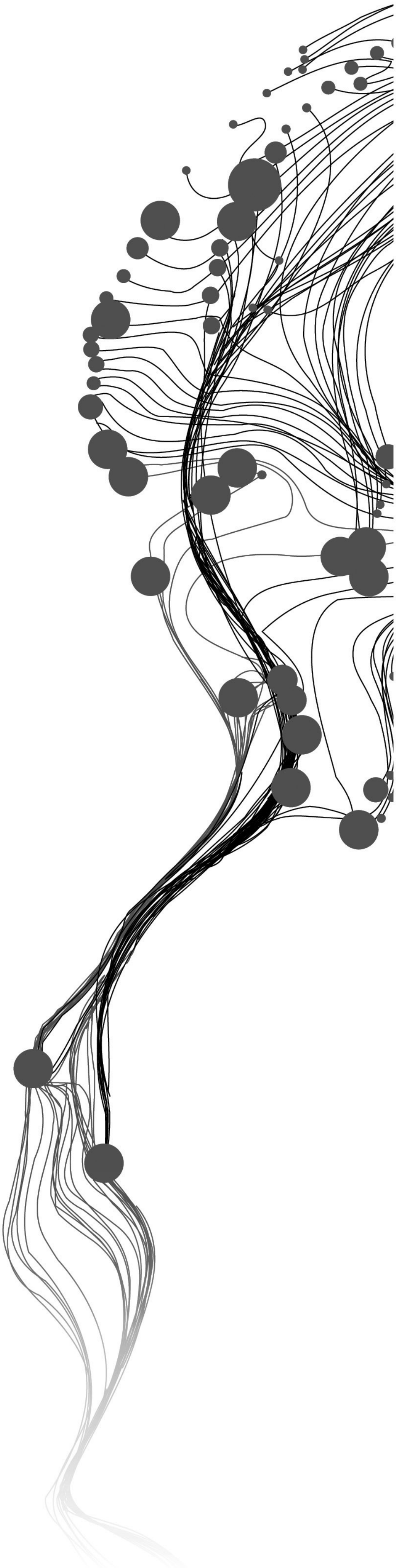
CROWDSOURCING GEOPORTALS FOR RAPID POST-DISASTER DAMAGE MAPPING.

ARIE YULFA
May, 2012

SUPERVISORS:

Dr. Ir. R.L.G. Lemmens
Drs. B.J. Köbben



CROWDSOURCING GEOPORTALS FOR RAPID POST-DISASTER DAMAGE MAPPING.

ARIE YULFA
Enschede, The Netherlands, May, 2012

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
Drs. B.J. Köbben

THESIS ASSESSMENT BOARD:

Dr. Ir R.A.de By (chair)
Drs. M.E. de Vries (external examiner)

ABSTRACT

Rapid post-disaster damage mapping is important for emergency responses after disasters. Currently, many organizations and individuals have implemented this mapping method. However, those organizations and individuals have different missions and methodologies to create the rapid damage map within limited time. These may result in different data from different sources; hence, confusion occurs. In attempts to deal with such case, a geoportal - the medium for communication and crowdsourcing - as a method to collect data becomes the potential solution to overcome this problem, without necessarily interfering with the organizational missions and individual interests in rapid post-disaster damage mapping.

This research is aimed at designing the architecture of crowdsourcing geoportals to rapidly create post-disaster damage maps by integrating different geographic data sources. For this purpose, we have developed a prototype to implement this architecture using Ushahidi, Google Map Maker, and OpenStreetMap since these applications have already adopted crowdsourcing systems, and are commonly used in creating geoportals. Phases of prototype development include analysis of users' needs, identification of what a certain geoportal can function, implementation and, finally, evaluating the designed prototype.

The designed geoportal prototype consists of four elements, namely reporting damage, creating a base map, creating a request map, and showing the latest information. We used several scenarios to evaluate if the prototype functions as expected. Through request map tool, geoportal prototype differs from other existing geoportals in creating rapid post-disaster damage maps, in the sense that it allows efficiency in mapping post-disaster damage area. The request will help contributors to prioritize their contributions and understand what and where to contribute. The result of this research shows that the design can be used as a model to take advantage of crowdsourcing in creating post-disaster damage map.

Keywords

Crowdsourcing, Geoportals, Rapid Post-disaster damage mapping

ACKNOWLEDGEMENTS

I am heavily indebted to my first and second supervisors, Dr. Ir. R.L.G. Lemmens, and Drs. B.J. Köbben for their time and willingness to support and guide me in accomplishing this thesis.

I am also grateful for the assistance that Dr. Norman Kerle has provided me in giving me the access to data of Haiti damage assessment, and concepts on post-disaster damage mapping. Dr. Ir R.A.de By has been helpful to me and I cannot thank you enough for your suggestions. Sincerely, I am very grateful for the time provided by Ms. Clarisse Kagoyire with regard to discussion about Ushahidi. My study in ITC would not have happened if it was not for the funding from STUNED, therefore I would like to extend my sincere gratitude for the chance they have given me in pursuing this study in The Netherlands.

Earnest appreciation also goes to my friends Rani, Fitri, Idham, Iday, Fesly, Tiur, Doddy, Debi, Nugroho, and the Indonesian Student Association, for your support. I am not going to forget my colleagues in ITC, and I want you all to remember if someday you plan to visit West Sumatra, you've got a friend over there. Thank you so much for the friendship.

Finally, I am greatly indebted to my family for their love and support. Last, but not least, I wish to thank my Department of Geography at Padang State University (Universitas Negeri Padang), Indonesia.

May the Almighty God bless us with peace and harmony.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1
Introduction

1.1 MOTIVATION AND RESEARCH PROBLEM

An earthquake is one of the natural disasters that can cause structural damages. Locations of the damaged area are very crucial for the efforts of rescue and immediate relief. Many organizations will support these attempts by providing maps. The Yogyakarta-Indonesia earthquake, occurred in 2006, was an example of how international organizations supported emergency responses using enormous maps, generated from satellite data, and how accurate the maps is to fulfill the intended needs (Kerle, 2010). These organizations assembled in the International Charter Space and Major Disasters to produce post-disaster damage maps deriving from satellite imageries. But, most of these data were varied or not in the same standard due to different sources and methodologies in their interpretation of the imageries.

Recently, crowdsourcing (see details in section 2.1) has become a potential source in collecting data from people to solve some tasks, such as reporting building damages or creating a base-map, in addition to satellite-based method. In general, these people are not trained surveyors but they can access Internet to contribute spatial data (Heipke, 2010). The 2010 Haiti earthquake was another evidence how crowdsourcing established a new trend in disaster damage mapping other than Charter-data processing agencies (Kerle, 2011). Doan et al. (2011) described four challenges in executing crowdsourcing. Those challenges are inviting users to the system, leading users to a certain goal, integrating users contribution, and evaluating users and their contributions.

The large amount of data produced in post-disaster damage mapping is often hard to find in a structured way and is usually poorly described and archived. Current efforts on interoperability standards of open geospatial consortium (OGC) and international organization of standardization (ISO) make it possible for organizations to share their geo-information with others based on service transaction instead of raw data transfer (Maiyo et al., 2010). To do so, spatial data infrastructure (SDI) is seen as the answer to the growing need in organizing data across different disciplines and organizations, for supporting decision-making process (Feeney et al., 2001). The key elements of any SDI are geoportals. Geoportals are gateways to organize data and services of geographic information through Internet (Maguire and Longley, 2005).

The Open Geospatial Consortium Inc (2004) sets a common Geospatial Portal Reference Architecture in efforts to support data sharing, discovery, visualization and retrieval. This architecture specifies four classes of services required to implement the geoportal using OGC interoperability standards:

1. *portal service* is the gateway to discover and access data, including management and administration facilities;
2. *catalog service* offers information about data and related services;
3. *portrayal service* focuses on mapping and styling;
4. *data service* provides data access and processing capabilities.

To implement and deploy these different class of services, OGC proposes the use of web services technology that permits access to the distributed data and services through Uniform Resource Locators (URLs). This URL-based mechanism allows the publication of standardized services over a network, typically the Internet, in spite of the implementation (e.g. data format, storage), or on which platform it is executed. This mechanism leverages the real potential of interoperability by allowing web services to be seamlessly coupled, reusable and available for a wide variety of applications.

To prevent users from overlooking geographical resources of interest when searching for geospatial data on geoportals, the arrangement of geographic information in metadata catalogs is preferably modified to enable the encapsulate data semantics in question. Nikolaos et al. (2005) recommend applying Semantic Web in the geoportal to solve this problem. They propose solutions to this problem, by implementing ontologies to manage spatial data and providing the result of spatial data management for visitors on geoportals.

Giuliani and Peduzzi (2011) propose a test for the geoportal prototype, to see how far it can meet user requirements. Different users from disaster management communities are invited to perform the test. If problems persist, they will be used to correct the geoportal. In terms of quality data, crowdsourced data contributed by the users are validated among users and repetitions will occur more frequently. In conclusion, statistically, the data is rather acceptable (Heipke, 2010).

The nature of rapid post-disaster damage mapping is it should be created in a short time and accurate to provide information on the structurally-damaged areas, for instance, during the immediate relief and rescue activities (Saito and Spence, 2004; Kerle, 2011). Broek et al. (2009) also specified that information on damages should be available within hours (or up to a few days for larger areas) and contains an overview rather than a detailed damage information. . In preparing a rapid damage map, four processes are involved (Saito and Spence, 2004):

1. Collecting data, such as footprints, type and function of buildings.
2. Classification of the level of damages.
3. Production of damage maps by grid or individual building.
4. Maps indicating road purposes.

Combining these processes is assumed to be beneficial in the preparation of rapid post-disaster mapping pertaining to the immediate relief and rescue activities after a disaster takes place.

In general, the specific properties of post-disaster mapping process in constructing of geoportals is considered very challenging: the data is produced within a very limited time frame, by a multitude of organizations from different places, using different techniques and technologies. The urgency and rarity of these mapping efforts means that meta-data descriptions, production of catalogue, archive, and adherence to standards are not priorities.

Consequently, the challenge is to come up with methods and techniques to construct a geoportal for rapid post-disaster damage mapping. This geoportal should provide functionalities and information for people who want to contribute and need information related to emergency responses after an earthquake.

1.2 RESEARCH IDENTIFICATIONS

This research identifies the desired objectives to solve problems and these objectives generate some questions so that the research employs proper methods.

1.2.1 Research objectives

The main objective is to design an architecture of crowdsourcing geoportals to rapidly create post-disaster damage maps by integrating different geographic data sources. In order to achieve the main objective, several objectives are divided into four sub-objectives:

1. To determine relevant user types for rapid post-disaster damage mapping.
2. To determine criteria and actions in integrating map sources quickly.
3. To identify the key crowdsourcing concept for rapid post-disaster damage mapping.
4. To design and implement proof of concept software stack for the geoportal client/server architecture.

1.2.2 Research questions

For sub-objective 1:

1. What types of users can benefit from the geoportal for rapid post-disaster damage mapping?

For sub-objective 2:

1. What problems will arise on integrating map sources ?
2. How to solve these problems so that integrating process can be done ?

For sub-objective 3:

1. What is the principal concept of crowdsourcing for rapid post-disaster damage mapping?
2. How to combine crowd-sourced information based on different attribute classifications?

For sub-objective 4:

1. What is the design framework of geoportal for post-disaster damage mapping?
2. What is the concept software stack for a prototype of the geoportal client/server architecture?
3. How to implement proof of concept software stack for the geoportal client/server architecture?

1.2.3 Innovation aimed at

This research aims to combine information from different providers and crowdsourced data to map post-disaster damage areas rapidly.

1.2.4 Related work

A number of studies have been carried out and recommended several approaches to tackle problems of rapid post-disaster damage mapping by combining geoportals and crowdsourcing concepts. Saito and Spence (2004) defined how to rapidly create a damage map from satellite imageries and ground-observed data that can visualize the distribution of damage and type of damages observed from buildings to an accurate level of map to properly support early emergency and rescue planning. Similar research also did by Broek et al. (2009) and they added damage assessment on this map but still rely on the interpretation of satellite imagery experts.

Following 2010 Haiti earthquake, Kerle (2011) elucidated, a crowd of people become a source of data, including web 2.0 technology, and the implementation of these two things has helped the maturity of collaborative mapping on rapid post-disaster damage mapping and change it into a new trend. Heipke (2010) reviewed researches about this and found that crowdsourcing is a potential methodology to apply on collecting geospatial data. Maiyo et al. (2010) presented a prototype of geo web service architecture for a collaborative post-disaster damage mapping system.

Geoportals as gateways of geographic information and services through World Wide Web technology can facilitate and coordinate data sharing, access, and use among different partners. Giuliani and Peduzzi (2011) demonstrated, a geoportal is accommodating in the aspects of global data provision and sharing about risks to natural hazards. This application is operable if it observes the clearly defined SDI conceptual model. As an attempt to enhance the ability of geoportals in offering meaningful data, Nikolaos et al. (2005) established semantic web on geoportals. Their methods are based on the organization of the geo-data on semantic level through proper geographic ontologies, and the exploitation of this organization through the user interface of the geoportal.

The launch of the Ushahidi platform in Haiti demonstrated the implementation of how to use crowdsourced information and map it for emergency response (Heinzelman and Waters, 2010). The crowd could input information through short message service (sms) and Internet browsers. These participations would be presented on a map and a downloadable csv-format file. Ushahidi used products of Google Map and Openstreetmap as their base maps. Google (2011) and OpenStreetMap (2011) provide an updating mechanism for their maps through the implementation of collaborative mapping.

These articles contribute to this research due to the fact that they discuss rapid damage mapping through geoportals by utilizing crowdsourced information. But, the focus is on how the combination of crowdsourcing and geoportals can rapidly produce post-disaster damage maps that comply with OGC standards.

1.3 METHOD ADOPTED

In order to meet the above-mentioned research objectives, we did the following approaches:

1. Literature review involves researches on principles of crowdsourcing, geoportals, semantic web, and rapid damage mapping to build the basic concepts for designing rapid post-disaster damage mapping mechanism.
2. User requirement analysis is to determine relevant users who utilize the geoportals. This step will analyze who will be the potential users of this geoportal and what their demands are.
3. Design a first prototype of a skeletal geoportal based on literature review phase. This prototype will be embedded with the user requirement analysis.
4. Determine the criteria for rapid post-disaster damage mapping based on the user requirement analysis and the first prototype.
5. Design the second prototype based on criteria previously determined with the datasets.
6. Test and evaluate the designed mechanism on geoportal.

1.4 STRUCTURE OF THE THESIS

The thesis outline is arranged as follows:

- Chapter 1 (Introduction) introduces motivation, problem statements, and objectives of where they should be achieved in this research.
- Chapter 2 (Needs for rapid post-disaster damage mapping) provides basic concepts of crowdsourcing, geoportal, rapid post-disaster damage mapping. These concepts will be used to formulate the user requirement analysis.

- Chapter 3 (Case study) describes the 2010 Haiti earthquake as a reference concerning situations and efforts made by the people to organize data in creating a rapid post-disaster damage map.
- Chapter 4 (Design a geoportal prototype) describes the process of how to design a geoportal prototype with regard to user requirement analysis.
- Chapter 5 (Implementation and testing) describes the implementation of the design and test the design if it fulfills user requirement analysis.
- Chapter 6 (Results and Discussion) provides the results of the implementation of the prototype and discuss how the result can meet the objectives.
- Chapter 7 (Conclusion and recommendations) concludes the research and gives recommendation for future works, which are not covered in this research.

