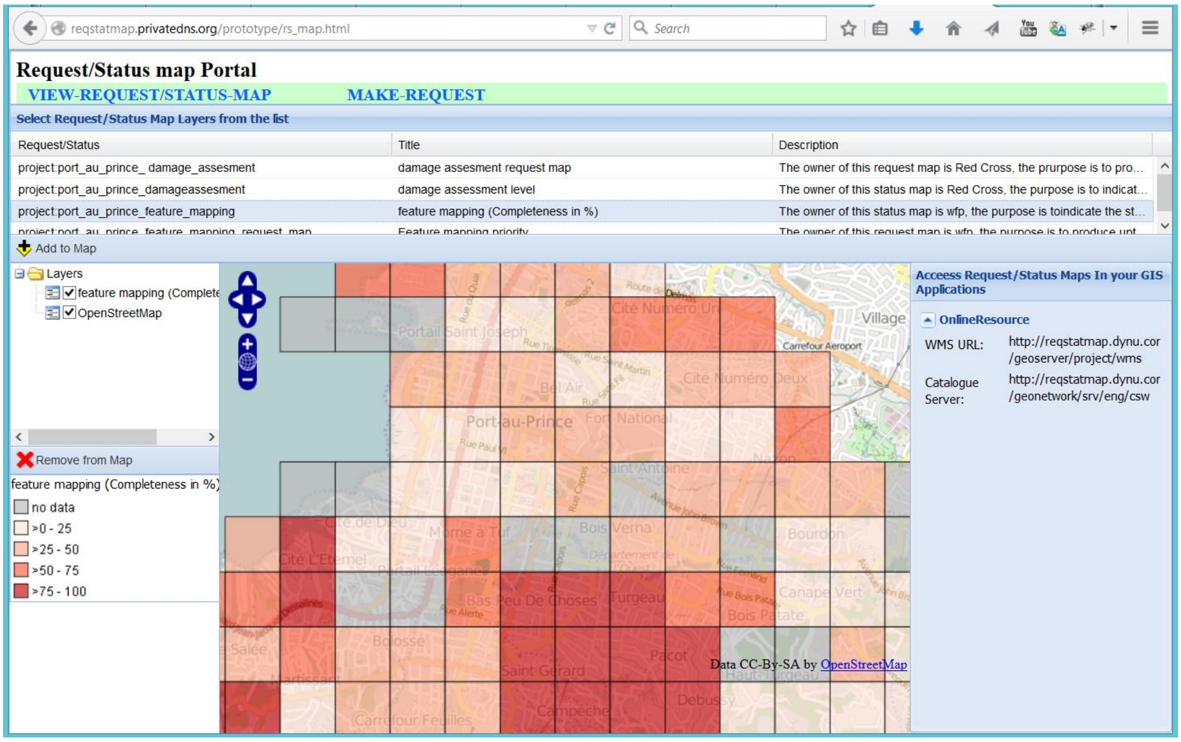

Figure 11: R/S map portal interface for volunteers

layers and its associated information such as layer name, title, abstract and others from the WMS GetCapabilities response. When the 'add to map' button is clicked for a particular layer selected on the grid, a WMS GetMap request will be established and the response map image will be displayed on the map panel of the portal, which is the central part of the interface that helps to display R/S maps on top of the OSM base map. On the left side there are two panels dedicated for layer management and legend information display. The layers added by the user will be listed on the layer management panel which helps to control the visibility and also to rearrange the order of the layers by dragging them up and down. The legend panel displays a legend information for the layers in the layer management panel. On the right side information about the WMS url and catalogue server url is given for users who are interested to access the R/S maps in their GIS software.

The second category of the interface represented with 'Make-Request' tab is dedicated for organizations to publish their request maps and manage their R/S maps through time. Functionalities provided for volunteers are also available in this category for organizations to explore requests made by other organizations before they make a request and also to see their own R/S maps. There are two additional components under this category. The panel on the right side provides functionalities for organizations to provide detail of their request and also upload request maps as shape file to publish their request. So they can make a request by completing the required information (both the request details and request map shape file) and finally submit the information. The last component is a grid panel located to the south of the interface which provides functionalities to update the R/S maps. It provides the corresponding WFS layer

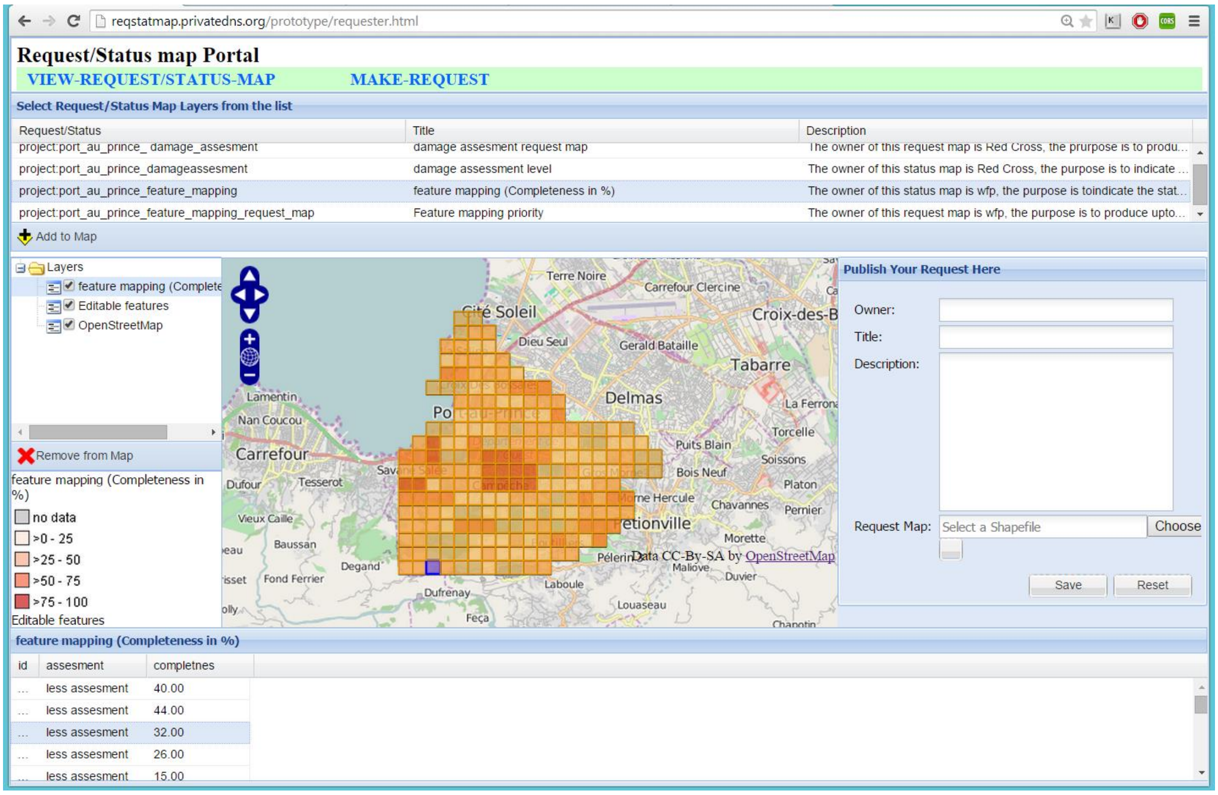

Figure 12: R/S map portal interface for requesters

of the R/S map layer selected on the layer management panel. The grid panel will be populated with individual features and attributes of the selected layer (see **Error! Reference source not found.**) so the user can edit both the attributes and geometries of features by using editing functionalities of WFS-T (the WFS-T part is not implemented).

5.3. Prototype testing

5.3.1. Interoperability Test

For an R/S map to be interoperable it needs to be accessible in applications where the R/S map is required to be accessed. So based on the use cases the applications where the R/S map is to be accessible was identified. OSM, Geo-Wiki and Ushahidi from VGI systems, and ArcGIS and QGIS from desktop GIS applications were identified. But the test was conducted for ushahidi and QGIS.