Chapter 2
Needs for rapid post-disaster damage mapping

2.1 CROWDSOURCING IN GEOINFORMATION

2.1.1 Concepts of crowdsourcing

According to Howe (2006), crowdsourcing is an effort to accomplish some tasks done by people in different locations or different knowledge levels related to the tasks given. Crowdsourcing is not really a new principle to request people in accomplishing certain tasks. There are other terms, which have a similar meaning to crowdsourcing, such as user-generated content, collaborative systems, mass collaboration, human computation and others, related to the participation of civilian (Doan et al., 2011). For this research, the term of crowdsourcing is used to describe how the systems to guide people in performing several tasks, viz., mapping and reporting building damages.

Doan et al. (2011)) said that there are four challenges to encounter in leveraging crowdsourcing for the completion of any given tasks. Those challenges are how to recruit contributors, what they can do, how to combine their contributions, and how to manage abuse. In general, building a crowdsourcing system will consider the *degree of manual effort*, *role of human users*, and *standalone versus piggyback architectures*. *Degree of manual effort* means the way for a developer of the crowdsourcing system to decide the extent of the system when it runs in an automatic mode and how much efforts the users or the owners have to do to overcome those challenges. *Role of human users* has four types that are slaves, perspective providers, content providers, and component providers. These roles will help the developer to decide how to recruit users where users will have the opportunity to play multiple roles. *Standalone versus piggyback* means how the crowdsourcing system will be built - on a standalone system or attached to a well-established system.

There are some solutions that Doan et al. (2011) proposed to solve the challenges encountered in the crowdsourcing systems. For the first challenge (how to recruit contributors), solutions that they proposed are:

1. Requiring users to contribute into the system.
2. Giving money to users for their contributions.
3. Asking for volunteers.
4. Requiring users to contribute into the crowdsourcing system because of using a service from another system.
5. To manipulate users traces of a well-established system.

Other than inviting users, another important thing is how to retain users. To do this, Doan et al. (2011) proposed by providing instant gratification, an enjoyable experience or a necessary service, and reputation establishment. For example, immediately giving the result of what users have contributed to the system.

For the second challenge (what they can do). Solutions to solve this challenge are (Doan et al., 2011):

1. Classifying users based on cognitive contribution. For example, low-level users (i.e. guests) tend to give easy contribution, such as answering simple questions.
2. Measuring the impact of contributions on the crowdsourcing systems. For example, editing some words on a web page will create small impact than editing a rule on the web page.
3. Distributing tasking between human users and machines to contribute into the crowdsourcing system. Not all contributions that are input easily by human users can also be done similarly by the machines, for example matching two images on the crowdsourcing system.
4. Designing an easy interface for users to contribute.

For the third challenge (how to combine their contributions), Doan et al. (2011) said that there are two approaches in handling this challenge, namely

1. The existing automatic solution determines the weighted scores graded from users.
2. Manual dispute management works based on users' argument.

Mostly, the second approach is more preferable in the current crowdsourcing systems. Although many of the crowdsourcing systems do not combine the contributions from users, complex crowdsourcing systems combine users' contributions.

For the fourth challenge (how to manage abuse), the concern is to evaluate users and their contributions. Three combination techniques are employed to solve it: block, detect, and deter. *Block* means filtering users who can contribute certain data into the database system. *Detect* means how to check users and their contributions automatically or manually. *Deter* means banning users and their contribution, which are not reliable. But, these techniques cannot perfectly manage malicious users and worthless contributions. To do so, the system should provide an undo procedure.

2.1.2 Crowdsourcing in geoinformation

In geoinformation context, crowdsourcing actually has been implemented to collect geoinformation (geographic information) for map production. People will be invited to contribute this information with a current thematic task, such as post-disaster events to acquire damage maps. Goodchild and Glennon (2010) said, quality level of geoinformation from countless observers is logically higher than a few observers. Heipke (2010) described that crowdsourcing in geoinformation refers to contribution from large groups of people, who mostly do not have mapping knowledge, but are supported by geo-referencing and web 2.0 technologies to create geospatial data.

Geo-referencing techniques determine the location of geospatial data from people's contributions into a particular spatial reference system. For example, using global positioning system (GPS) or digitizing data on top of georeferenced images. The web 2.0 technology enables people to mash up, collaborate and share information via web sites. Data sources used in crowdsourcing system are mainly GPS tracks, georeferenced images and local knowledge. Those data will lead users to a recognized location where they want to create a map (Heipke, 2010).

How is the quality of geographic information data from crowdsourcing systems? This question is frequently discussed and many researchers try to unravel. Heipke (2010) said, mostly crowd-sourced data are contributed from people who have local knowledge on a particular area. This local knowledge is beneficial to decrease errors of crowdsourced data. He observed, data from OpenStreetMap, which implements crowdsourcing, and data from National Mapping Agencies by considering standardized quality parameters such as completeness, up-to dateness, relative and absolute geometric accuracy, attribute correctness, logical consistency. He found that, statistically, repetitions of data contributions will diminish errors. Goodchild and Glennon (2010) stated that the geographic information quality depends on what the purpose of these data. For example, a 15-m-error of a street position will have no significant impact to a car navigation system, when compared to errors on a georeferenced satellite image.

2.2 GEOPORTAL

2.2.1 Concepts of geoportals

In 1993, the US National Research Council introduced a term spatial data infrastructure (SDI) to standardize geographic information access. According to Nebert (2004, p. 8), SDI is defined as "the relevant base collection of technologies, policies and institutional arrangements that facilitate the availability of and access to spatial data". Nebert (2004) defines a gateway needed as an interface between SDI provider and SDI client.

Maguire and Longley (2005) said that geoportals are gateways that organize geographic data and services, such as directories, search tools, community information, support resources, data and applications through web technologies. He grouped geoportals into two: catalog geoportals and application geoportals. Catalog geoportals offer publishing and accessing geographic information. Application portals provide geographic web services such as routing and mapping services. In addition, he described the role of a geoportal in an SDI (Figure 2.1).

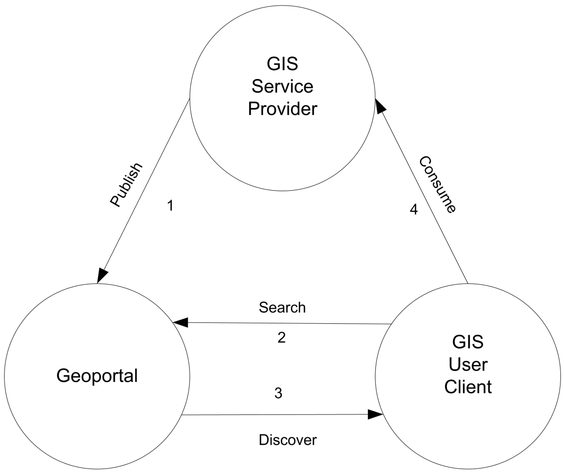


Figure 2.1: The role of a geoportal in an SDI (Source: Maguire and Longley, 2005).

This diagram is similar with the publish-find-bind concept (Open Geospatial Consortium Inc, 2004). This concept shows how services communicate within the geoportal. A geoportal plays a role as a broker that registers services published by a GIS service provider (step 1). This geoportal provides registered services for a GIS user client, as well, to search what services requested (step2). Geoportals will respond to this request and offer services based on some criteria (step 3). Subsequently, GIS user client can use such services (step 4).

2.2.2 Implementation of geoportals for disaster responses

Putra et al. (2010) designed a geoportal for evacuation planning for Mount Merapi disaster in Sleman regency, Indonesia. Reasons behind this geoportal were to synchronize coordination between local departments of Sleman government in risk disaster management activities and to support the Sleman government's spatial data infrastructure. They built this geoportal based on several steps: collecting information on the existing SDI and a procedure of disaster risk management, identifying user requirements, designing a prototype, and implementing the prototype in a simulated evacuation process, conducted by three different department officers. As a result, this geoportal fulfilled their needs to collaborate in risk disaster management activities.

In South America, the Andean region (Bolivia, Colombia, Ecuador, and Peru) also proved that sharing spatial data through a geoportal is helpful for them in disaster risk management. This geoportal is the key point of multinational SDI in this region. It is part of the Andean Information System for Disaster Prevention and Relief (SIAPAD). This geoportal provides a thematic search engine and a geographical viewer . Users are defined into two groups: those who are familiar with risk disaster management activities and general public (Molina and Bayarri, 2011).

2.3 RAPID POST-DISASTER DAMAGE MAPPING

According to the United Nations International Strategy for Disaster Reduction (UNISDR)(2009, p. 9), "disaster is a serious disruption of the functioning of a community or a society involving widespread human, material, economic or environmental losses and impacts, which exceeds the ability of the affected community or society to cope using its own resources". An event is considered as disaster if natural hazards (e.g. earthquakes, volcanic eruptions, hurricanes) occur in populated area and cause damage to the community or society (Westen, 2011).

To provide description of latest situation following a disaster, damage mapping will be needed as soon as possible by implementing various methods in mapping. Remote sensing, in most cases, is most likely used as the solution to meet this need (Kerle, 2011). The reasons are that it can capture ground situation from the air in a relatively short period of time and cover a large area than ground-based mapping (Saito and Spence, 2004; Broek et al., 2009; Kerle, 2011; Voigt et al., 2011).

For damage classification, European Macro Seismic Scale 98 (EMS1998) is one of references that has been used to build damage classification. It has five grades of building damage (see the left image in figure 2.2). In the aftermath of the 2010 Haiti earthquake, satellite-based mapping communities, such as The UN Institute for Training and Research (UNITAR), Operational Satellite Applications Programme(UNOSAT), and the Center for Satellite based Crisis Information (ZKI) of the German Aerospace Center (DLR), implemented the magnitude of the building damage into four grades instead of five grades in EMS98's scale (see the right image in figure 2.2. They reduced the grade because of the limitation of information captured in satellite imageries. The second grade (moderate damage) is difficult to be interpreted visually. Visual interpretation is preferred by the community to produce damage maps because of interpretability, reliability and timeliness (UNOSAT, 2010; Voigt et al., 2011).

Saito and Spence (2004) used two methods in their experiment, visual interpretation and texture analysis, to map building damage from satellite imagery with a 0.5 m positional accuracy. The result for the visual interpretation method was only able to classify damage more than the third grade of EMS 98. Texture analysis was not successful to determine building damage classification, but it worked for non-buildings in the image. For visualization of building damage maps, they proposed two types of visualization: a grid-based damage map (extensive damage area) and pinpoint maps (a small damage area).

Spatial literacy outside professional mapping communities is potential resources for post-disaster damage mapping (Kerle, 2011). If these potential resources can be utilized with the growing of Web 2.0 technology it will definitely be easy for professionals and lay persons to collaborate. Google Map Maker (GMM) and OpenStreetMap (OSM) are two examples of the Web 2.0 products which can provide base layers of geofeatures. These products proved how collaborative mapping was implemented for post-disaster damage mapping in the aftermath of 2010 Haiti earthquake.

'Rapid' post-disaster damage mapping refers to producing building damage map in a relatively

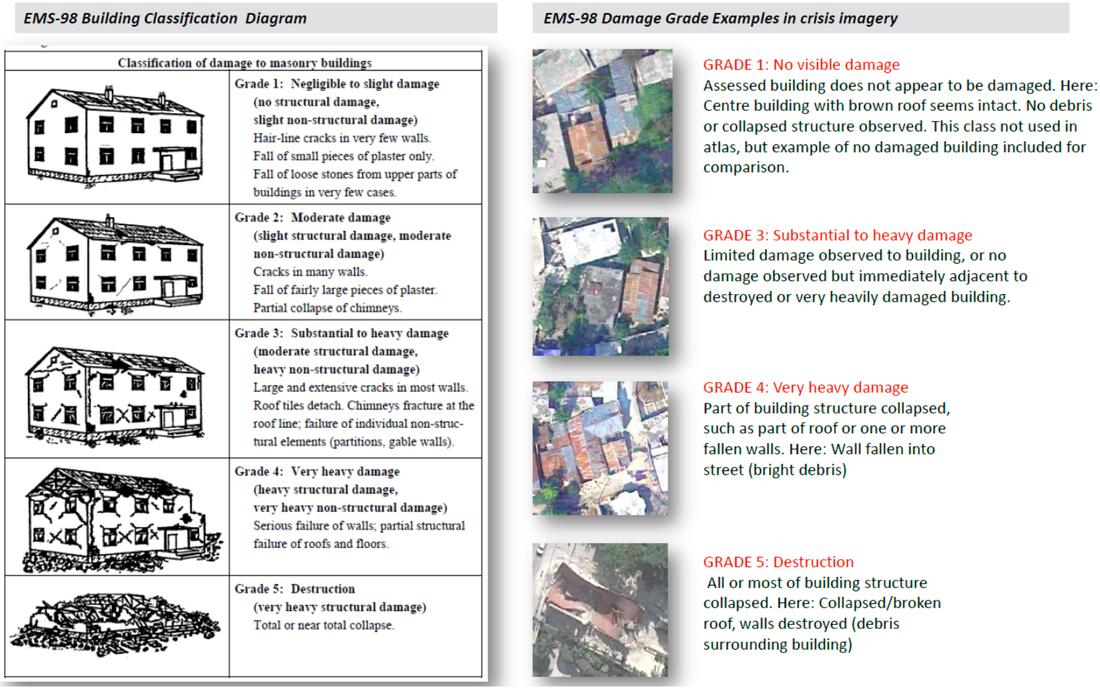


Figure 2.2: Damage classification: EMS 98(left) and UNOSAT-ECJRC-GEOCAN adapted from EMS 98(right) (Source: EMS,1998; UNOSAT,2010).

short time period. Saito and Spence (2004) defined rapid as a time range to produce damage information in two days following the earthquake struck. Rapid is to provide information in a few hours up to a few days, based on the extent of the destruction area (Broek et al., 2009). The rapid response from damage mapping communities actually related to support relief actions in mitigation. Rapidness in damage mapping also brings a consequence, which is to balance the accuracy and detailed information on one hand, and a quick response on the other (Voigt et al., 2011).

2.4 SEMANTIC WEB IN GEOPORTAL

Semantic Web is introduced by Berners-Lee et al. (2002). In this article, he elaborates on Semantic web, which refers to direct or indirect web data processing through machines, generating higher degree of automation in developing data through well-expressed mode. The presently available technologies in developing Semantic Web: eXtensible Markup Language (XML) and Resource Description Framework (RDF). With XML, users can create random structure to their documents but XML does not state any meaning regarding the structure. On the contrary, RDF expresses meaning that are encoded in sets of triples, as found in elementary sentences, such as subject, verb, and object. Ontology is the very suitable prospective method to achieve mutual understanding [i.e., the meaning of Semantic web data] between humans and computers. This is due to the reason that Ontology bears taxonomy (classifying classes of objects and relation among them) and logical rules.

According to Nikolaos et al. (2005), advantages of implementing Semantic web-based geoportals arrangement are apparent. They affirms these advantages of semantic-based geoportal after implementing it in the efforts to provide natural disasters information. Below are the outcomes:

- The suggested arrangement of geoportal metadata catalogs presents the means to gain the semantics of data in question.
- It enables homogeneous integration and utilization of diverse geo-ontologies (RDF Schemata), where every one of them describes the geographical metadata with their respective set of metadata.
- Implementation of Semantic query languages diminishes duration of data searching by users.

In another article, Nikolaos et al. (2009) state what problems are likely to occur in the absence of semantic-based web geoportal, as described in the following issues

- It is difficult for users to determine criteria with regard to data searching.
- Standards of geospatial metadata elements limit the clarity of queries.
- Keyword-based techniques of queries are incapable of encapsulating the semantics of information and, as a result, the quest to obtain knowledge will encounter semantic heterogeneity.
- Specific elements of geospatial metadata standard [e.g. ISO 19115 of FGDC] are very crucial for the result of querying.

To cope with these problems,they propose two approaches

- Describing the geoportal information through ontology-based metadata, which is written in RDF.
- In order to enhance users' navigation in the geoportal interface the arrangement or organization of ontology-based metadata is employed.

2.5 ONLINE INCIDENT ROOM

According to Oxford dictionaries (http://oxforddictionaries.com/), "incident room is a center, set up by the police, to coordinate operations connected with a particular crime, accident, or other incidents". If we use this definition in the context of rapid post-disaster damage mapping, then we determine online incident room as a place where coordination between users and producers happens in creating a damage map through a geoportal.

2.6 USER REQUIREMENT ANALYSIS

In connection to the shortcomings of the existing geoportals after 2010 Haiti earthquake (see Section 3.1), there are actually organizations who intend to be involved in disaster management. These organizations need help from others to collect data, such as reporting damage, or creating base maps. For this reason, it is obvious that they need a medium to communicate with others. Through this medium, they can determine the area where external assistance is needed and inform others what are the immediate needs.

Outside this organizational circle, there are countless other people, who wish to participate, in sharing, or creating data. Therefore, these prospective volunteers should be invited to the related medium, where they can fulfill the request needed by such organizations. The medium will also provide instructions to lead volunteers on how to deliver assistance. In addition, this medium provides a facility that reports the extent of collaborative work between volunteers, organizations, and other related parties. This way, everybody knows the progress or status of their request and contribution, which is important on setting priorities (further actions) in certain area(s). Semantic

Web was not available in geoportal, despite the fact that research conducted by Nikolaos et al. (2005) prove that Semantic Web in geoportal will help users in searching information on semantic level.

In accessing the geoportal, the developers chose a thin client application for users to access their services. The reasons behind this were to accommodate any users with varied knowledge of computer technology, and the contributors are volunteers, thus they would not have to spend all their time with respect to their participation, which meant there would be no complex analysis on the geoportal (no functionality for spatial analysis).

Rapid and trustworthy information of ground situation, following the disaster, is crucial for stakeholders in connection to their action plans (Broek et al., 2009). For example, emergency response officers and humanitarian aid need the information of damage locations to evacuate the victims and to distribute help (Saito and Spence, 2004). In addition, according to Molina and Bayarri (2011)), disaster management should involve those who are familiar with this aspect and lay persons. This is the reason why rapid damage mapping exists.

The information, such as locations of disaster, victims, refugees, magnitude, and impact of disaster is needed in a short time frame. Other than the above-mentioned information, another important issue to consider is that rapid post-disaster damage mapping requires information on how to share the data and who will manage the information.

The users of rapid damage mapping may play different roles, viz., data provider or data user. For example, during 2010 Haiti earthquake, there were several international institutions that performed these two roles, such users are the World Food Program (WFP), the Federal Office of Civil Protection and Disaster Assistance of Germany (BBK), and the Monitoring and Information Center of the European Commission (MIC), etc.

Maps are needed to facilitate the need to be informed of damage distribution and the situation for different users (see Figure 3.1). It is frequently encountered that some stakeholders create different maps in different ways, which contain similar information (see Figure 3.2). In order to reduce excessive mapping and prevent confusion on the part of user community, it is obligatory to have synchronization, standardized processing, map production with clear quality assurance, procedure of processing and analysis using cross-validation of data sharing, and visualization rules (Voigt et al., 2011). As gateways, portal can be employed to overcome the problem of coordination. Portals are developed with their own specialties, such as Ushahidi, to gather damage reports; Google Map Maker and OpenStreetMap to create base map from the people. Users need these functionalities, especially if they wish to contribute information that relates to rapid post-disaster damage mapping. Portals are used to support rescue operation planning and coordination. They are usually updated to combine new information and to refine damage assessment. A number of websites and platforms that host data, and maps, usually cause map and satellite imagery excess. Users will have difficulties to determine the best and most accurate satellite-based maps because they are available in large quantity (Voigt et al., 2011). In addition to accuracy, rapid damage mapping should provide information quickly.

To overcome the problem in coordination, there have to be a guideline and rules of engagement in collaborative rapid emergency mapping because the users of the rapid damage mapping vary from professionals people who are not familiar with familiar with rapid damage mapping (Voigt et al., 2011). The guideline should first include the scope and the users in question. Some elements that should be included in the user-guidelines are the trigger, request, communication in the mechanism, and capacities of the guideline itself. Examples of such elements are guidelines for simple and basic handling, satellite data processing, interpretation standard, legends for main damage assessment, and topographic features. This guideline plays important role in coordination, due to the fact that rapid damage mapping involves many users. Moreover, the guideline plays a role as a quality control measure. The information to be included onto the map should follow the standards, especially in basic visualization and mapping standards, because they are important, considering that users from all over world differ in culture and communication styles. The guideline facilitates the same media in communicating and exchanging data.

2.7 CONCLUDING REMARKS

Implementing crowdsourcing for rapid post-disaster damage mapping is the potential solution. In performing this, a geoportal can be used to integrate the crowdsourcing effort with others, such as satellite-based mapping, prepared by professionals. All these efforts are intended to meet user's requirements, such as distributional spread of damaged areas, data verification, and priority of immediate relief actions.

Chapter 3

Case study

3.1 EARTHQUAKE IN HAITI

Haiti was jolted by an earthquake on Tuesday, January 12, 2010, with the magnitude of 7 on the Richter scale (www.usgs.gov). Following the disaster, Haiti government was unable to respond to the needs of the people because government infrastructures were destroyed. This earthquake ruined more than 97,000 houses and claimed about 300,000 lives. The earthquake was a result of eastward movement of the Caribbean plate against the North America plate (USGS, 2011a).

Many organizations responded to it with their own specialties. For example, rapid mapping organizations, such as International Charter Space and Major Disasters, activated upon requests from relief organizations on January 13, 2010 to map the situation. This map helped relief organizations to plan their actions. Figure 3.1 is one of rapid post-disaster damage mapping products produced by DLR/German Aerospace Center (a member of International Charter Space and Major Disasters) . This map presented the information on the distribution and the extent of damages according to EMF98's scale and visualized in 250 m x 250 m grid cells. (Voigt et al., 2011).

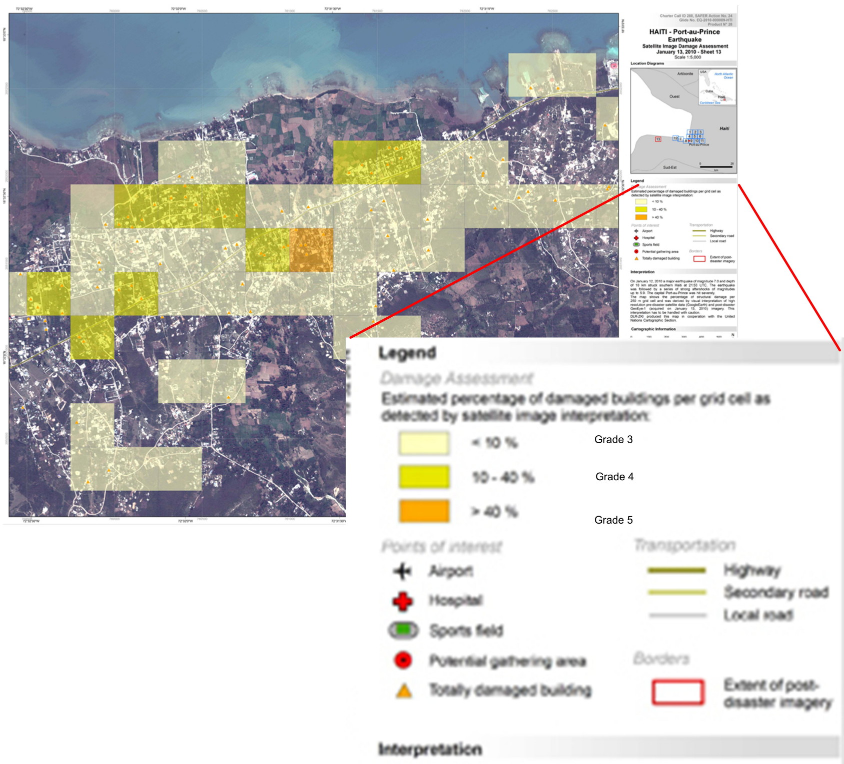


Figure 3.1: The satellite damage assessment in Haiti-Port-au-Prince on January 13, 2010, produced by DLR with 3-damage scale considered EMS 98 damage scale. (Source: www.disasterscharter.org).

Other organizations built portals to provide data and facilitate users in contributing data to support emergency response activities in the aftermath of the earthquake. These portals helped users in presenting and enabling access to the data. For example, (1) USGS use Hazards Data Distribution System (HDDS) portal, where people could access satellite imageries from different agencies. It offered pre- and post-event imageries of damaged areas, caused by disaster, from all over the world (e.g. the 2010 Haiti earthquake) (USGS, 2011b; Duda and Jones, 2011). (2) Ushahidi, together with Tuft University, employed Ushahidi as the platform to build portals for reporting damages in Haiti. People were invited to report events; (3) Google and OpenStreetMap (OSM) created portals containing base maps and provided services to users in contributing spatial data (Kerle, 2011).(4) Image Cat Inc. and Earthquake Engineering Research Institute (EERI) built Virtual Disaster Viewer (VDV) as an online platform for GEO-CAN (Global Earth Observation - Catastrophe Assessment Network) community to quantify building damages in Port-au-Prince within less than a week for the World Bank (Aardt et al., 2011). Examples of these portals can be seen in Figure 3.2. Next up, we are going to discuss the shortcomings of each existing portal employed during Haiti earthquake.

HDDS-USGS and VDV

These portals merely provide data for viewing, while facilities for editing or adding information are not available for lay persons, despite the fact that contribution from these people are potential sources in collecting data to map the damages.

HAITI USHAHIDI

1. Facility for users who want to participate in producing base map is not available.
2. Crucial directions on prioritized area(s) to work on, especially for those who want to collaborate in reporting building damages, are not available as well.

SUPPORT DISASTER RELIEF-GOOGLE AND HAITI CRISIS MAP-OPENSTREETMAP

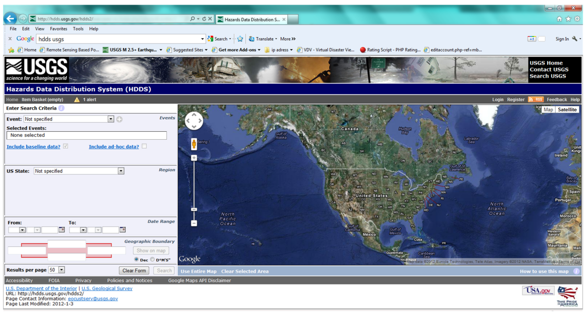
1. Users cannot report damaged buildings because the related facility for this purpose is not offered.
2. Facility for collaborative damage mapping is not available for users. For example, one organization might consider an area as a priority. For this purpose, the organization requests help from people to map the area in question.

In these geoportals, we do not find others that implement Semantic Web, while the implementation of Semantic Web, according to Nikolaos et al. (2005),may help users to search for data on a semantic level.

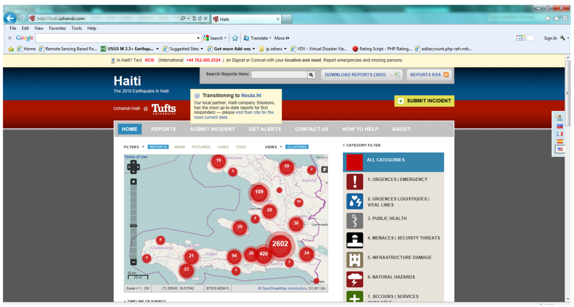
The aforementioned brings us to a question, "For the sake of time-saving, could we create one geoportal where the existing others can still be utilized, instead of building one new geoportal with all functionalities adopted from these portals?" Actually, these shortcomings may be improved if all concepts of the existing geoportals are combined into one so as to produce better geoportals that generate more reliable information in rapid post-disaster damage maps.

3.2 USE CASE SCENARIOS

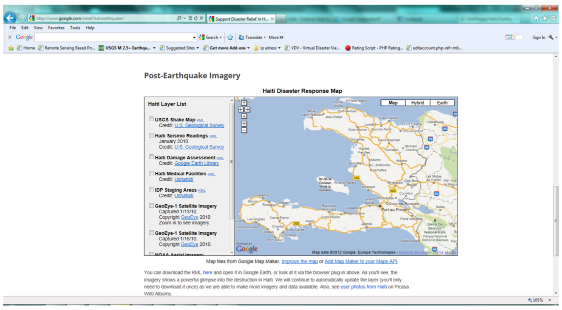
A geoportal is platform where providers and users communicate. Providers and users are dependent on their needs and ability to supply information through geoportals. Though fictive, the parallel scenarios below are created to understand how communication runs, as if it is an online



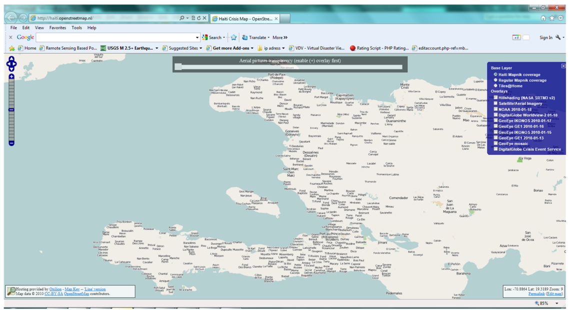
(a)



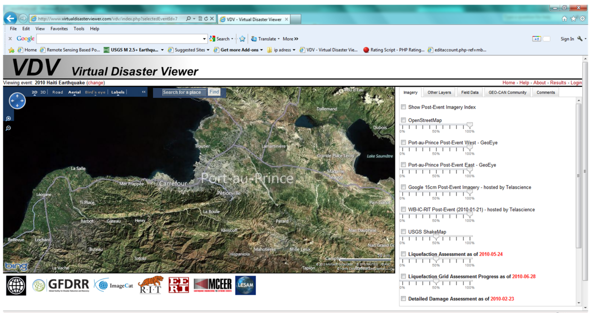
(b)



(c)



(d)



(e)

Figure 3.2: (a)HDDS-USGS, (b) Ushahidi Haiti, (c) Support Disaster Relief in Haiti- Google, (d) Haiti Crisis Map-OSM, (e) Virtual disaster viewer (Source: (USGS, 2011b; Ushahidi, 2010; Google, 2011; OpenStreetMap, 2010; Imagecat, 2010).

incident room following the earthquake (see Section 2.5 Where Oxford dictionaries defines "incident room is a center, set up by the police to coordinate operations connected with a particular crime, accident, or other incidents.") Thus, we may assume or consider online incident room as a place where coordination between users and providers takes place in creating a damage map through a geoportal. To clarify this point, the following scenarios try to explain this analogy:

Scenario 1 (For announcing a geoportal)

A few hours following Haiti earthquake, on January 12, 2010, administrators set up a geoportal. The geoportal link is then attached to social media and other public media. Twitter is an example of how administrator invites people to contribute data to create rapid post-disaster damage map for the purpose of relief actions. Administrators send a geoportal link and messages with hashtag haiti_eq (#haiti_eq) to Twitter. This action will likely create popular topic and people will be informed about the website where they can contribute.