The following interoperability criteria were identified for the test:

- To check the accessibility of R/S maps in different VGI systems and VGI system editors
- To check the accessibility of R/S maps in different GIS applications

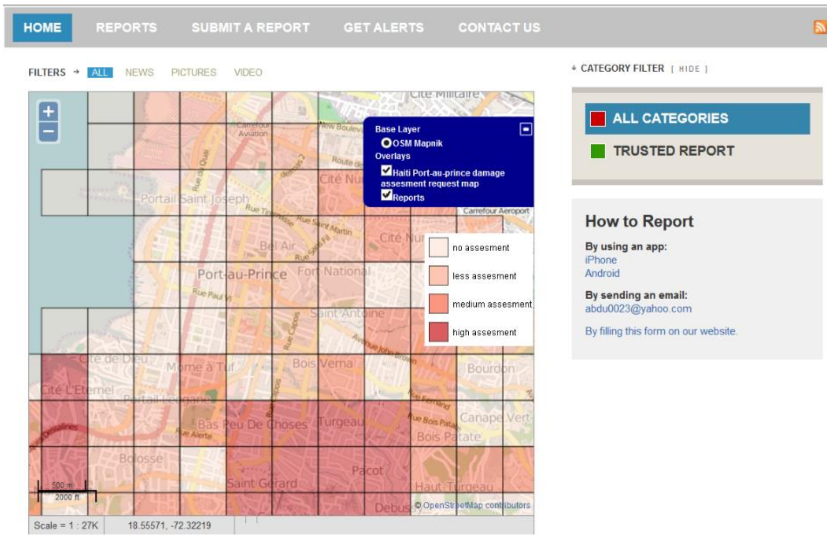

Figure 13: Port-au-Prince R/S map in Ushahidi

Based on the first criteria the test was conducted on Ushahidi and JOSM—OpenStreetMap editor. In ushahidi several R/S maps were configured for the test using Ushahidi's WMS plugin. All the maps of different places were successfully interoperable (see Figure 13**Error! Reference source not found.**). In OSM the same R/S maps were configured using the WMS/TMS functionality of JOSM editor (see Figure

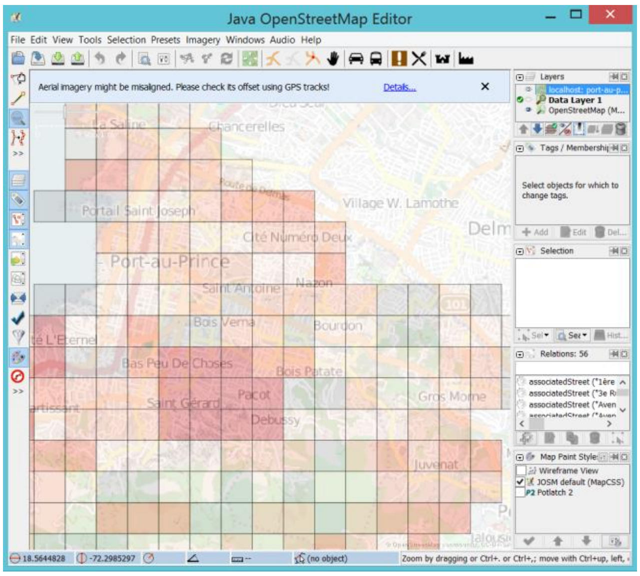

Figure 14: Port-au-Prince R/S map in JOSM editor

14) which provides WMS layers to be accessed for users as background layer. The maps were success fully interoperated with these VGI systems editor.

As explained in the use case above, a request to produce damage assessment map using Ushahidi and OSM by volunteers was made by Red Cross organization. The R/S maps for this request were able to be accessed both in ushahidi and OSM editor, JOSM. This capability is the importance of using standards based OGC services to represent the R/S maps that makes it interoperable in different VGI based systems.

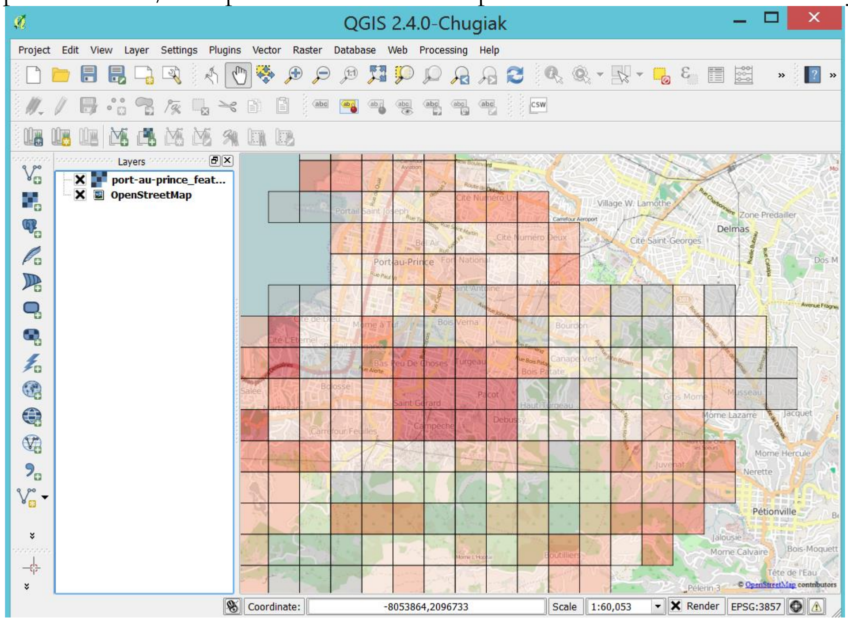

Figure 15: Port-au-Prince R/S map in QGIS

Based on the second criteria the test was carried out in QGIS 2.4 (see Figure 15) software. The software were used to access the R/S maps and display them with other layers loaded in the software as a background layer. The CSW client of QGIS (see Figure 16) was used to access the R/S maps metadata stored in Geonetwork catalogue server and then it presented the available WMS layers for the user to display them using the WMS online resource address from the metadata. The Geonetwork URL was configured for the CSW client to give the source catalogue server address. The test was successfully interoperable in both software.

5.3.2. Usability Test

Each application program, be it web or desktop based application has its own intended users (targeted audiences) and uses that aims to reach. The design of an application have to consider both use and user requirements to produce usable products. basically usability test is meant for testing ease of use of graphical user interface of a certain application (system) for the intended audience, but nowadays this concept has also been employed in cartographic research in which the effectiveness and efficiency of map users to draw the intended conclusion for what the map is produced is being tested (Kraak & Ormeling, 2010).Usability test helps to design a better application (system) that narrows down the gap between an application program capacity and users potential to benefit from the functionalities of the program (Manson, Kne, Dyke, Shannon, & Eria, 2012). In web mapping applications there are two elements to be tested: the visualization approach (method) used for the map and the graphical user interface which is going to be used by web map users. R/S maps shares these two elements, in one side it is a map by itself and on the other side it has a use environment, on web and on desktop GIS applications. So the usability test needs to consider both the map and the interface.

R/S maps are meant to be used In VGI environments, and there are varieties of users in the VGI world which can be categorized as producers and consumers. The producers can be further classified as amateur citizens which is the case most often and expert mapping professionals. So the composition of the participants for the test have to consider both types of volunteers and the consumers (requesters) too. The participants were randomly selected from ITC MSc students. All of them have geo-information background with different specializations: Land Administration, Geoinformatics, and Urban Planning and Management. Those participants from Geoinformatics and land administrations have various levels of web mapping and VGI background and the participants from UPM have basic understanding of VGI.

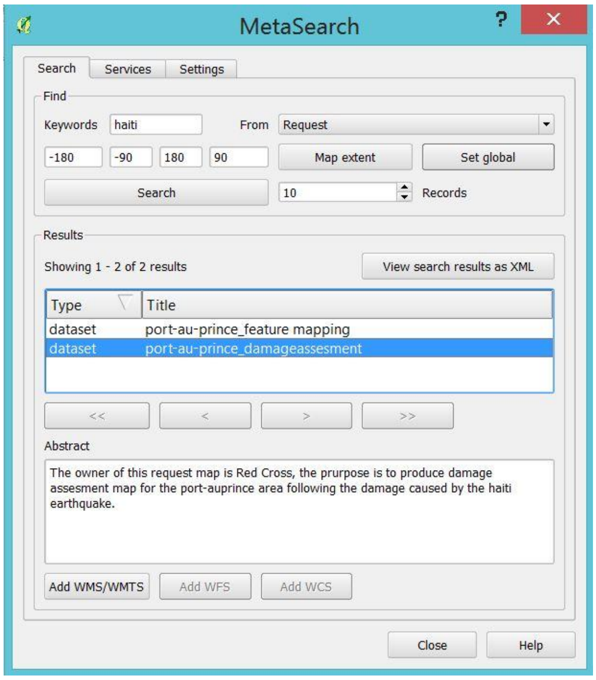

Figure 16: QGIS CSW client

Even if other usability methods and techniques such as eye tracking might give more beneficial information, according to the scope of this project interview method was applied just to do basic usability test. In this project 2 types of test persons were required for the test: volunteer (expert and novice users) and requester (organizations who wants to request for VGI datasets) type test persons. 7 test persons participated in the test. Each of them were interviewed in 2 rounds for 15-20 minutes based on selected questions which addresses the ease of understanding of the request and status maps and the ease of using the R/S map portal. On both rounds a general introduction was given to each participant about request and status maps concept and the portal. On the first round a scenario was presented to them to act as a volunteer to provide mapping contribution for request made by WFP organization to produce up-to-date base map of the affected region in the aftermath of Haiti earthquake. After the introduction and scenario description a follow up interview questions were asked before and after doing the requested task and their responses were recorded. On the second round the test persons were representing requester organizations role, specifically as Red Cross relief operation manager and a scenario to make request for damage assessment map production in OSM by volunteers following Haiti earthquake was presented to them and interview questions were asked as the first round but here specific to the role they are representing (manager of requester organization).

The result of the test can be summarized as follows, based on their interaction with the portal interface and the R/S map itself. Most of the test persons showed good understanding in interpreting request and status maps except for some colour problems which confused them with the overlay of the request and status maps. There were also some difficulties they came across in using the portal. They found it difficult to get a corresponding status map for a particular request map since the status map and request maps were represented as separate maps. For some of the users the interface was not simple to get the request and status maps and they also preferred to use searching functionalities than selecting from lists.

Based on the response of test persons some modifications were applied such as giving distinct colours for request and status maps which enables to view both maps at the same time. And also single R/S maps which transfers both the request and status map message were designed using ArcGIS's multiple attribute visualization technique. In order to apply the same visualization for the R/S maps we need to define several presentational rules using SLD to be able incorporate styles required for each attribute by defining the required rule.

6. DISCUSSION, CONCLUSIONS AND RECOMMENDATIONS

6.1. Discussion on research questions

1) What are request maps and status maps and how do they relate to each other?

Understanding R/S maps were one of the research question the project aimed to answer. So this project started by studying what R/S maps are and what their characteristics are based on the selected VGI systems and use case characteristics. After analysing the nature of R/S maps finally they were defined and understood in the following way.

As explained above (see Section 3.1) request maps are part of the R/S map which gives the information concerning the extent and the location of the request. It also provides information about the characteristics of the request which helps to inform volunteers about the requirements of the mapping activity for that particular request so that they will be able to produce the required data according to the request specification. Status maps are also part of R/S maps which specifically deals with providing status information based on different pieces of status boundaries (grids) that lies within the extent of the request map. Both maps can be represented together as a single map or separately as individual maps. This depends on the nature of the VGI system—where the R/S map needs to be published—and use case characteristics. For instance when the objective of the request is to inform volunteers to map crisis incidents by reporting to Ushahidi, the request map can be represented with simple polygon which shows the extent of the request and also provides information about the request. The VGI system's clustered point representation can represent the status map that shows highly dense areas of incidents with bigger point size than less dense incident areas (see section 3.3). Request and status maps complement one another. Change on the status map affects also the state of the request map. For instance when the completeness status value of grid cells within high priority area of the request map becomes high (completely mapped) the priority value of the request area changes from high to low.

2) What types of request/status maps are relevant for different type of VGI-based activities?

As explained above in section 3.3, different types of R/S maps are identified, each with unique and common characteristics. VGI system characteristics, use case nature and type of data required by requesters have a role in determining the type of R/S map that needs to be published for particular request. Based on VGI systems characteristics and use cases different types of R/S maps were identified by analysing requirements of R/S maps for OpenStreetMap, Ushahidi and Geo-Wiki. Table 1 illustrates the result of comparison of R/S map characteristics for the abovementioned VGI systems under topographic base mapping and crisis mapping use cases. Different comparison criteria were identified such as attributes required for both request and status maps, description of request, representation (visualization) of R/S maps, limit of duplication level of VGI datasets prior to status variable attribute value calculation for a status map, management of R/S maps, data type and updating frequency required. Characteristics of R/S maps such as Attribute, duplication level and updating frequency were found to show variation between different VGI systems and use cases. The variation was also noticed with in same VGI system but with different use case. For instance when OSM was considered in both use cases, characteristics of R/S maps required for both use cases showed variation in their attributes, duplication level, and updating frequency. The type of data requested by requesters can affect also the type of R/S maps. For instance a request to label local place names was required in the case of Pakistan flooding for humanitarian workers to get easily from one place to another without miscommunication during their relief operation process. In this case representation of the request map will be a polygon which covers the extent of the request. Whereas for a request to map features within the affected area the R/S map will be represented with vector grid map in which each grid shows status of the mapping process. In the previous case the annotated (marked) place names will be enough to differentiate between labeled and not labeled places. Besides the representation there is also a difference in other characteristics of the R/S map such as attribute of the R/S map. In the labeling process the tag and the language are most important attributes that should be described for volunteers because the name of the place names will be affected if it is not based on common standards explained in the request.

3) What are appropriate data models for request/status maps?

The data type of R/S maps didn't show a variation among the selected VGI systems. Vector data model was used to represent R/S maps. There are various reasons for this, the first and most important thing is that, R/S maps need to be accessed by requesters as raw vector data so that they will be able to modify it, and this is facilitated by WFS-Transaction service which functions based on vector datasets published as WFS. The other reason is that, most of VGI datasets are stored (represented) as vector datasets so using the same representation as the source data for the R/S maps helps for various analyses required to calculate the status variable (e.g. completeness) attribute value of the status map. Besides it is also necessary (precondition) to access each status polygons of the R/S map so that to be able to calculate and update its status variable attribute value and this can be easily managed when the data model is in vector.

4) How to handle request/status maps' multiplicity in source and time?

One of the research question was dealing with multiplicity nature of R/S maps in source and time. Since various organizations can make request for VGI datasets, this results in multiple R/S maps. Through time organizations can make varieties of requests for different purposes on different places or even a single organization can make multiple requests on same area each with different data requirement. This increases the number of R/S maps and all this maps needs to be catalogued and stored in an organized way. In this project a catalogue server which implements CSW standard was used to manage R/S maps. Sample R/S maps with its associated information were documented in GeoNetwork catalogue server and exposed based on OGC CSW interface. The metadata was successfully searchable using QGIS's CSW-client.

5) What are proper modes of operation for request/status maps (easy deployability, Ownership, update frequency)

The design takes into consideration the complexity level required for R/S maps application implementation. So that it can be easily setup with less effort, yet gives effective services. For this, the method was designed based on state of the art open source technologies which gives efficient services with little complexity required. Geonetwork catalogue server was used in this project, it is a full-fledged catalogue server to manage geospatial information and makes eases the process of integrating geoinformation metadata from various sources such as geo-web services, to a Geonetwork node. GeoServer was used as a web mapping server since it provides a user friendly interface to publish R/S maps and once a feature is published it can be exposed as WMS and WFS without additional effort to publish each of them separately. The design can also be implemented with other technologies, it is not dependent on the selected technologies for prototype implementation. It also considers managing ownership of R/S maps through registration of R/S map requesters and each with their own credentials to access specific R/S maps which belongs to them only.

6) Who are the users of request/status map and how do they use them?

Two types of users were identified in this project. They ca be classified as requesters and volunteers (see Section 3.4). Each of them have different requirements for the R/S map. Volunteers need to access and explore R/S maps published by various organizations, whereas Requesters needs to see requests made by other organizations and makes their own request. The design considered the requirement of both types of users.

7) What are appropriate visualisations for request/status maps?

A visualization method, which communicates both request and status information in an effective way was identified. For this, characteristics of users of R/S maps, use case, the VGI system itself and the way R/S maps are going to be represented (separately or single R/S map) were considered. These are the factors, the research found to be affecting cartographic visualization technique/method. In OSM the feature types are point, line and polygon, each symbolized with different symbols and colors, where as in Ushahidi besides the base maps such as OpenStreetMap or Google maps the crisis incidents are also represented with aggregated points each having its own size, color and label. So we need to consider all these situations to select appropriate cartographic visualizations for the R/S maps when it is going to be visualized with the base map so that users can understand the map easily.