Scenario 2 (For reporting damage)

DLR, a member of satellite-based damage mapping communities, needs information about damaged buildings (i.e, the location and damage scales of buildings) from people to verify their satellite-based damage mapping. To do this, DLR delineates a request map through the geoportal from a web browser and invites people to report the damaged buildings in the requested area (see Figure 4.2 Line 15, 16 and 17). On the other side, Asterix, for example, a local inhabitant as one of crowdsourced-data suppliers, finds a geoportal link from Twitter and visit the geoportal. From this geoportal, he will see some outlined directions to participate in reporting damaged buildings, such as a request map, where he can report damages that need prioritizing, and a status map where he can see the progress of contributions made by others. Ushahidi is used as a tool for visitors to report the damaged buildings and this tool also offers the result of users' entries to be downloaded by DLR.

Scenario 3 (For creating a base map)

UN-Minustah, an official institution of UN in Haiti, visits the geoportal and delineates a request map, expecting people to map roads in certain area(s).Another instance is, Harry, a lay person who is active in social media but lives outside Haiti, visits the geoportal. Fortunately, he has the interest on mapping. After reading some directions and a request for digitizing roads, he decides to help by digitizing roads through an on-line map editor, such as Google Map Maker or OpenStreetMap. Then, the road maps, mapped by site visitors, can be downloaded via Google Map Maker or OpenStreetMap.

3.3 REQUIREMENTS DEFINITION

Considering user requirement analysis and use case scenarios, we define requirements to be implemented in geoportals. Requirements Definition tells us about specifications we should implement (Arlow and Neustadt, 2005). There are two types of requirements: functional and non-functional. The scenarios in Section 3.2,formulate the requirements that need to be adopted into geoportals for rapid post-disaster damage mapping.

Functional requirements: Functional requirements are statements of what a geoportal can do or what functionalities geoportals should have. Below, are the functional requirements of what geoportal should be able to perform:

- providing a request map: it gives users ability to delineate an area that they need other users to help them on that specific area.
- providing a status map: it informs users on the current work that has been completed by other users.

- providing a reporting damage tool: this is intended for people who want to report a damaged building.
- providing a map editor: people who want to participate in creating a base map layer can use this tool.

Non-functional requirements: these are requirements, which are specific property of the system. The followings are to be implemented when we determine non-functional requirements of geoportal:

- For a request map: We will need editing tools on an interactive web mapping and a database to store the request map. To publish what have been requested, we would also need a map server.
- For a status map: We will retrieve information from the database where the reports and base maps are stored.
- For a reporting damage tool: This tool should have a mechanism to adopt different categories and implement the crowdsourcing systems.
- For a map editor: it has a mechanism to process data from crowdsourcing where results can be used by other tools as base map layers.

3.4 CONCLUDING REMARKS

The 2010 Haiti earthquake is used as a case study to describe how people from all over the world should respond properly to a natural disaster. There are relief organizations and individuals who want to contribute to help Haiti. In this chapter, they provide information concerning situation on the ground, following the earthquake. Some scenarios, formulated into definitions, have described the relief efforts in several requirements.

Chapter 4
Design of a prototype

4.1 SYSTEM ARCHITECTURE

System architecture needs to consider user requirement analysis (see Section 2.6), shortcomings of the existing geoportal systems (see Section 3.1), and user case scenarios (see Section 3.2). From the above-mentioned considerations, we formulated capabilities that should be available in the prospective geoportal prototype. The intended capabilities would include:

- **request map function**, to accommodate users (such DLR, as explained in the 2nd scenario) who need help from others on certain area to be mapped and users (as in the case of "Asterix") who are confused on the priority areas that need mapping.
- **status map function**, to accommodate users (such as the administrators, DLR, "Asterix," UN-Minustah, and "Harry,"as mentioned in our scenario) who want to know the progress of the work.
- **reporting damage function**, to accommodate users (as in the case of "Asterix") who want to report damages.
- **mapping function**, to facilitate users (as in the case of "Harry") want to create a base map.

These four functionalities are interrelated in perfecting each other's task to generate rapid post-disaster damage map (see Figure 4.1a).

This prospective geoportal prototype shall possess two significant advantages, i.e. request map and status map. The two advantages will prevent users from getting confused during the process of delivering assistance, and have positive impact, i.e. efficiency of the work, where directions, as the guidance, will inform users to deliver their share of contribution. These functionalities are not available in the existing geoportals, which implement crowdsourcing systems. We try to establish an architechture based on these capabilities because they are important in creating rapid post-disaster damage mapping.

Initially, the idea to create the intended architecture will be carried out through the existing platform, such as Ushahidi, which will save time, instead of building a new one. The four functionalities are planned to be embedded together into Ushahidi. Unfortunately, it is difficult to fathom the workflow in Ushahidi. However, the good news is that one of them, viz., "reporting damage" function can still perform in Ushahidi. And actually, this function is already available in Ushahidi. The remaining three functionalities will be accommodated with different suitable platforms, but it will also bring consequences, meaning if we integrate different systems into one, problems, such as accessibility to retrieve data from each platform, reading products from other platforms, will likely to occur. As a result, it will be difficult for us to manipulate the systems to serve our purposes.

Therefore, we would rather settle with the second option, instead of employing the first option that can only perform a single function.

Below are several platforms or applications that we use to help establishing the architecture. They will have the capacity to generate the prototype.

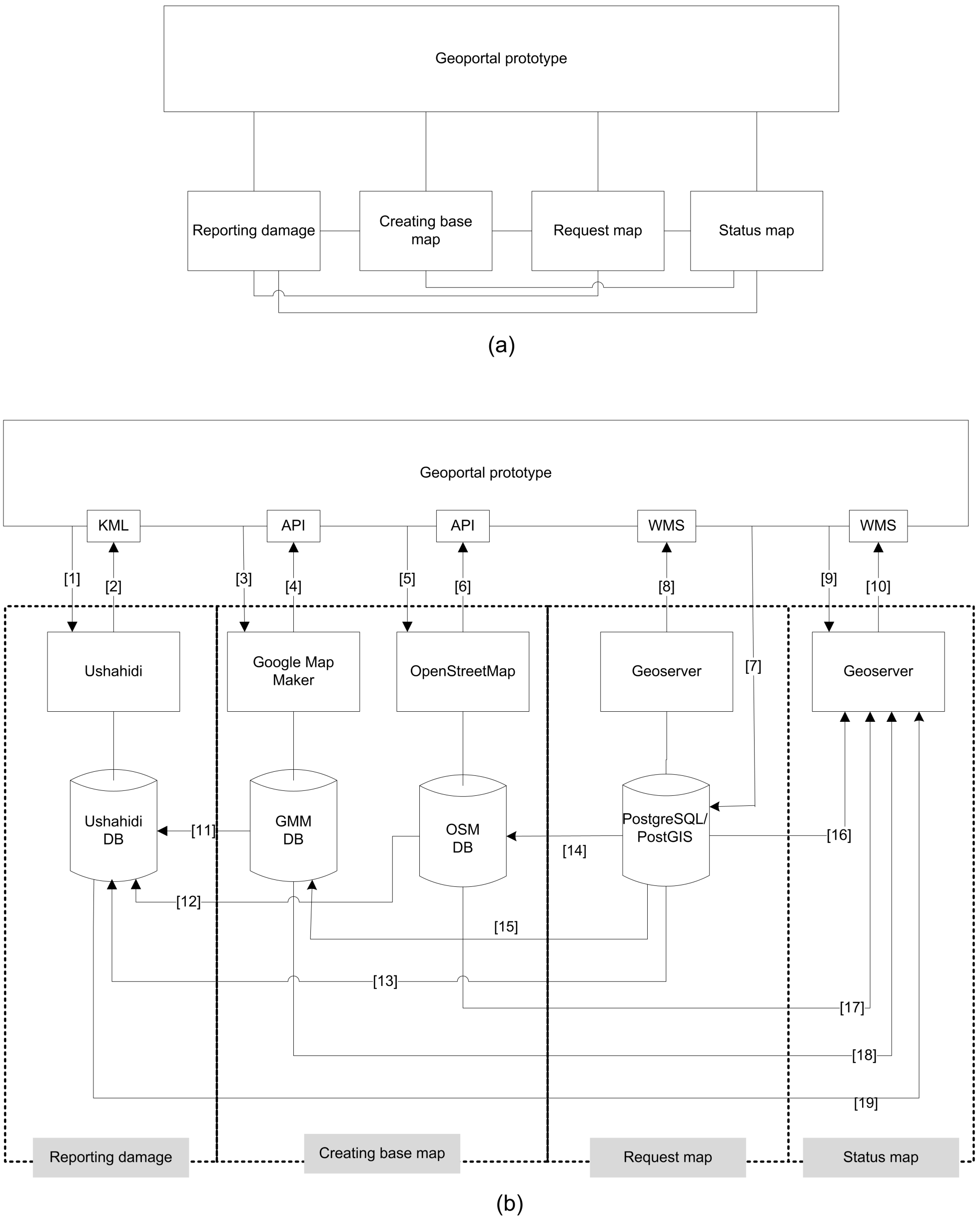


Figure 4.1: (a)The architecture of the prototype, (b) The detailed architecture of the prototype

Ushahidi

Ushahidi (www.ushahidi. com) is an open source crisis map platform. It provides services, such as information collection, visualization, and interactive mapping. This software was created in the beginning of 2008 to map reports of violence in Kenya. Since then, it has experienced collaborative development conducted by volunteers, under the direction of a core team (the founding members of Ushahidi). The Ushahidi's coding complies with that of Kohana standards, and they place the source code openly on the github site (https://github.com/ushahidi) for volunteers to develop this software (Ushahidi, 2011). Kohana is an open-source web application framework based on PHP 5. PHP 5 is a safe , simple, and user-friendly program to be developed by community in general (Kohana, 2011).

For reporting damage, we will use Ushahidi to collect information from users. This can be done by registered users (members) or non-registered users (public). Registered users are divided into two, namely administrators and members. The level of trustworthiness of their report is verified through a certain mechanism, viz., through comments and ranking by other users, and verification from administrators. Therefore, the information is sorted out based on the level of trust. The members and public have a full access to submit information to the server by using a form to create a report. This form is identical to that of the administrator's. The difference lies in the fact that administrators use a form equipped with evaluation form. Through this evaluation, administrators gain an access to verify the information. If the administrators approved the information, then the report can be published into an updated map. To access full application and menu, users must register themselves.

To enhance the quality of users' reports, administrators can add other layers into Ushahidi. For example, if there is an organization that needs assistance, they will create a location map where other users are able to participate. This request will be embedded onto a map layer. For the sake of further discussion, we will refer to this map as request map. Administrators will then put this layer as a background to lead users to certain spots on the map where they can update it. Such contribution should agree with the request.

The following are reasons why we choose Ushahidi as a tool for reporting damage:

- it provides a verified mechanism to measure the credibility of the reported data.
- it is equipped with add layer tool, which is useful to attach a layer onto the base map.
- it has implemented the crowdsourcing system.

Google Map Maker (GMM)

Google Map Maker is an online map editor that allows users to update geographic information to be seen from Google Maps and Google Earth. Google Map Maker takes the advantages of crowdsourcing to collect data. Through the involvement and contribution from all over the world, it is expected that local knowledge with comprehensive satellite imageries - reflecting the world - can be generated. The update will be reviewed and approved. Following the approval, administrators will display it online for all to see.

Google Map Maker has been used for crisis response (http://www.google.org/crisisresponse/) that makes the critical information about natural disaster and humanitarian crises more accessible. Some of the examples are 2011 Turkey earthquake, 2011 Thailand Flood, 2011 Japan Tsunami. Contributors can also customize the Google Map Maker by drawing shapes, adding text, and integrating live feeds of information, such as earthquakes while they are taking place. Google Map Maker has the advantage in visualizing an area before and after the crisis. The information that is input by contributors is not immediately available for public, because other users have to validate it in order to ensure the information can be represented accurately on the map. Unfortunately, Google restricts real-time access to their database, which is considered as one of the problems to manipulate the data. We are going to further discuss this aspect later on (in "Status Map" and "Semantic Web" sections).

For this research, we choose Google Map Maker as our map editor because it has various products, such as satellite imageries, topographies, and base maps. These products are advantageous for users on geoportal to use it as base map layers. In addition, this site [GMM] are familiar to countless registered users who have accounts in it, and are used as access to use Google map editor.

OpenStreetMap (OSM)

OpenStreetMap (http://www.openstreetmap.org/) is a scheme aimed to build a geographic database of the world, where every single geographic feature, such as streets, buildings, waterways, and woodlands can be recorded. In creating geospatial data from around the world, OSM uses data from imagery, maps, GPS, and other sources. Users can input and edit the features. The editing process of every object is stored as its history to trace errors. The interesting thing about OSM is that it will directly show the result of entries made by users. OSM also serves as another option instead of using GMM. Its mechanism is part of crowdsourcing system, known as gratification. The above-mentioned advantage of OSM are the reason that it was chosen as a map editor.

Openlayers

On the aspect of overlaying all layers, we use OpenLayers as the appropriate tool. This software is a client-side library based on JavaScript (http://www.openlayers.org). It is an open source software that can be used to make interactive web mapping.

It gives us the ability to build an entire mapping application. We can work on/with the application from the bottom up and customize every aspect of map-layer, such as control and event. This software is developed by Metacarta and it has grown remarkably into a framework that helps the community.

This software has capabilities, which supports this thesis, i.e., the ability to connect with different map servers, equipped with editing tools to create map, and commands in capturing coordinates from users' delineation.

We need software, such Open Layers, to mash up all layers or data from different tools or components of the geoportal to be a map interface for users.

Twitter

It is one of popular medium, because millions of people are using it (according to statements from various media, e.g. www.techland.time.com). It is a real-time information network with the capacity to relate to all stories, opinions, and news (https://twitter.com/about). Through this tool, we can find all stories and follow conversations on certain topics. For example, geoportal administrators can use Twitter to introduce the link to people. The nature of the introduction is of an open call, to participate in rapid post-disaster damage mapping. By connecting Twitter with geoportal, users can share their information concerning the disaster in a compact form. Since Twitter can only send 140 characters, it is suitable for the implementation of rapid tasking. These are the reasons why we opt for Twitter.

Geoserver

GeoServer (www.geoserver.org), an Internet map server, provides services (i.e, WFS, WMS, WCS) for users to distribute and update geoinformation. The Open Planning Project (TOPP) developed it in Java since 2001. It is free and open source software. It complies with OGC standards, which enable data sharing. Publishing data into different formats facilitates users' flexibility to display data through Google Earth, OpenLayers, and many other spatial data viewers. Data sources from external databases are accommodated. For users who prefer graphic user interfaces (GUIs) and wizards, GeoServer is the best option among other internet map servers.

The aforementioned paragraph leads us to select GeoServer as our map server in building the proposed geoportal. The Administrative interface of GeoServer is more convenient than configuration script. This is the main reason why we use it in this research.

PostGIS/PostgreSQL

PostGIS is a library that enhances the capability of PostgreSQL to manage spatial data (Obe and Hsu, 2011). Refractions Research began a project to develop an open source spatial database technology called PostGIS. As an open source, varied contributors have developed this software by adding new functions to it.

Combination PostGIS and PostgreSQL offers an option for users to choose it instead of other spatial database softwares. PostGIS has a lightweight engine than other spatial engines. A number of projects support the development of PostGIS, such as projection support (Proj4), Engine Open Source (GEOS) project, and Open Source Geospatial Foundation (OSGeo). For PostgreSQL, people say that it has competitive capabilities as other commercial softwares have, such as data storage up to terabytes, capability to manage index operators by creating new types and operators.