The use case can affect also the visualization method of R/S maps.Two different types of representations were identified for Ushahidi and OSM. Vector grid map was used for OSM. In Ushahidi two ways of representations were identified: vector grid map for damage assessment use case and a simple polygon map for crisis reporting use case. The way R/S maps represented (as separate or as single combined map) was also one of the factor that affected the cartographic visualization method selection. First, Separate request and status maps were produced; a request map with priority attribute was represented as a polygon partitioned in to different priority areas represented with colour which shows order of priorities. Status map was represented as a vector grid map which shows the status in terms of completeness by using a colour which ranges from light to dark. But during usability test this way of representation showed some difficulties for users to search corresponding status map for a selected request map. So we considered representing them as single combined map. Then a single map which communicates both status and request map information was designed using ArcGIS's multiple attribute visualization technique. Selecting a visual variable which effectively and efficiently communicates the request and status maps were an important factor. Several versions of single map representations were produced with different visual variables. A map which represents priority of the request map with value visual variable and status map's completeness attribute with proportional point symbol (see Figure 9) was easily understood by users asked informally after usability test. The selected visualization techniques were applied for the WMS representation of R/S maps as SLD file created for different visualizations and attached to the corresponding layer.

8) What are the criteria that make request/status maps interoperable?

Interoperability criteria were defined based on the platforms where R/S maps need to be accessible. In this project two criteria were identified. The first criterion is to access R/S maps in different VGI systems where the R/S map is published for and the second is to access R/S maps in various desktop GIS applications. Both criteria were applied during the interoperability test in which R/S maps were deployed in the abovementioned VGI systems successfully (see Section 5.3.1)

9) Can the current OGC Web services qualify (support) for request/status maps handling? If not what needs to be extended?

WMS, WFS, CSW and SLD OGC web services were identified to be required web services to implement R/S maps. WMS was used to visualize R/S maps in different platforms and it has good support in most of GIS applications such as ArcGIS and Quantum GIS, and in VGI systems except in OSM editor such as iD editor and potlatch, there is no direct support for WMS in both editors, it needs to convert the WMS URL in to custom URL to display WMS maps in both editors. The remaining OGC services were not directly involved with the VGI systems but they have their own role in the R/S map process. There was no any specific problems of OGC web services in relation with support for R/S maps.

6.2. Conclusions

In this project a method for interoperable R/S maps handling in VGI systems was designed and implemented to introduce a uniform way of handling R/S maps in different VGI systems. The method was designed based on requirements identified for three VGI systems characteristics namely OpenStreetMap, Ushahidi and Geo-Wiki selected based on two use cases, topographic base mapping and crisis mapping. The type of R/S maps required in different VGI based systems needs to be analysed in order to come up with a good R/S map type that fits the purpose of a certain request. VGI systems and use case characteristics were the two key factors which influence the type of R/S maps needed. They also affect the visualization method to communicate R/S maps with the target audience. Besides, the understanding of the characteristics of request and status maps helps very much in designing better visualization methods so that both maps are interpreted in the intended way, which in turn maximizes quality of contributions provided by volunteers. The possibility of implementing the method using open source technologies makes the R/S map easily implementable and contributes towards interoperability of R/S maps. The interoperability test proved that it is promising to implement interoperable R/S maps through standards-based OGC services but VGI systems should accept the R/S maps concept and facilitate the integration of R/S maps with their systems by providing more improved features than they currently support. Currently, the WMS support in VGI systems is not complete, it doesn't include legend support which needs to be there for the interpretation of R/S maps by users. So this feature needs to be incorporated in the VGI systems. So far the WMS support in VGI systems only supports integrating a map image as a background layer for volunteers to help their editing process with their own WMS layers as background, but it doesn't include legend of the map. This is actually since the initial objective of providing WMS support in VGI systems editors such as JOSM is for volunteers to use their own data as background layer instead of the available satellite imageries. When VGI systems accept and start to implement R/S maps this support can be upgraded to complete support level. The usability test indicated that still there is a need to provide more user-friendly user interfaces for R/S maps and this needs to be researched.

6.3. Recommendations

In this project three types of VGI systems were considered under two different use cases, but this doesn't mean these are the only VGI systems and cases. There are varieties of VGI systems and use case scenarios in the real-world. So R/S maps characteristics required specifically to this VGI systems and use cases needs to be researched.

R/S maps needs to be as easily accessible as possible to every potential volunteers who can contribute VGI datasets using their cell phones. So extending the accessibility of R/S maps to mobile phones helps to communicate R/S maps to as many volunteers as possible, which in turn increases the contribution and speeds up the response to a request made by several organizations.

In this project a basic usability test was carried out using interviews. The test persons were also expert volunteers who took both the volunteers and the requesters role during the test. Full-fledged usability test need to be carried out using other methods such as eye tracking which provides more beneficial information and with real test persons. This helps to get more beneficial feedback which contributes to improve both the request and status map visualization approach and the portal user interface.

REFERENCES

Ajmar, A., Boccardo, P., Tonolo, F. G., & Veloso, C. (2010). Earthquake damage assessment using remote sensing imagery. The Haiti case study. In O. Altan, R. Backhaus, P. Boccardo, & S. Zlatanova (Eds.), *Geoinformation for Disaster and Risk Management Examples and Best Practices* (p. 158).

Albrecht, J. (1999). Towards interoperable geo-information standards: A comparison of reference models for geo-spatial information. *The Annals of Regional Science*, *33*(2), 151–169. doi:10.1007/s001680050098

Amelunxen, C. (2010). *On the suitability of Volunteered Geographic Information for the purpose of geocoding* (pp. 1–8). Retrieved from http://www.agit.at/php_files/myagit/papers/2010/8145.pdf

Anderson, J. R., Hardy, E. E., Roach, J. T., Witmer, R. E., & Peck, D. L. (1976). A Land Use And Land Cover Classification System For Use With Remote Sensor Data. *A Revision of the Land Use Classification System as Presented in U.S. Geological Survey Circular 671*, *964*. Retrieved from http://www.mendeley.com/catalog/land-land-cover-classification-system-remote-sensor-data-79/

Ariffin, I., Solemon, B., Anwar, R. M., Din, M. M., & Azmi, N. N. (2014). Exploring the potentials of volunteered geographic Information as a source for spatial data acquisition. *IOP Conference Series: Earth and Environmental Science*, *20*(1), 12041. Retrieved from http://stacks.iop.org/1755-1315/20/i=1/a=012041

Auer, M., & Zipf, A. (2010). How do Free and Open Geodata and Open Standards fit together? From Sceptisim versus high Potential to real Applications. In *The First Open Source GIS UK Conference*.

Barron, C., Neis, P., & Zipf, A. (2014). A Comprehensive Framework for Intrinsic OpenStreetMap Quality Analysis. *Transactions in GIS*, n/a–n/a. doi:10.1111/tgis.12073

Coleman, D. J. (2013). Potential Contributions and Challenges of VGI for Conventional Topographic Base-Mapping Programs. In *Crowdsourcing Geographic Knowledge: Volunteered Geographic Information (VGI) in Theory and Practice* (pp. 245–263).

DIVA-GIS. (2014). DIVA-GIS. Retrieved December 25, 2014, from http://www.diva-gis.org/Data

Elwood, S., Goodchild, M. F., & Sui, D. Z. (2012). Researching Volunteered Geographic Information: Spatial Data, Geographic Research, and New Social Practice. *Annals of the Association of American Geographers*, *102*(3), 571–590. doi:10.1080/00045608.2011.595657

ESRI. (2015). ArcGIS. Retrieved November 07, 2014, from http://www.esri.com/software/arcgis

Fast, V., & Rinner, C. (2014). A Systems Perspective on Volunteered Geographic Information. *ISPRS International Journal of Geo-Information*, *3*, 1278–1292. doi:10.3390/ijgi3041278

Fritz, S., McCallum, I., Schill, C., Perger, C., See, L., Schepaschenko, D., … Obersteiner, M. (2012). Geo-Wiki: An online platform for improving global land cover. *Environmental Modelling & Software*, *31*, 110–123. doi:10.1016/j.envsoft.2011.11.015

GeoExt. (2015). GeoExt. Retrieved January 07, 2015, from http://geoext.org/

GeoNetwork. (2014). GeoNetwork. Retrieved November 25, 2014, from http://geoserver.org/

GeoServer. (2014). GeoServer. Retrieved November 08, 2014, from http://geoserver.org/

Girres, J.-F., & Touya, G. (2010). Quality Assessment of the French OpenStreetMap Dataset. *Transactions in GIS*, *14*(4), 435–459. doi:10.1111/j.1467-9671.2010.01203.x

Goodchild, M. F. (2007). Citizens as sensors: the world of volunteered geography. *GeoJournal*, *69*(4), 211–221. doi:10.1007/s10708-007-9111-y

Google. (2014). What is Google Map Maker? Retrieved from https://support.google.com/mapmaker/answer/157176?hl=en

Harrison, J. (2010). Haiti SDI - WMS reveal a city of tents and need. Retrieved from http://carboncloud.blogspot.nl/2010_01_01_archive.html

Heipke, C. (2010). Crowdsourcing geospatial data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *65*(6), 550–557. doi:10.1016/j.isprsjprs.2010.06.005

Herold, M., Hubald, R., & Di Gregorio, A. (2009). *Translating and evaluating land cover legends using the UN Land COver Classification System(LCCS)* (p. 189).

HOT Tasking Manager. (2014). Retrieved from http://learnosm.org/en/coordination/tasking-manager/

HOT-OSM. (2014). Tasking Manager. Retrieved from http://learnosm.org/en/coordination/tasking-manager/

Kraak, M.-J., & Ormeling, F. (2010). Map use and usability. In *Cartography: Visualization of Spatial Data* (Third Edit., p. 187).

Liu, S. B. (2014). Crisis Crowdsourcing Framework: Designing Strategic Configurations of Crowdsourcing for the Emergency Management Domain. *Computer Supported Cooperative Work (CSCW)*, 389–443. doi:10.1007/s10606-014-9204-3

Manson, S. M., Kne, L., Dyke, K. R., Shannon, J., & Eria, S. (2012). Using Eye-tracking and Mouse Metrics to Test Usability of Web Mapping Navigation. *Cartography and Geographic Information Science*, *39*(1), 48–60. doi:10.1559/1523040639148

Meier, P. (2014). Results of the Crowdsourced Search for Malaysia Flight 370 (Updated). Retrieved from http://irevolution.net/tag/tomnod/

Meyer, R. (2013). How Online Mapmakers Are Helping the Red Cross Save Lives in the Philippines. Retrieved from http://www.motherjones.com/environment/2013/11/how-online-mapmakers-are-helping-red-cross-save-lives-philippines

Mitchell, T. (2007). *Web Mapping Illustrated: Using Open Source GIS Toolkits (Google eBook)* (p. 400). O'Reilly Media, Inc. Retrieved from http://books.google.com/books?id=IdGoy2rZSyIC&pgis=1

OGC. (2004). *Geospatial Portal Reference Architecture A Community Guide to Implementing Standards-Based Geospatial Portals. Portal* (p. 23).

OGC. (2006). *Web Map Server Implementation Specification.* doi:http://www.opengeospatial.org/standards/wms

OGC. (2007a). *Catalogue Services Specification 2 . 0 . 2.*

OGC. (2007b). *Styled Layer Descriptor profile of the Web Map Service Implementation Specification* (Vol. 0, p. 53). Retrieved from http://scholar.google.com/scholar?q=related:KgaxkB8pozMJ:scholar.google.com/&hl=en&num=30&as_sdt=0,5\npapers2://publication/uuid/1B5E861F-CE8F-4602-84D1-4BAAA0C763A3

OGC. (2010). *OpenGIS Web Feature Service 2.0 Interface Standard* (p. 253). Retrieved from http://www.opengeospatial.org/standards/wfs

OGC. (2011). OGC Reference Model. *Wikipedia.org*. Retrieved from http://en.wikipedia.org/wiki/OGC_Reference_Model\npapers2://publication/uuid/61C18A45-50DB-4993-8BE1-DD13ECBAD21D

OGC. (2015). About OGC. Retrieved February 14, 2015, from http://www.opengeospatial.org/ogc

OpenLayers. (2015). OpenLayers. Retrieved November 05, 2014, from http://openlayers.org/

OpenStreetMap. (2014). Retrieved from https://www.openstreetmap.org/about

OpenStreetMap Forum. (2013). How to use wms link (from rectified/warped map) as background in Potlatch or JOSM. Retrieved from https://help.openstreetmap.org/questions/13868/how-to-use-wms-link-from-rectifiedwarped-map-as-background-in-potlatch-or-josm

OpenStreetMap Forum. (2015). How to use a WMS layer as background in iD editor? Retrieved from https://help.openstreetmap.org/questions/40158/how-to-use-a-wms-layer-as-background-in-id-editor

OpenStreetMap Wiki. (2014). Using WMS in JOSM. Retrieved from http://wiki.openstreetmap.org/wiki/WMS

OSM Community members. (2014). Retrieved from http://neis-one.org/

OSM Tasking Manager. (2014). Retrieved from http://wiki.openstreetmap.org/wiki/OSM_Tasking_Manager

PostgreSQL. (2014). PostGIS. Retrieved November 05, 2014, from http://www.postgresql.org/

QGIS DevelopmentTeam. (2015). QGIS Geographic Information System. Retrieved November 10, 2014, from http://qgis.osgeo.org

QGIS OSM Plugin. (2014). Retrieved from http://wiki.openstreetmap.org/wiki/QGIS

Rautenbach, V., Coetzee, S., & Iwaniak, A. (2013). Orchestrating {OGC} web services to produce thematic maps in a spatial information infrastructure. *Computers, Environment and Urban Systems*, *37*(0), 107–120. doi:http://dx.doi.org/10.1016/j.compenvurbsys.2012.08.001

Reichardt, M. (2014). Standards: Key to VGI success. Retrieved from http://www.geospatialworld.net/Paper/Technology/ArticleView.aspx?aid=25328

Roche, S., Propeck-Zimmermann, E., & Mericskay, B. (2011). GeoWeb and crisis management: issues and perspectives of volunteered geographic information. *GeoJournal*, *78*(1), 21–40. doi:10.1007/s10708-011-9423-9

Sencha. (2015). ExtJS. Retrieved January 07, 2015, from http://www.sencha.com/products/extjs/

Team Google Map Makerpedia. (2014). Pakistan Flooding. Retrieved from https://sites.google.com/site/mapmakerpedia/community-projects/pakistan-flooding

Tomnod. (2014). Missing Airplane: Malaysia Airlines MH370. Retrieved from http://www.tomnod.com/campaign/mh370_indian_ocean/map/1fgx12y6e

WampServer. (2014). WampServer. Retrieved October 25, 2014, from http://www.wampserver.com/en/

Wikipedia contributors. (2015). Topographic map. Retrieved from http://en.wikipedia.org/wiki/Topographic_map