The most important aspect in this research is that PostGIS fulfills the OGC/ISO standards. This compliance will help us manage various formats of data, as long as they comply with the standards.

4.2 PROPOSED SYSTEM ARCHITECTURE

From the aforementioned, we are going to elaborate a detailed architecture for the geoportal prototype that can be seen in Figure 4.1b. Below are some important aspects that are going to be discussed respectively:

Reporting Damage

Line 1 is an access for users to Ushahidi. This is a component of the geoportal prototype to report damage. Ushahidi provides basic services for registered or unregistered users, such as reporting, commenting, and downloading the reports. For registered users, they can customize the platform, depending on their roles (i.e. member, admin, and super admin). For example, unregistered users cannot upload request map, represented by Line 13, to Ushahidi. Base maps in Ushahidi come from Google Map API [Line 11] and OSM API [Line 12]. The reports are in the Keyhole Markup Language (KML) format, to be shared and displayed to the interface [represented in Line 2]. It is more applicable to share this format with other geoportal components.

Creating a base map

We provide two online map editors for users to create a base map. They are Google Map Maker (GMM) and OpenStreetMap (OSM). People commonly use them to map certain areas on Earth and use their APIs to be the base maps of their applications. In this research, we find that these map editors have implemented crowdsourcing concepts in collecting data from contributors. Each of them applies different methods in approving the contributed data. In general, we do not want to compare these map editors. We only intend to accommodate any data sources produced by crowdsourcing map editors and use them as choices for users who need them as base maps. However, we notice that both have different and similar aspects. For example, one of the differences lies in publishing data, where GMM publishes data after commenting on other users, in advance. This will take time to process, especially if the area is unknown to other users. On the other hand, OpenStreetMap directly reflects people's contributions following their submission, and then other users will verify them. If we consider producing a base map in a rapid manner, OpenStreetMap will be the first choice. On the contrary, if we consider the aspect of reliability, we will choose GMM data as our base map. The similar things are these online map editors both have a mechanism of verifying crowdsourced data and share data in API.

In Figure 4.1b, there is a box to create a base map. In this box, we have two map editors that will function as components of the geoportal prototype. From the geoportal interface, users will find two choices of map editors to create base maps. Typically, users choose a map editor based on familiarity. For example, in Line 3, Users who are familiar with GMM, can go to this map editors. First, they upload a request map (coming from Line 15) into GMM, as a guidance for delineating map. In Line 5, we facilitate users who are familiar with OSM. They can contribute through it after uploading the request map layer from Line 14. Their contributions are sent to the geoportal interface through Line 4 (in Google Maps API) and Line 6 (in OSM API), and other components, such as reporting damage tool (Lines 11 and 12).

Creating a request map

Users are able to employ the request map tool through Line 7 and invite people to participate in mapping or reporting damages. This tool is developed from several software, namely OpenLayers [to capture geometries and the demanded attributes], PostGIS/PostgreSQL [to store data], and GeoServer [as the map server]. The request map is published in Web Map Service (WMS) and Keyhole Markup Language (KML) formats with the following reasons

- For WMS, users require a facility that can show the requested area with information in it. We assume this service can meet the requirement. This service is provided for the geoportal user interface, represented in Line 8.
- For KML, each geoportal component has a tool to read KML formats, such as add layer tool in Ushahidi, OSM, and GMM. The KML-request map goes to each component as a background layer, through Line 13 [for Ushahidi], and Lines 14 and 15 [for OSM and GMM].

Status Map

We proposed this function to give users information about the undergoing current works. We retrieve data from Ushahidi (Line 19), GMM (Line 18), and OSM (Line 17) databases through Geoserver.

Each data, entered by visitors, will be recorded. This is how users are informed on the progress status of the work; it displays the results of work that has been done or underway. An example of this is if an organization seeks help pertaining an area (request map) of the damage buildings. Some users may already know and report the locations of damaged buildings, including the status of certain area(s). All the while, there will also be some other users who wish to update the previous reports, without having to start the work from beginning. It is advisable that geoportals have this particular functionality.

4.3 GEOPORTAL WORKFLOW

For details, the workflow in Figure 4.2 is created to illustrate how this geoportal prototype runs. We classify contributors to create rapid post-disaster damage map into three groups: crowd-sourcing, official institutions, and satellite-based damage mapping communities.

The first group consists of individuals or institutions who want to report incidents (Line 1), or to create a base map (Lines 2 and 3), such as landmarks, roads, or built-up areas. The base map (Line 5) will be the base map layers on Ushahidi, connected through application programming interface (API) of Google maps and OpenStreetMap (Line 4). This group can see the status of the latest work from the Status Map (Line 24).

The second group, official institutions, comprising of organizations considered as governments and non-governmental organizations (NGOs). Based on their mission, these organizations can request help from people to collaborate in mapping certain area(s) (Line 6). This request will be converted into KML format by a map server (GeoServer) (Line 7) and it is added as a base layer on Ushahidi (Line 17). Another scenario: official institutions upload their data (Line 8) into Ushahidi in the KML/KMZ formats, or the web map service (WMS) format. They might even georeference a damage map from satellite-based damage mapping communities (Line 14], and convert it (lines 10, 11, and 12) into the KML/KMZ or WMS formats in order to use them as a background, which is useful for users while they are reporting damaged buildings. in Ushahidi.

The last group is satellite-based damage mapping communities. This group consists of communities who produce rapid post-disaster damage maps based on satellite imageries (Line 13). They need users to perform ground observations in identifying location, analysis, and obtaining feedback. By requesting a location (Line 15), the system will convert it into WMS and KML formats (Line 16] as a base layer for other users to report damage buildings, based on specific damage classifications (see figure 2.2 on page 11).

The reports from Ushahidi can be downloaded in KML format (Line 18) and use it as the reported damage buildings layer in Geoportal (Line 19) through OpenLayers, or it will be overlaid on other portals, depending on users' purposes. For status of reported damage maps, the request map is overlaid on OpenLayers, together with the reported damaged building layer. All these layers will be overlaid in OpenLayers that eventually produce the post-disaster damage map (Line 22). The post-disaster damage map is employed again by users who wish to participate in data contributing for this geoportal (Line 23).

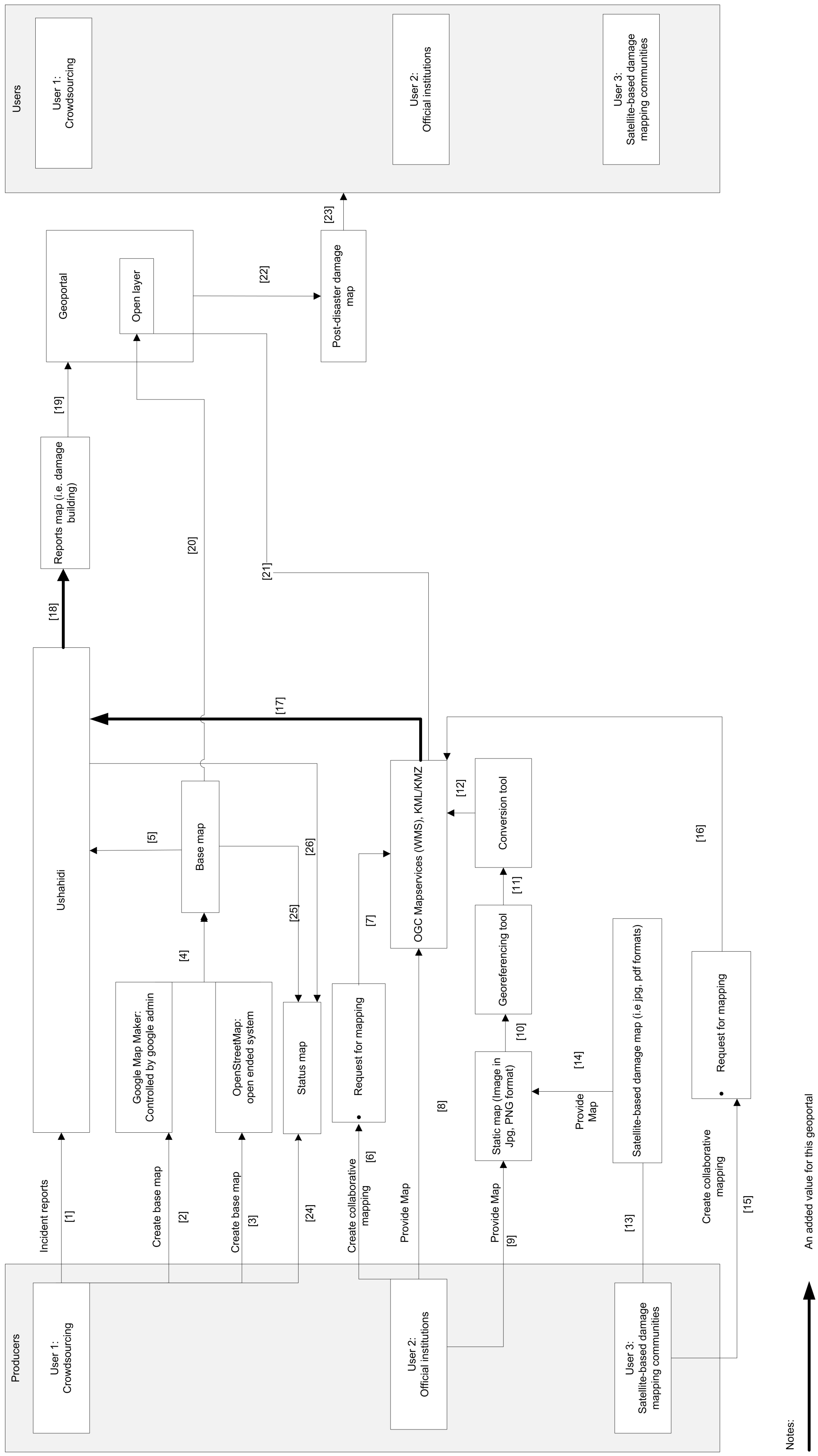


Figure 4.2: A workflow of crowdsourcing geoportal for rapid post-disaster damage mapping

4.4 CONCLUDING REMARKS

The geoportal architecture is designed to develop the prototype. Supporting the proposed geoportal will require certain software, namely Ushahidi (to gather incident reports), Google Map Maker and Open Street Map (to create base maps), OpenLayers (to mash up the maps), and Twitter (to introduce links to the geoportal for the public).

Chapter 5
Implementation and testing

5.1 MATERIALS

To implement the designed architecture, we prepared preliminary data and software. For data, we have

- Haiti administration boundary in shape file from UN-Minustah.
- Roads in shape file from UN-Minustah.
- Earthquake intensity from the PREVIEW Global Risk data platform.
- Google map API from Google.
- OpenStreetMap API from OpenStreetMap.

and for the software we use

- Xampp software package, which consists of Apache as a web server, PHP as a server-side scripting language, and MySQL as database.
- Ushahidi application for reporting damage.
- PostgreSQL/PostGIS for spatial database.
- GeoServer as the Internet map server.
- Adobe Dreamweaver 5 as the website editor.
- OpenLayers for the interactive map.
- Hyper Text Markup Language (HTML) and JavaScript to support the construction of geoportal website.

5.2 USER INTERFACE IMPLEMENTATION

Actually, a geoportal is a website, where people can find geographic information from different sources. To do so, we start by constructing a website. The first step to build the website is preparing a list of components that the geoportal should provide. The main components are as follows

- Home. It provides information for people on how to contribute to this geoportal. It has links to the request map function, Ushahidi for reporting damage, Google Map Maker and Open-StreetMap for creating a base map, including Twitter widget to call people to participate in the geoportal.
- Tools. It is a collection of functionalities to support users in converting data and georeferencing a map.
- Post-disaster damage map. It facilitates people to see and overlay maps from other users' contribution.

When users open the website, we need to cope with challenges in crowdsourcing systems, such as how to recruit contributors, how the system provides instructions, how to combine their contributions, and how to manage abuse (Doan et al., 2011). Nevertheless, user interface will not be able to overcome all these challenges, because other parts of the geoportal will do such tasks. What the user interface can solve is how to recruit contributors and what visitors may do.

The website for this geoportal derives from the website template, as part of practical session in module 10 (Web technology for GIS and mapping) in Geoinformatics-ITC. Again, we do not initiate the design from the beginning because of limited time, but we try to cope with the challenges in crowdsourcing systems by providing compact and clear information.

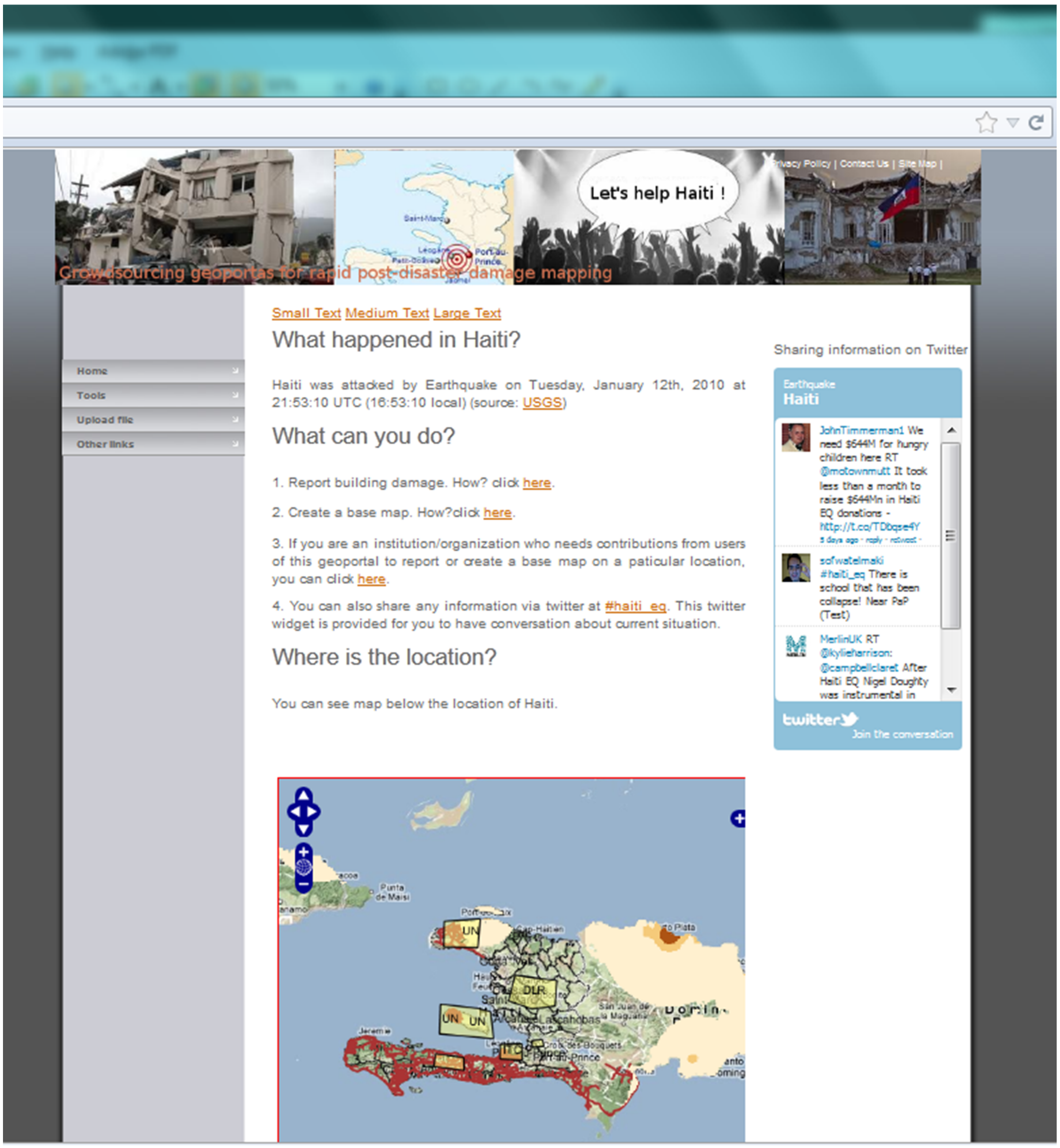


Figure 5.1: The main page of geoportal.

5.3 REQUEST MAP

The 'Request map' function facilitates organizations (users) to ask other users in assisting them to map their requested areas (in polygons). To build the function, we use OpenLayers to capture the area that users delineate, PostgreSQL/PostGIS to store the area as a polygon, and GeoServer as the Internet map server to publish it as map services to users.

From OpenLayers document (www.openlayers.org), we found two methods in sending and requesting the data back to the OpenLayers (see figure 5.2). In the first method, users data are recorded into database (PostgreSQL/Post GIS), and Polygon data are stored in WKT format. Then, the stored data are retrieved through Geoserver in WMS format, shown on the OpenLayers. While for the second method, we propose WFS transaction. Users' data are processed in WFS format. For this research, we implement the first method because we found proxy problem. This problem is discovered because Geoserver is using Jetty as a web server, of which port differs from another webserver where geoportal and Ushahidi were installed. We realize this problem after we set up the application. Right now, we have yet not tested which one is better. However, if we compare the number of lines coding that should be written, the second method is simpler than the first one because, in our point of view, it will generate different speed of data processing.

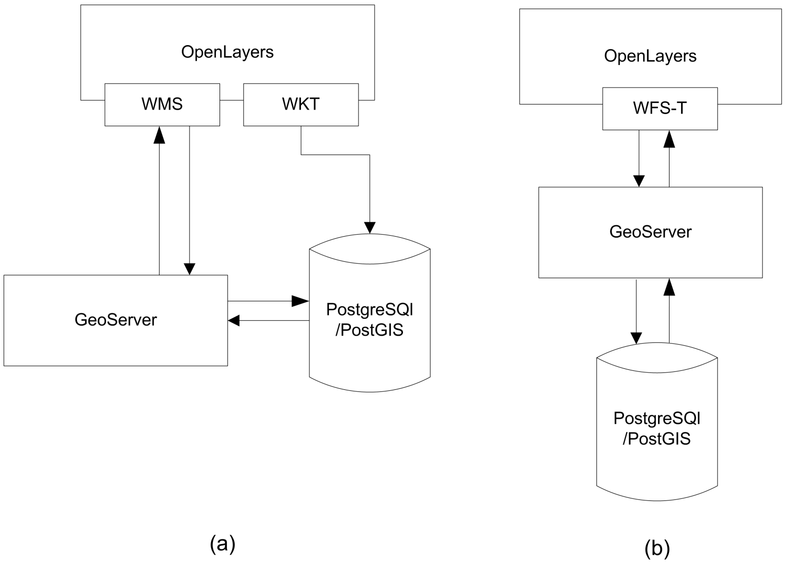


Figure 5.2: (a) Request map method 1, (b) Request map method 2

The workflow of requesting is users delineate area(s) they want to request and fill the request form (name, note, e-mail address), while to capture the coordinates, we use the following command (for details see in appendix A.1)

```
var geom = evt.geometry;
var x =geom.toString();
```

We define variable **geom** to recognize the event that users digitized and record it with variable x into the Well-known text (WKT) format. We use the WKT format because it is supported by OpenLayers and it can be stored into PostgreSQL/PostGIS. In the database, we create a table to store the data. The coordinate system that we use is spherical Mercator (EPSG:900913) in compliance with Google map and OpenStreetMap projection. By doing so, OpenLayers will read the coordinates in this coordinate system. When users push the submit button, all these data are stored into the created table with the following PHP command (for details see in appendix A.2):

```
$query = "INSERT INTO extent(ex_id, the_geom, users, note, email)
VALUES('" . $eqids ."',ST_GeomFromText('$wkt',900913), '" . $users . "',
 '" . $note . "', '" . $email . "')";
```

The submit button will activate PHP command to send data into the 'extent' table and insert the polygon, user name, note and email-address. Users who request for help can inform contributors what damage classifications should be followed and a link to a website that will assist them. These data will be shown as popup for contributors when they click the request map. For example, there is a request from World Bank for reporting damaged buildings in a certain area; they demand people to use EMS 98 for damage classifications and provide a link for more details. The machine records this request about 7 pm, on February 6, 2012. This information is meant to avoid users misleading in selecting a damage scale. (see Figure 5.3)

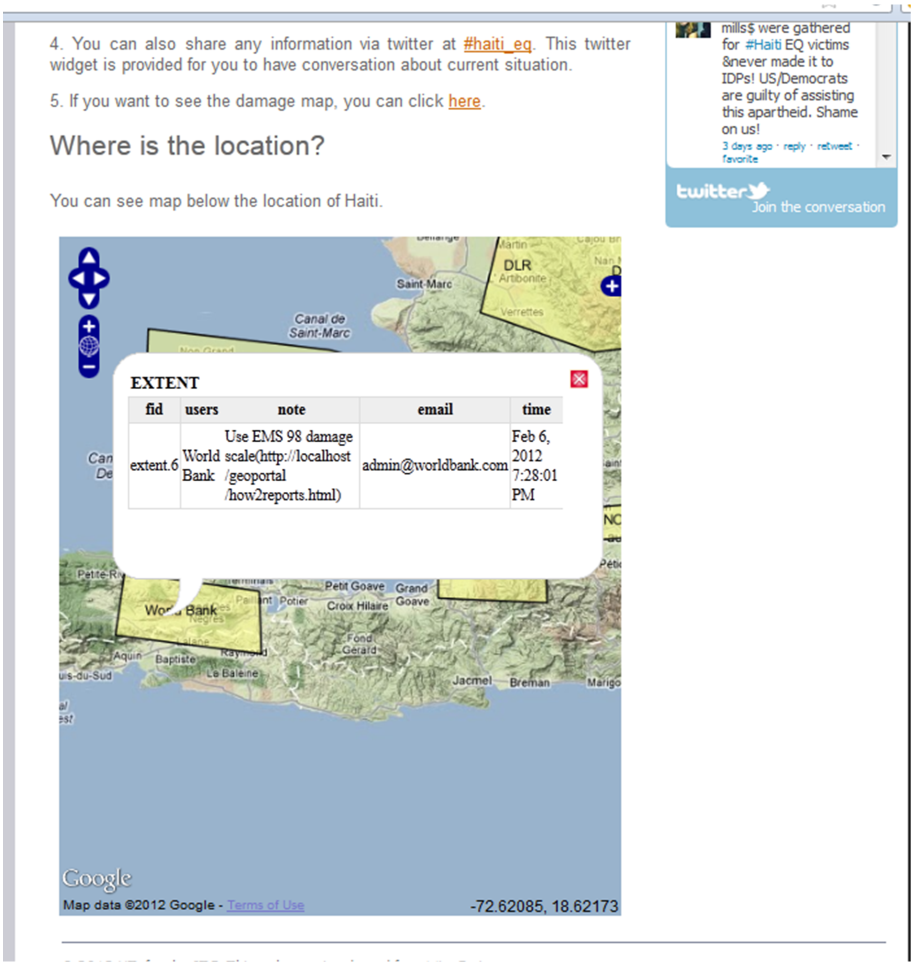
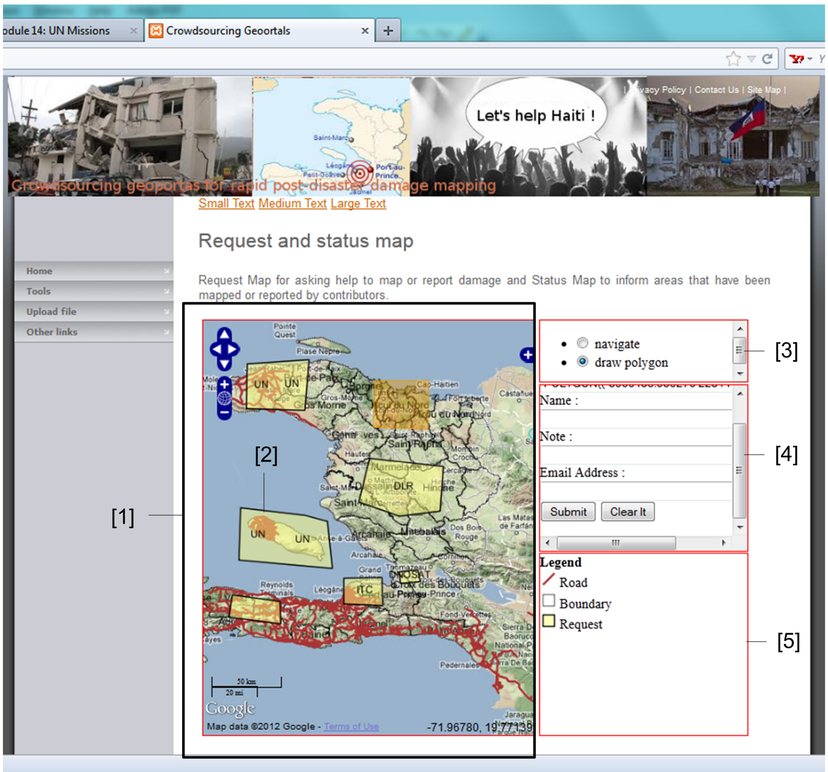


Figure 5.3: Popup info facility to inform users about the request.

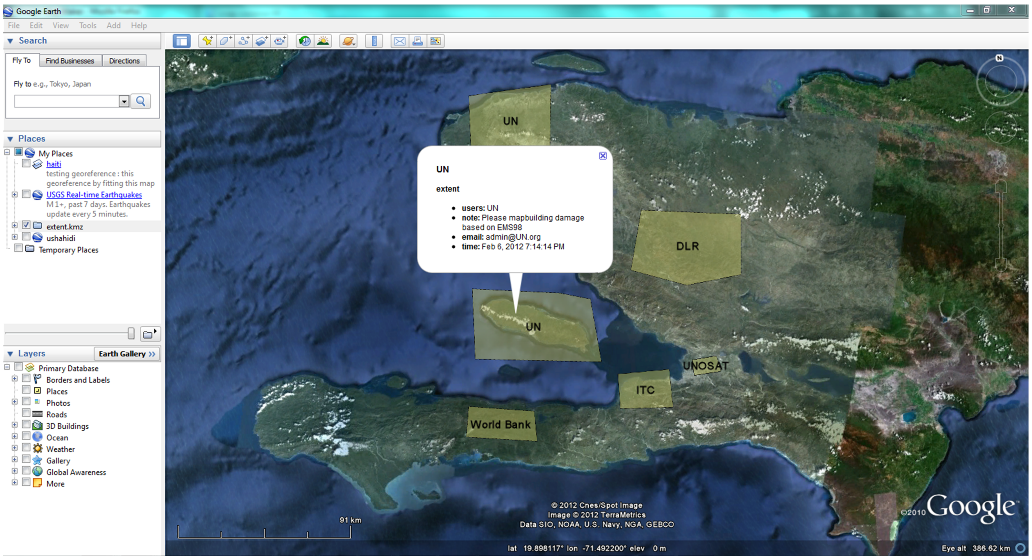
Figure 5.4 (a) shows the interface of request map, which consists of the map viewer [1], an example of the requested area [2], editing tools to draw the request area [3], the form where users can fill out information about their request [4], and the map legend [5]. We also provide KML/KMZ files for users to be downloaded. Figure 5.4 (b) is the request map shown on Google Earth. This request map is connected in real time to the database (PostgreSQL/PostGIS).

5.4 STATUS MAP

We do not implement this function because problems are encountered in manipulating codes for Ushahidi and it was also because of the fact that we are using different platforms. Consequently, each platform has their respective database,but we cannot access some of them because a number



(a)



(b)

Figure 5.4: (a) Request map, (b) Request map seen from Google Earth (in the KML format).

of platforms have their own restrictions, while the working principle of the status map is to report the latest update from users' entries. These entries are recorded in different databases, which depend on the platforms in use. We need raw data to generate the status map, but we cannot do so if we use an application programming interface (API).

For example, if users report damage, we accommodate them through Ushahidi that has its own database. While for creating a base map, users are provided with functionalities in GoogleMapMaker or OpenStreetMap, which are also equipped with different databases. This is one of the obstacles that makes it difficult to retrieve the data into our system as a status map.

5.5 SEMANTIC WEB

The idea to implement this concept is to facilitate users in searching data and to harmonize the data contributions. Examples are synchronization of different damage classifications and location names. For different damage classifications, calling people to contribute data about damaged buildings will bring a problem in to the database. Each of them will use different classifications based on their knowledge. Currently, there is no such agreement to use a certain standard of damage categories within international disaster management communities. For location names, a location could have more than one name because of different languages or local knowledge to name it. Unfortunately, we have not implemented Semantic Web in geoportals. In this case, difficulties lie on (1) different databases (2) accessibility to those databases. Although we cannot use Semantic Web, we can still create rapid post-disaster damage map.

5.6 TWITTER

This is one of social media used as the medium to invite people in undertaking some tasks in geoportal. To implement this idea, as an example, we choose Twitter. The consideration is that Twitter limits 140 characters that can be shared with other users in a compact form. This is beneficial when sharing information in emergency situation, where users will not spend their time reading long information. To connect Twitter with geoportal, we can use a widget, provided by Twitter, for people who want to attach it on their websites. This widget can show all information containing specific, popular topic(s) that users want to see on their websites. For this geoportal, we put the topic with hashtag haiti_eq (#haiti_eq)inside the Twitter widget command, as shown below

```
search: '#haiti_eq'
```

(for details see appendix A.3). We can see the Twitter widget in figure 5.1.

5.7 REPORTING DAMAGE

To implement reporting damage, described figure 4.1b, we use Ushahidi in our geoportal prototype. For installation, Ushahidi needs three software, installed in advance. They are Apache (as a webserver), PHP (as the server-side scripting language) and MySQL (as the database of Ushahidi).

Damage classification used is EMS 1998 (see more details in Section 2.3). We can group damage classification into two classes, i.e. from ground observation and satellite imagery interpretation (see figure 5.5). In addition, we provide two inquires that users can answer: (1) Can you still enter the damaged building? (2) And how many stories are there in the building? These questions are additional, intended to obtain additional information for relief actions. They are adopted from the question *did you feel it*? in USGS earthquake portal (http://earthquake.usgs.gov/earthquakes/dyfi/).