APPENDIX

A.1 R/S map portal

Listing 1.1: View-R/S map

```
<html>
  <head>
    <title>RS Map Portal</title>
    <link rel="stylesheet" type="text/css" href="ext/resources/css/ext-all.css">
    <script type="text/javascript" src="ext/adapter/ext/ext-base.js"></script>
    <script type="text/javascript" src="ext/ext-all.js"></script>
    <script src="openlayers/lib/OpenLayers.js"></script>
    <script type="text/javascript" src="geoext/lib/GeoExt.js"></script>
                <script type="text/javascript" src="FileUploadField.js"></script>
                <style>
                #banner ul  {
      overflow: hidden;
                          list-style: none;
                          margin-left: auto;
      margin-right: auto;
      width: auto;
      background-color: #CCFFCC;
    }

    #banner  li {
      float: left;
    }

    #banner a:link,  #banner a:visited {
      display: block;
      width: 250px;
      font-weight: bold;
      color: #0066FF;
      background-color:         #CCFFCC;
      text-align: center;
      padding: 2px;
      text-decoration: none;
      text-transform: uppercase;
    }

     #banner a:hover,  #banner a:active {
      background-color: #00CCFF;
    }
```

```
    </style>
    <script type="text/javascript">
            Ext.BLANK_IMAGE_URL = "ext/resources/images/default/s.gif";
    var controls = [];

    Ext.onReady(function() {
             var fromProjection = new OpenLayers.Projection("EPSG:4326"); // WGS
1984
        var toProjection = new OpenLayers.Projection("EPSG:900913"); // Spherical

      var map = new OpenLayers.Map(null, {
        projection: new OpenLayers.Projection("EPSG:3857"),
        allOverlays: true,
        maxExtent: new OpenLayers.Bounds(-20037508.34, -20037508.34,
                20037508.34, 20037508.34),
        units: 'm'
      });
             var osmLayer = new OpenLayers.Layer.OSM();
             map.addLayer(osmLayer);
             map.addControl(new OpenLayers.Control.Navigation(),
        new OpenLayers.Control.Attribution(),
        new OpenLayers.Control.PanPanel(),
        new OpenLayers.Control.ZoomPanel());

             mapPanel = new GeoExt.MapPanel({
        region: 'center',
        height: 400,
        width: 600,
        map: map,
        center: new OpenLayers.LonLat(6.8958,
52.22).transform(fromProjection,toProjection),
        zoom: 14
      });

            capsGrid = new Ext.grid.GridPanel({
             xtype: "grid",
  title: "Select Request/Status Map Layers from the list",
  //region: "north",
                  //width:500,
  height: 150,
  viewConfig: {forceFit: true},
  store: new GeoExt.data.WMSCapabilitiesStore({
    url: "data/getcap_1.3.0.xml",
            autoLoad: true
  }),
  columns: [
```

```
            {header: "Request/Status", dataIndex: "name", sortable: true},
            {header: "Title", dataIndex: "title", sortable: true},
            {header: "Description", dataIndex: "abstract"}
        ],
                        bbar: [{
            text: "Add to Map",
                                        iconCls: 'add',
            handler: function() {
                capsGrid.getSelectionModel().each(function(record) {
                    var clone = record.clone();
                    clone.getLayer().mergeNewParams({
                        format: "image/png",
                        transparent: true
                    });
                    mapPanel.layers.add(clone);
                    mapPanel.map.zoomToExtent(
                    OpenLayers.Bounds.fromArray(clone.get("llbbox")).transform(new
OpenLayers.Projection("EPSG:4326"), new OpenLayers.Projection("EPSG:3857"))
                    );
                });
            }
        }]
    });

                        var tree = new Ext.tree.TreePanel({
                xtype: "treepanel",
                // region: "west",
                width: 200,
                                        height: 200,
                                autoScroll: true,
                enableDD: true,
                root: new GeoExt.tree.LayerContainer({
                    expanded: true
                }),
                bbar: [{
                    text: "Remove from Map",
                                            iconCls: 'remove',
                    handler: function() {
                        var node = tree.getSelectionModel().getSelectedNode();
                        if (node && node.layer instanceof OpenLayers.Layer.WMS) {
                            mapPanel.map.removeLayer(node.layer);
                        }
                    }
                }]
            });

                            var legend = new GeoExt.LegendPanel({
```

```
                    xtype: "gx_legendpanel",
    //region: "east",
    width: 200,
    autoScroll: true,
   // padding: 5
});

                    mainPanel = new Ext.Panel({
    region: "west",
    layout: "auto",
    width: 200,
                              //split: true,
    items: [tree, legend]
});

                    var tabs = new Ext.TabPanel({
    //renderTo: 'my-tabs',
    activeTab: 0,
    items:[
        {contentEl:'tab1', title:'Volunteers'},
        {contentEl:'tab2', title:'Requesters'}
    ]
});

                    tabGrid = new Ext.Panel({
    region: "north",
    layout: "auto",
                              height: 200,
    //width: 200,
                              //split: true,
    items: [{
                                contentEl: heading,
        region: "north",
        height: 50,
        bodyStyle: {padding: "5px"}
    }, capsGrid]
});

                    /*requestPanel1 = new Ext.FormPanel({
    //title: 'Publish Your Request Maps here',
    labelWidth: 75, // label settings here cascade unless overridden
    url: 'save-form.php',
    frame:true,
    bodyStyle:'padding:5px 5px 0',
    width: 300,
    region: 'east',
```

```
    layout:'column', // arrange items in columns
    defaults: {      // defaults applied to items
        layout: 'form',
        border: false,
        bodyStyle: 'padding:4px'
    },
    items: [{
        // Fieldset in Column 1
        xtype:'fieldset',
        columnWidth: 1,
        title: 'Request details',
        collapsible: true,
        autoHeight:true,
        defaults: {
            anchor: '-20' // leave room for error icon
        },
        defaultType: 'textfield',
        items :[{
            fieldLabel: 'Owner'
        }, {
            fieldLabel: 'Title'
        }, {
                            xtype: 'textarea',
            fieldLabel: 'Description',

        }
        ]
    }]
});*/

requestPanel2 = new Ext.FormPanel({
    title: 'Acceess Request/Status Maps In your GIS Applications',
    labelWidth: 75, // label settings here cascade unless overridden
    url: 'save-form.php',
    frame:true,
    bodyStyle:'padding:5px 5px 0',
    width: 250,
    region: 'east',
    layout:'column', // arrange items in columns
    defaults: {      // defaults applied to items
        layout: 'form',
        border: false,
        bodyStyle: 'padding:4px'
    },
    items: [{
        xtype:'fieldset',
```

```
                columnWidth: 1,
                title: 'OnlineResource',
                collapsible: true,
                autoHeight:true,
                defaults: {
                    anchor: '-20' // leave room for error icon
                },
                defaultType: 'textfield',
                items :[{
                                        xtype: 'label',
                    fieldLabel: 'WMS URL',
                                                text:
'http://reqstatmap.dynu.com/geoserver/project/wms'
                                    }, {
                                        xtype: 'label',
                    fieldLabel: 'Catalogue Server',
                                                text:
'http://reqstatmap.dynu.com/geonetwork/srv/eng/csw'
                                    }

                ]
            }]
        });

        new Ext.Viewport( {
            layout: "border",
                                        //contentEl: heading,
            items: [mapPanel, tabGrid, mainPanel, requestPanel2/*, {
                                            contentEl: heading,
                region: "north",
                height: 50,
                bodyStyle: {padding: "5px"}
            }*/]
        });
    });
                </script>

    </head>
    <body>
            <!--<div id="my-tabs"></div>
        <div id="tab1" class="x-hide-display"><p>This page is dedicated for volunteers to explore
published request/status maps</p></div>
        <div id="tab2" class="x-hide-display"><p>This page is dedicated for organizations to
publish and manage their request maps</p></div>
            !-->
                <div id="heading" class="x-hide-display">
```

```
                <h1 style="font-size:125%">Request/Status map Portal</h1>
                <nav id="banner">
                <ul id = "banner">
                  <li><a href="http://reqstatmap.dynu.com/prototype/mapfnl4.html">View-
request/status-map</a></li>
            <li><a href="http://reqstatmap.dynu.com/prototype/requester.html">Make-
request</a></li>
        </ul>
                </nav>

                </div>
  </body>
</html>
```