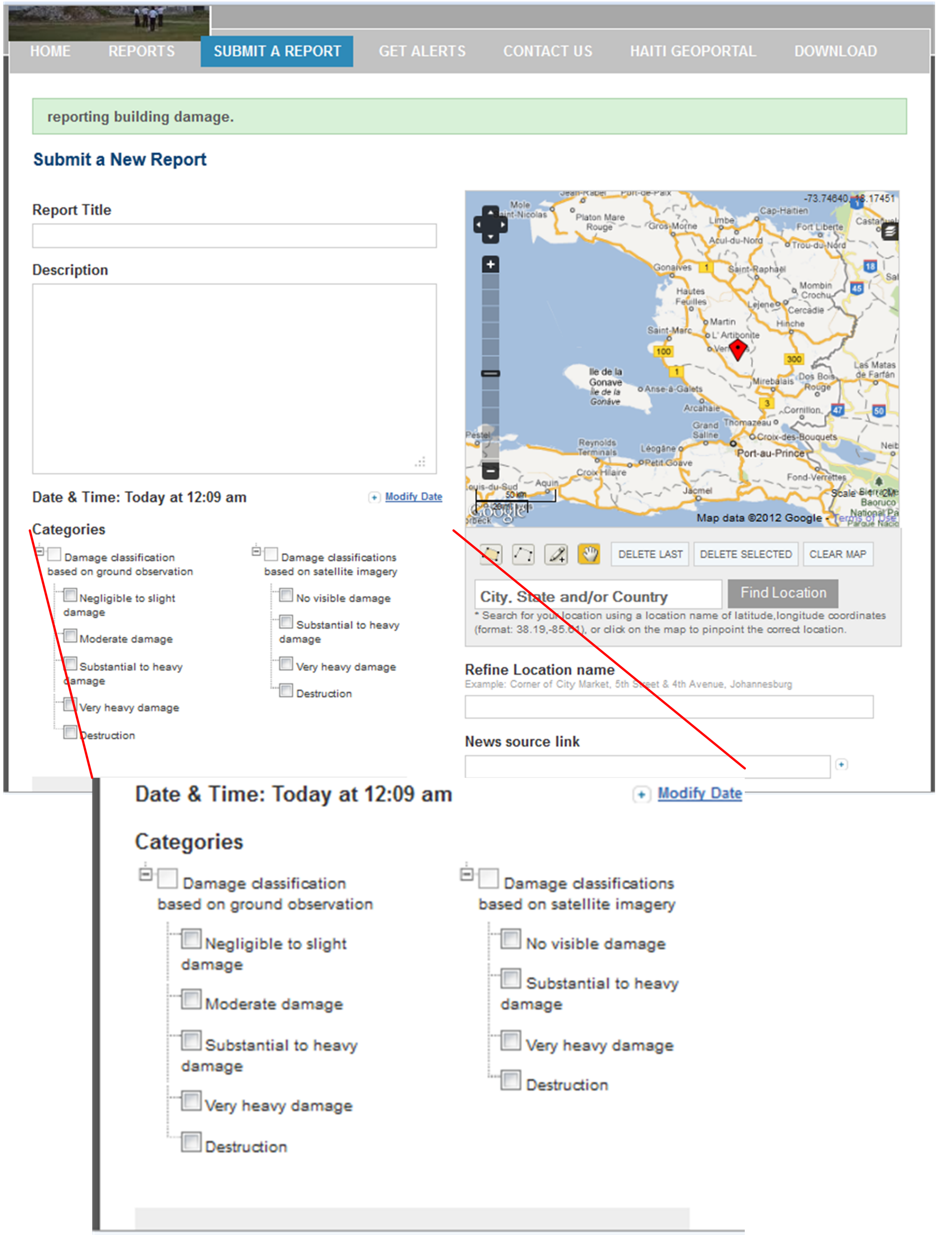
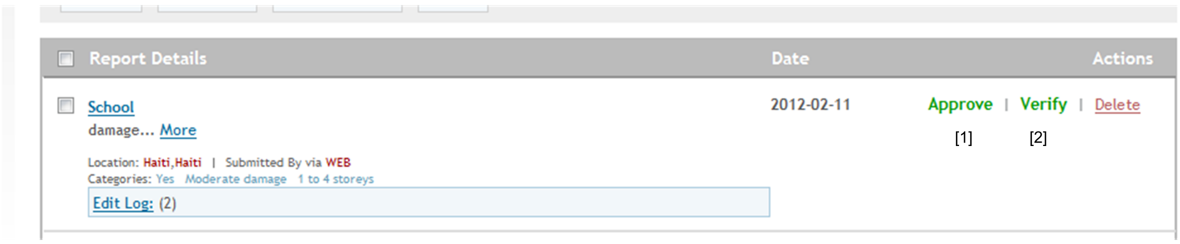
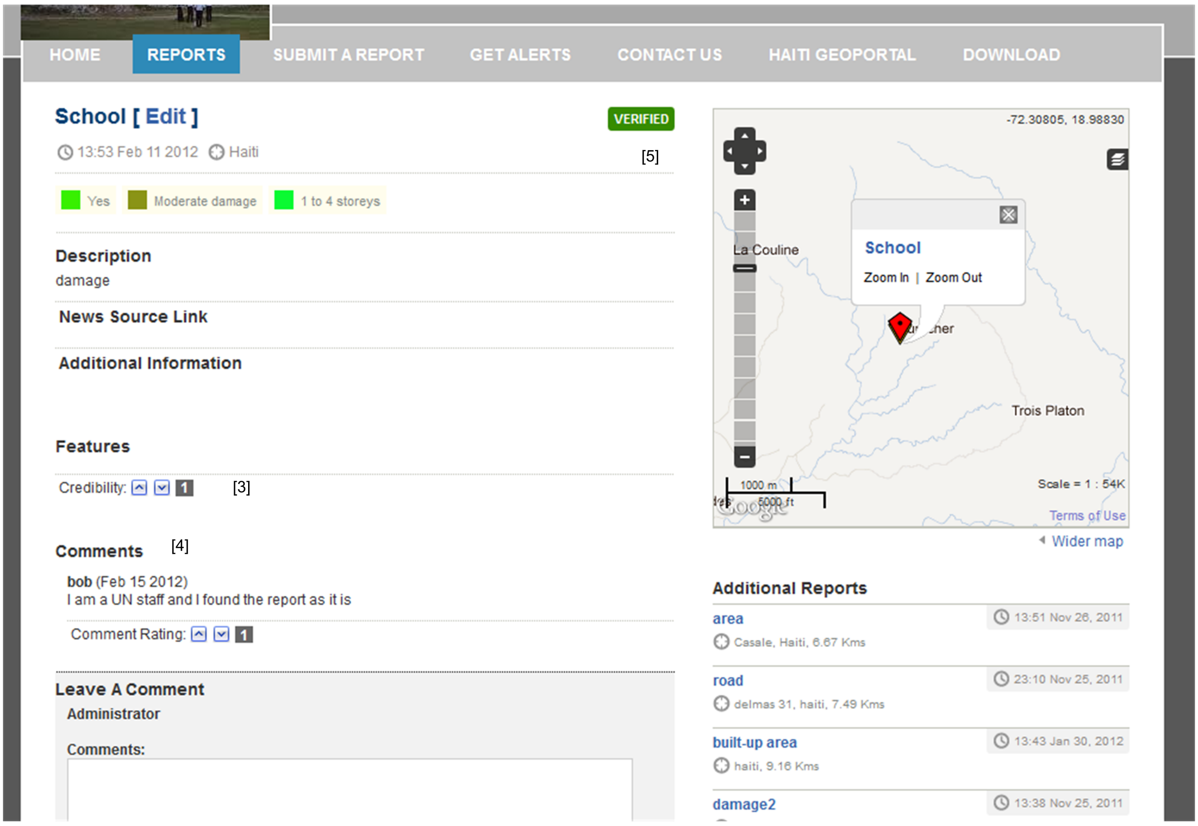


Figure 5.5: Reporting damage on Ushahidi

Reports from users will be publicly shown if these reports have been approved by administrators. For instance, how reliable are the data, the administrators will consider it based on comments, ratings (credibility) of the reports by other users. Administrators have three options with respect to any report: approve, verify, and delete. If the administrators approve and verify the reports, it means these reports are trustworthy. However, if the administrators only approve, the reports will be shown but not verified. Figure 5.5 shows an example a damage report of a school building is approved [1], and verified [2] by the administrator, after considering the rating [3], and comments [4] on this report. It certifies that the report has been verified [5] for other users.



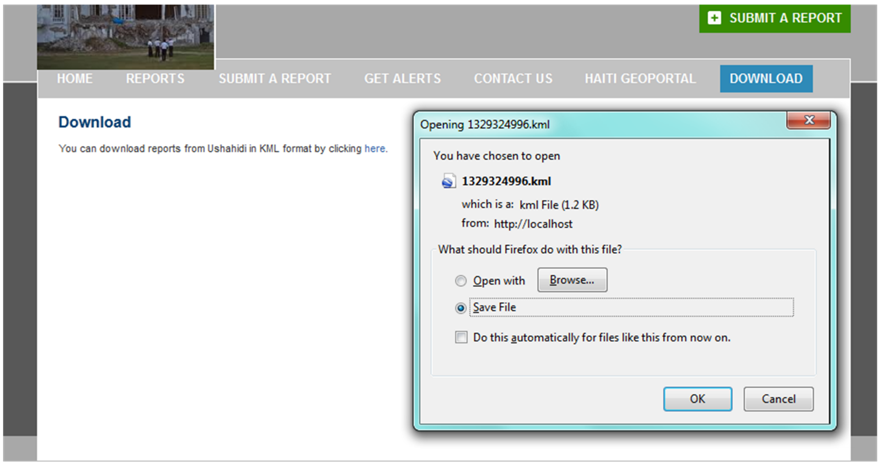
(a)



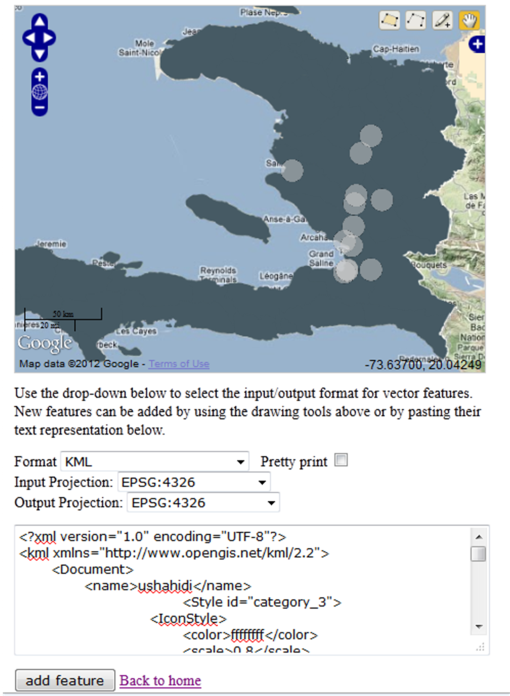
(b)

Figure 5.6: (a) The reports approval by the administrator, (b) The reports shown by users.

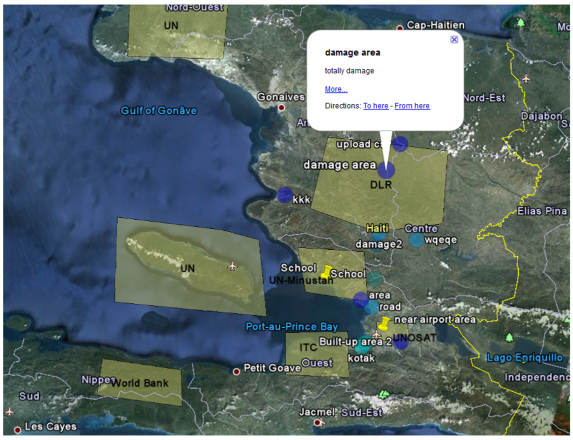
We also provide a link for users to download all reports in the Ushahidi database (figure 5.7a). The format is in KML, because it will give users a chance to overlay it on OpenLayers to create a damage map on geoportal (figure 5.7b) or on another system, such as Google Earth (figure 5.7c). To do so, we install a KML report plug-in, created by David Kobia, into Ushahidi.



(a)



(b)

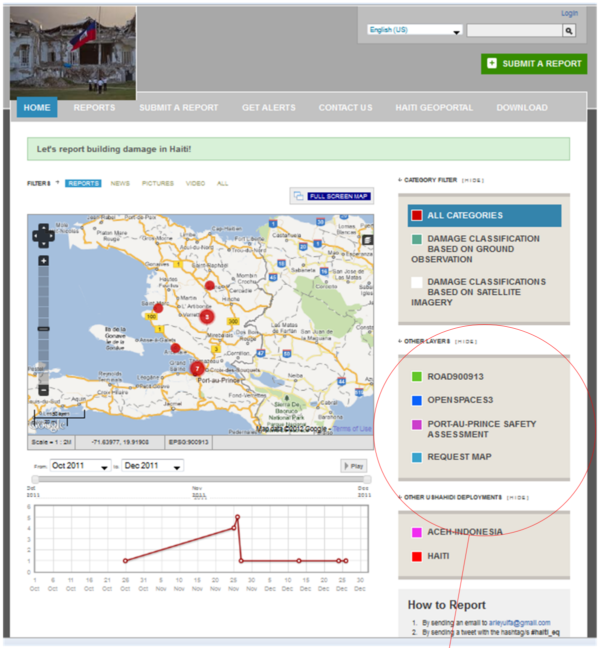


(c)

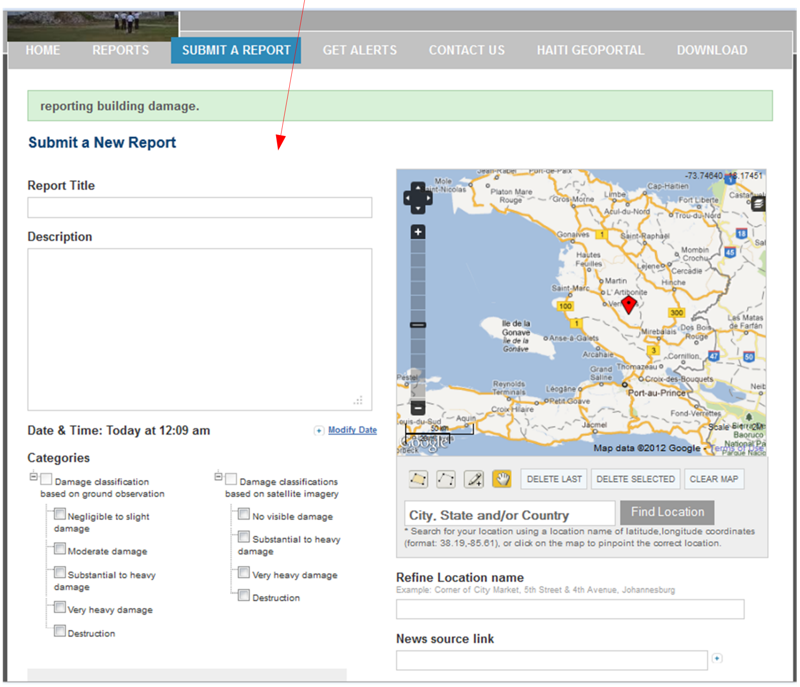
Figure 5.7: (a) Download reports (b) Upload reports to create rapid damage map, (c) Overlay the reports on Google Earth.

Ushahidi has a function to add a layer. This function can be accessed only by the administrator. It only reads data in KML/KMZ format. Unfortunately, this function cannot open KML/KMZ files containing images and links to other database. Due to this limitation, there are some workflows in Figure 4.2 that cannot be implemented. But, the idea to overlay satellite imageries and base maps can still be done using Google Map, Google Satellite, and OpenStreetMap, as basemap layers, on where users can post their reports.

Users can activate KML/KMZ layers, added by the administrator, on the front page of Ushahidi. According to the workflows of the prototype, it will be useful if we can shift it to the submit-report page (figure 5.8a and b), where users can use this as a base layer while reporting the damage buildings.