Make-Request

```
<html>
  <head>
    <title>RS Map Portal</title>
    <link rel="stylesheet" type="text/css" href="ext/resources/css/ext-all.css">
    <script type="text/javascript" src="ext/adapter/ext/ext-base.js"></script>
    <script type="text/javascript" src="ext/ext-all.js"></script>
    <script src="openlayers/lib/OpenLayers.js"></script>
    <script type="text/javascript" src="geoext/lib/GeoExt.js"></script>
                <script type="text/javascript" src="FileUploadField.js"></script>

                <style>
                #banner ul  {
      overflow: hidden;
                      list-style: none;
                      margin-left: auto;
      margin-right: auto;
      width: auto;
      background-color: #CCFFCC;
    }

    #banner  li {
      float: left;
    }

    #banner a:link,  #banner a:visited {
      display: block;
      width: 250px;
      font-weight: bold;
      color: #0066FF;
```

```
    background-color:           #CCFFCC;
    text-align: center;
    padding: 2px;
    text-decoration: none;
    text-transform: uppercase;
  }

  #banner a:hover,  #banner a:active {
    background-color: #00CCFF;
  }

            .add {background-image:url(custom_pictures/adddata.png);
            }

            .remove {background-image:url(custom_pictures/removebutton.png);
            }
  </style>
            <script type="text/javascript">
            Ext.BLANK_IMAGE_URL = "ext/resources/images/default/s.gif";
  var controls = [];
            var app;
            var vectorLayer = new OpenLayers.Layer.Vector("Editable features");

  Ext.onReady(function() {

var fromProjection = new OpenLayers.Projection("EPSG:4326"); // WGS 1984
          var toProjection = new OpenLayers.Projection("EPSG:900913"); // Spherical

    var map = new OpenLayers.Map(null, {
      projection: new OpenLayers.Projection("EPSG:3857"),
      allOverlays: true,
      maxExtent: new OpenLayers.Bounds(-20037508.34, -20037508.34,
                  20037508.34, 20037508.34),
      units: 'm'
    });
              var osmLayer = new OpenLayers.Layer.OSM();
              map.addLayer(osmLayer);

              map.addControl(new OpenLayers.Control.Navigation(),
      new OpenLayers.Control.Attribution(),
      new OpenLayers.Control.PanPanel(),
      new OpenLayers.Control.ZoomPanel());

              mapPanel = new GeoExt.MapPanel({
      region: 'center',
      height: 400,
```

```
            width: 600,
            map: map,
            center: new OpenLayers.LonLat(6.8958,
52.22).transform(fromProjection,toProjection),
            zoom: 14
        });

                        mapPanel.map.addLayer(vectorLayer);

                        capsGrid = new Ext.grid.GridPanel({
                            xtype: "grid",
            title: "Select Request/Status Map Layers from the list",
            //region: "north",
                                    //width:500,
            height: 150,
            viewConfig: {forceFit: true},
            store: new GeoExt.data.WMSCapabilitiesStore({
                url:
"http://localhost:8085/geoserver/wms?SERVICE=WMS&REQUEST=GetCapabilities&VERSI
ON=1.1.1",
                        autoLoad: true
            }),
            columns: [
                {header: "Request/Status", dataIndex: "name", sortable: true},
                {header: "Title", dataIndex: "title", sortable: true},
                {header: "Description", dataIndex: "abstract"}
            ],
                            bbar: [{
                text: "Add to Map",
                                                iconCls:'add',
                handler: function() {
                    capsGrid.getSelectionModel().each(function(record) {
                        var clone = record.clone();
                        clone.getLayer().mergeNewParams({
                            format: "image/png",
                            transparent: true
                        });
                        mapPanel.layers.add(clone);
                        mapPanel.map.zoomToExtent(
                        OpenLayers.Bounds.fromArray(clone.get("llbbox")).transform(new
OpenLayers.Projection("EPSG:4326"), new OpenLayers.Projection("EPSG:3857"))
                        );
                    });
                }
            }]
        });
```

```
                var tree = new Ext.tree.TreePanel({
    xtype: "treepanel",
   // region: "west",
    width: 200,
                                height: 200,
                        autoScroll: true,
    enableDD: true,
    root: new GeoExt.tree.LayerContainer({
        expanded: true
    }),
    bbar: [{
        text: "Remove from Map",
                                        iconCls: 'remove',
        handler: function() {
            var node = tree.getSelectionModel().getSelectedNode();
            if (node && node.layer instanceof OpenLayers.Layer.WMS) {
                mapPanel.map.removeLayer(node.layer);
            }
        }
    }]
});

                        var legend = new GeoExt.LegendPanel({
                            xtype: "gx_legendpanel",
    //region: "east",
    width: 200,
    autoScroll: true,
   // padding: 5
});

                        mainPanel = new Ext.Panel({
    region: "west",
    layout: "auto",
    width: 200,
                                //split: true,
    items: [tree, legend]
});

                        tabGrid = new Ext.Panel({
    region: "north",
    layout: "auto",
                                height: 200,
    //width: 200,
                                //split: true,
    items: [{
```

```
                                    contentEl: heading,
        region: "north",
        height: 50,
        bodyStyle: {padding: "5px"}
    }, capsGrid]
});



var fp = new Ext.FormPanel({
    region: 'east',
    fileUpload: true,
    width: 350,
    frame: true,
    title: 'Publish Your Request Here',
    autoHeight: true,
    bodyStyle: 'padding: 10px 10px 0 10px;',
    labelWidth: 75,
    defaults: {
        anchor: '95%',
        allowBlank: false,
        msgTarget: 'side'
    },
    items: [{
        xtype: 'textfield',
        fieldLabel: 'Owner'
    }, {
                    xtype: 'textfield',
                    fieldLabel: 'Title'
                }, {
                    xtype: 'textarea',
        fieldLabel: 'Description',
                    style:{overflow:'auto'},
        width:200,
        height:150
                }, {
        xtype: 'fileuploadfield',
        id: 'form-file',
        emptyText: 'Select a Shapefile',
        fieldLabel: 'Request Map',
        name: 'shapefile-path',
        buttonText: '',
        buttonCfg: {
            iconCls: 'upload-icon'
        }
    }],
```

```
buttons: [{
    text: 'Save',
    handler: function(){
        if(fp.getForm().isValid()){
                fp.getForm().submit({
                url: 'file-upload.php',
                waitMsg: 'Uploading your shapefile...',
                success: function(fp, o){
                    msg('Success', 'Processed file "'+o.result.file+'" on the server');
                }
                });
        }
    }
},{
    text: 'Reset',
    handler: function(){
        fp.getForm().reset();
    }
}]
});

                featureGrid = new Ext.grid.GridPanel({
    title: "Update Your Request/Status Map",
    region: "south",
    height: 150,
    sm: new GeoExt.grid.FeatureSelectionModel(),
    store: new GeoExt.data.FeatureStore({
        fields: [
                                {name: "id", type: "int"},
            {name: "completness", type: "int"},
            {name: "assesment", type: "string"},
                                    {name: "priority", type: "string"}

        ],
        proxy: new GeoExt.data.ProtocolProxy({
            protocol: new OpenLayers.Protocol.WFS({
            url: "http://localhost:8085/geoserver/wfs",
            version: "1.1.0",
            featureType: "port_au_prince_feature_mapping",
            featureNS: "http://localhost:8085/geoserver/project",
            srsName: "EPSG:3857"
            })
        }),
        autoLoad: true
    }),
    columns: [
```

```
                                {header: "id", dataIndex: "id"},
        {header: "completness", dataIndex: "completnes"},
        {header: "assesment", dataIndex: "assesment"},
                                         {header: "priority", dataIndex: "priority"}

    ],
    bbar: []
});
                    featureGrid.store.bind(vectorLayer);
featureGrid.getSelectionModel().bind(vectorLayer);
                    var rawAttributeData;
var read = OpenLayers.Format.WFSDescribeFeatureType.prototype.read;
OpenLayers.Format.WFSDescribeFeatureType.prototype.read = function() {
    rawAttributeData = read.apply(this, arguments);
    return rawAttributeData;
};

                    function reconfigure(store, url) {
    var fields = [], columns = [], geometryName, geometryType;
    // regular expression to detect the geometry column
    var geomRegex = /gml:(Multi)?(Point|Line|Polygon|Surface|Geometry).*/;
    var types = {
        // mapping of xml schema data types to Ext JS data types
        "xsd:int": "int",
        "xsd:short": "int",
        "xsd:long": "int",
        "xsd:string": "string",
        "xsd:dateTime": "string",
        "xsd:double": "float",
        "xsd:decimal": "float",
        // mapping of geometry types
        "Line": "Path",
        "Surface": "Polygon"
    };
    store.each(function(rec) {
        var type = rec.get("type");
        var name = rec.get("name");
        var match = geomRegex.exec(type);
        if (match) {
          // we found the geometry column
          geometryName = name;
        } else {
            // we have an attribute column
            fields.push({
                name: name,
                type: types[type]
```

```
            });
            columns.push({
                xtype: types[type] == "string" ?
                    "gridcolumn" :
                    "numbercolumn",
                dataIndex: name,
                header: name
            });
        }
    });
    featureGrid.reconfigure(new GeoExt.data.FeatureStore({
        autoLoad: true,
        proxy: new GeoExt.data.ProtocolProxy({
            protocol: new OpenLayers.Protocol.WFS({
            url: url,
            version: "1.1.0",
            featureType: rawAttributeData.featureTypes[0].typeName,
            featureNS: rawAttributeData.targetNamespace,
            srsName: "EPSG:3857",
            geometryName: geometryName,
            maxFeatures: 250
            })
        }),
        fields: fields
    }), new Ext.grid.ColumnModel(columns));
    featureGrid.store.bind(vectorLayer);
    featureGrid.getSelectionModel().bind(vectorLayer);
}

                    function setLayer(model, node) {
    if(!node || node.layer instanceof OpenLayers.Layer.Vector) {
        return;
    }
    vectorLayer.removeAllFeatures();
    featureGrid.reconfigure(
        new Ext.data.Store(),
        new Ext.grid.ColumnModel([])
    );
    var layer = node.layer;
    var url = layer.url.split("?")[0]; // the base url without params
    var schema = new GeoExt.data.AttributeStore({
        url: url,
        // request specific params
        baseParams: {
            "SERVICE": "WFS",
            "REQUEST": "DescribeFeatureType",
```

```
                "VERSION": "1.1.0",
                "TYPENAME": layer.params.LAYERS
            },
            autoLoad: true,
            listeners: {
                "load": function(store) {
                    featureGrid.setTitle(layer.name);
                    reconfigure(store, url);
                }
            }
        });
    }

                    tree.getSelectionModel().on(
        "selectionchange", setLayer
                                );


    new Ext.Viewport( {
        layout: "border",
        items: [mapPanel, tabGrid, mainPanel,fp, featureGrid]
    });

  });
            </script>
</head>
<body>
        <!--<div id="my-tabs"></div>
    <div id="tab1" class="x-hide-display"><p>This page is dedicated for volunteers to explore
published request/status maps</p></div>
    <div id="tab2" class="x-hide-display"><p>This page is dedicated for organizations to
publish and manage their request maps</p></div>
        !-->
            <div id="heading" class="x-hide-display">

            <h1 style="font-size:125%">Request/Status map Portal</h1>
            <nav id="banner">
            <ul id = "banner">
                <li><a
href="http://reqstatmap.privatedns.org/prototype/RS_map.html">View-request/status-
map</a></li>
        <li><a href="http://reqstatmap.privatedns.org/prototype/requester.html">Make-
request</a></li>
    </ul>
            </nav>
```

```
                  </div>
    </body>
</html>
```

SLD file for feature mapping completeness in percentage

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<StyledLayerDescriptor version="1.0.0"
    xsi:schemaLocation="http://www.opengis.net/sld StyledLayerDescriptor.xsd"
    xmlns="http://www.opengis.net/sld"
    xmlns:ogc="http://www.opengis.net/ogc"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <NamedLayer>
    <Name>Attribute-based polygon</Name>
    <UserStyle>
      <Title>RS map Style</Title>
      <FeatureTypeStyle>
        <Rule>
          <Name>no data</Name>
          <Title>no data</Title>
                    <ogc:Filter>
            <ogc:PropertyIsEqualTo>
              <ogc:PropertyName>completnes</ogc:PropertyName>
              <ogc:Literal>0</ogc:Literal>
            </ogc:PropertyIsEqualTo>
          </ogc:Filter>
          <PolygonSymbolizer>
            <Fill>
              <CssParameter name="fill">#bdbdbd</CssParameter>
                              <CssParameter name="fill-opacity">0.7</CssParameter>
            </Fill>
                            <Stroke>
              <CssParameter name="stroke">#000000</CssParameter>
              <CssParameter name="stroke-width">0.26</CssParameter>
              <CssParameter name="stroke-linejoin">no</CssParameter>
            </Stroke>
          </PolygonSymbolizer>
        </Rule>
        <Rule>
          <Name>0 - 25</Name>
          <Title>>0 - 25</Title>
                    <ogc:Filter>
                    <ogc:And>
            <ogc:PropertyIsGreaterThan>
              <ogc:PropertyName>completnes</ogc:PropertyName>
```

```
      <ogc:Literal>0</ogc:Literal>
    </ogc:PropertyIsGreaterThan>
    <ogc:PropertyIsLessThanOrEqualTo>
      <ogc:PropertyName>completnes</ogc:PropertyName>
      <ogc:Literal>25</ogc:Literal>
     </ogc:PropertyIsLessThanOrEqualTo>
    </ogc:And>
  </ogc:Filter>
  <PolygonSymbolizer>
    <Fill>
      <CssParameter name="fill">#fee5d9</CssParameter>
                    <CssParameter name="fill-opacity">0.7</CssParameter>
    </Fill>
                   <Stroke>
      <CssParameter name="stroke">#000000</CssParameter>
      <CssParameter name="stroke-width">0.26</CssParameter>
      <CssParameter name="stroke-linejoin">no</CssParameter>
    </Stroke>
  </PolygonSymbolizer>
</Rule>
<Rule>
  <Name>25 - 50</Name>
  <Title>>25 - 50</Title>
  <ogc:Filter>
    <ogc:And>
      <ogc:PropertyIsGreaterThan>
        <ogc:PropertyName>completnes</ogc:PropertyName>
        <ogc:Literal>25</ogc:Literal>
      </ogc:PropertyIsGreaterThan>
      <ogc:PropertyIsLessThanOrEqualTo>
        <ogc:PropertyName>completnes</ogc:PropertyName>
        <ogc:Literal>50</ogc:Literal>
      </ogc:PropertyIsLessThanOrEqualTo>
    </ogc:And>
  </ogc:Filter>
  <PolygonSymbolizer>
    <Fill>
      <CssParameter name="fill">#fcae91</CssParameter>
                    <CssParameter name="fill-opacity">0.7</CssParameter>
    </Fill>
    <Stroke>
      <CssParameter name="stroke">#000000</CssParameter>
      <CssParameter name="stroke-width">0.26</CssParameter>
      <CssParameter name="stroke-linejoin">no</CssParameter>
    </Stroke>
  </PolygonSymbolizer>
```

```
</Rule>
<Rule>
  <Name>50 - 75</Name>

  <Title>>50 - 75</Title>

  <ogc:Filter>
    <ogc:And>
      <ogc:PropertyIsGreaterThan>
        <ogc:PropertyName>completnes</ogc:PropertyName>
        <ogc:Literal>50</ogc:Literal>
      </ogc:PropertyIsGreaterThan>
      <ogc:PropertyIsLessThanOrEqualTo>
        <ogc:PropertyName>completnes</ogc:PropertyName>
        <ogc:Literal>75</ogc:Literal>
      </ogc:PropertyIsLessThanOrEqualTo>
    </ogc:And>
  </ogc:Filter>
  <PolygonSymbolizer>
    <Fill>
      <CssParameter name="fill">#fb6a4a</CssParameter>
                          <CssParameter name="fill-opacity">0.7</CssParameter>
    </Fill>
    <Stroke>
      <CssParameter name="stroke">#000000</CssParameter>
      <CssParameter name="stroke-width">0.26</CssParameter>
      <CssParameter name="stroke-linejoin">no</CssParameter>
    </Stroke>
  </PolygonSymbolizer>
</Rule>
                  <Rule>
  <Name>75 - 100</Name>

  <Title>>75 - 100</Title>

  <ogc:Filter>
    <ogc:And>
      <ogc:PropertyIsGreaterThan>
        <ogc:PropertyName>completnes</ogc:PropertyName>
        <ogc:Literal>75</ogc:Literal>
      </ogc:PropertyIsGreaterThan>
      <ogc:PropertyIsLessThanOrEqualTo>
        <ogc:PropertyName>completnes</ogc:PropertyName>
        <ogc:Literal>100</ogc:Literal>
      </ogc:PropertyIsLessThanOrEqualTo>
    </ogc:And>
```

```
        </ogc:Filter>
        <PolygonSymbolizer>
          <Fill>
            <CssParameter name="fill">#cb181d</CssParameter>
                            <CssParameter name="fill-opacity">0.7</CssParameter>
          </Fill>
          <Stroke>
            <CssParameter name="stroke">#000000</CssParameter>
            <CssParameter name="stroke-width">0.26</CssParameter>
            <CssParameter name="stroke-linejoin">no</CssParameter>
          </Stroke>
        </PolygonSymbolizer>
      </Rule>
    </FeatureTypeStyle>
  </UserStyle>
 </NamedLayer>
</StyledLayerDescriptor>
```