(a)



(b)

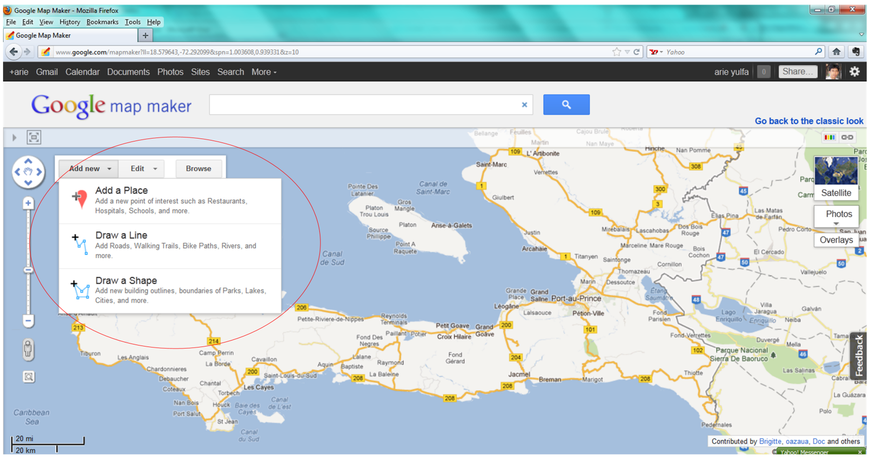
Figure 5.8: (a) Front page of Ushahidi (b) Submit report page of Ushahidi.

5.8 CREATING A BASE MAP

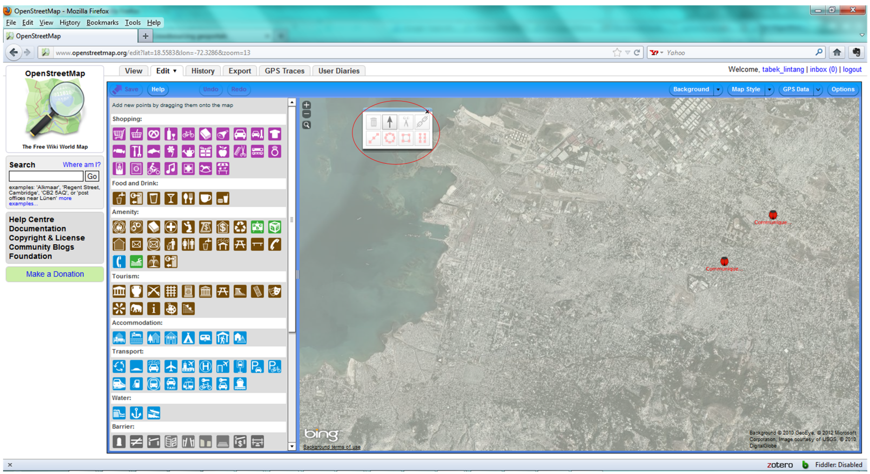
Users are enabled to create a base map utilizing Google Map Maker or OpenStreetMap as their tool. To execute this action, we provide how to do that on geoportal and links to these map editors. The request map link is in KML format. It is used to lead people to locations that need digitizing.

Using Google Map Maker

With Google Map Maker (figure 5.9a), we direct people through a link. To use the Google Map Maker, users should own a Google account. Then, users overlay the KML-request map where they can map the prioritized spot(s). To do so, users click the overlay menu and give a link where the KML-request map is stored (see figure 5.10a). But, this function does not work because KML-request map contain a link to other database (called dynamic file). It works when the KML layer is static (without a link).



(a)



(b)

Figure 5.9: (a) Google map maker editor, (b) OpenStreetMap editor.

Google provides satellite imagery, map, and terrain that are important while digitizing a base map. Before shown publicly, there is a mechanism in creating base map that users should observe.

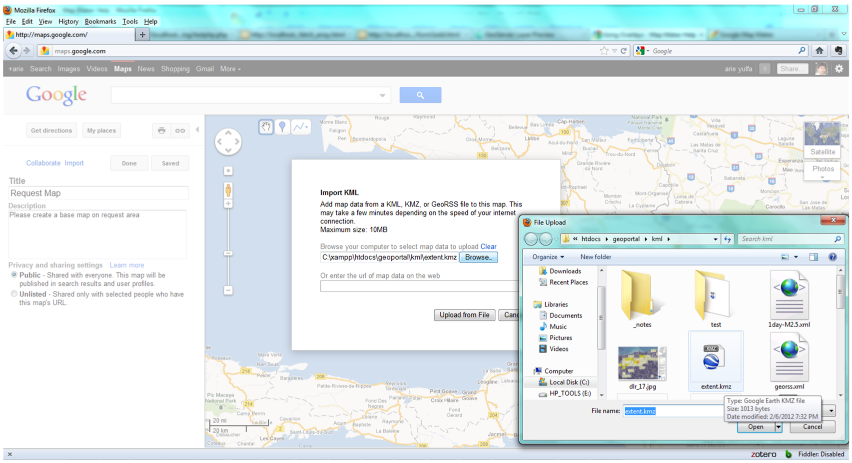
First of all, users should be active reviewing others' editing, and second his editing needs more review from other users. These requirements are good for the aspect trustworthiness but, on the other hand, it is not advised to delay the result of work done by users because it will contradict one of the challenges of crowdsourcing system (i.e. "how to retain users.") Users can be retained if they receive some kind of gratification, such as to witness the result of their contribution immediately.

Using OpenStreetMap

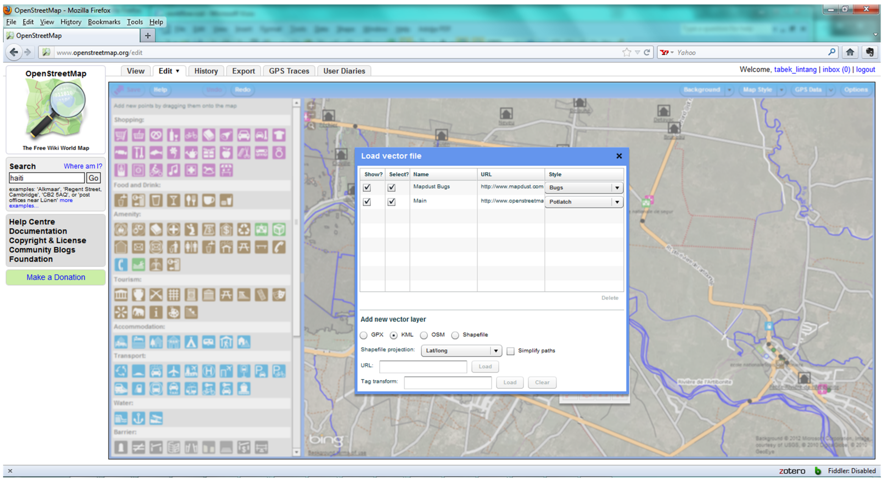
Another option in creating a base map is OpenStreetMap (figure 5.9b). Similar to Google Map Maker, we only need to give a link to OpenStreetMap editor and provide information for users on how to contribute to OpenStreetMap in geoportal. The mechanism of trustworthiness in OpenStreetMap is similar to that of a wiki-like system. All the changes in editing process will be recorded as history. So, errors can be returned based on the recorded history. The result of users editing will be shown instantly. It solves the challenge on "how to retain users," because it gives a kind of gratification by displaying what has been mapped. To lead users to spot(s) that needs mapping users can overlay the request map onto OpenStreetMap (figure 5.10b). However, similar to Google Map Maker, this overlay function cannot read the request map because it contains a link to other database (PostgreSQL/PostGIS).

5.9 DAMAGE MAP WITH OPENLAYERS

To display the result of crowdsourced-data, we use OpenLayers. We provide uploading and exporting tools in several formats, such as GeoJSON, Atom, KML, GML, WKT, and GPX. These formats represent static layers, where the coordinates are not stored and connected to other database. This type of damage map only recognizes two map projections, namely geographical (EPSG:4326), and spherical Mercator (EPSG:900913). Unfortunately, according to cartographic standards, the result of this damage map is still poor and we do not support it with printing function, where users can print it in PDF or other formats. Actually, it is possible to do such task (printing) by adding some command lines to realize it, but we have limited time to fix it.



(a)



(b)

Figure 5.10: (a) Overlaying KML in Google Map Maker editor, (b) Overlaying KML in OpenStreetMap editor.

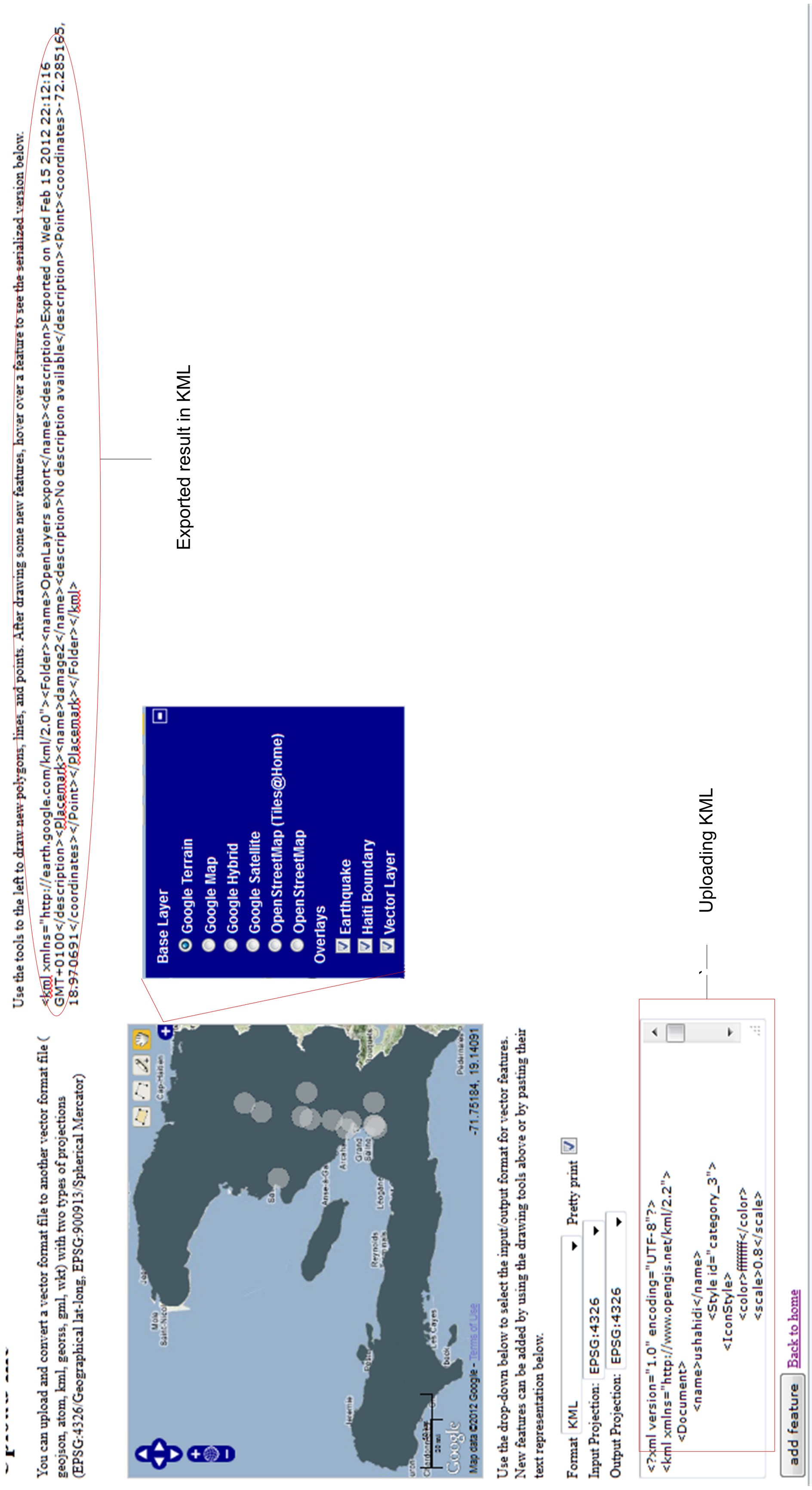


Figure 5.11: Viewer of post-disaster damage map

5.10 EVALUATION

5.10.1 Reporting damage

Functionalities to report damage is running on the installed Ushahidi of which purpose is to accommo-date people who want to contribute. However, not all of the designed architecture can be implemented through Ushahidi because some problems exist:

1. Ushahidi cannot add KML/KMZ layers, which contain links to external database (dynamic). In this research, we encounter that the request map layer (in KML/KMZ format) cannot be shown when we add it with add layer tool.
2. Ushahidi cannot display the KML/KMZ file containing satellite imagery, as a base layer, while users are reporting the damaged building.
3. Ushahidi provides option categories in form of checklists, where users may fill more than one options of the Damage categories. This way will mislead the damage building classes in the reports.

From our viewpoint, we wish to propose the following to solve the problems:

1. For the first and second problems, we can edit the scripts inside Ushahidi because they are open-source, and Ushahidi is built by volunteers who are led by the core team from Ushahidi. All the source codes are placed at https:\\github.com\ushahidi\Ushahidi_Web. We tried to solve these problems but halted the effort because it was time-consuming to improve them. As a beginner in Ushahidi software, we need to learn Kohana web framework in which Ushahidi platform complies with. Based on the trial we conducted, we can at least start again to solve it by manipulating the code in manage.php under this root directory "..\application\controllers\admin\".
2. For the third problem, there are two solutions: before users fill the list of categories, provide them with clear direction on the geoportal such as popup info facility and replace the checklist options into radio button option.

We choose the KML/KMZ format because (1) Ushahidi has the add layer tool for KML/KMZ, which will shorten the time to manipulate the tool, instead of (compared to) building a new one in Ushahidi (2) This format is also used in Google Map Maker and OpenStreetMap editors.

5.10.2 Create a base map

We provide two options for users to create a base map, i.e. Google Map Maker or OpenStreetMap editor. Problem will arise if we use the KML/KMZ format, which is of a dynamic type. Similar to Ushahidi, these map editors cannot display the file. We use KML/KMZ file to overlay request map to give directions for users where their help is needed. In contrast, Google earth can perfectly read this file. In solving this problem, we recommend both map editors to improve their programs, especially Google Map Maker, where they can find solution from Google Earth capability in reading KML/KMZ files (dynamic type).

5.10.3 Rapidness

From the literature review, we do not find any common standards that are used to measure rapidness of the damage mapping. We only find prediction of time that will be consumed, such as a few hours, one to two days, even a month. It depends on the extent of the damaged area.

What we should do to evaluate the rapidness producing the damage map is to invite potential users of the geoportal and apply the scenarios regarding to user requirements. Unfortunately, instead of inviting potential people to play the scenarios, we play them by ourselves. The result shows that the geoportal cannot fully meet the designed workflow because we still have to deal with technical problems. The impact also suggests that users have to perform it manually, instead of having it successfully done in automatic mode (i.e. downloading manually the request map and uploading it into Ushahidi, GMM or OSM). This impact will take time longer than automatic mode.

5.11 CONCLUDING REMARKS

In this chapter, we try to implement and evaluate the designed architecture of the geoportal prototype. After an evaluation, we find that some workflows cannot be implemented. However, we can still apply the basic idea of geoportal for producing the rapid post-disaster damage map. In this case, there are some solutions that we can use as a starting point to implement the designed prototype.

Chapter 6

Results and Discussion

The result of this crowdsourcing geoportal promises efficiency between users who need and wish to help. An example of this efficiency can be seen from request map where this function will guide contributors in sharing information to fulfill the request, such as reporting damaged buildings to the appropriate damage classification where it belongs. If the status map can be implemented, it will also improve the efficiency of work because it will inform users which of the requested areas that do not have data yet. Four functionalities (request map, status map, reporting damage and creating a base map) provided on the geoportal are leading users to produce rapid post-disaster damage map. These functionalities aim to accommodate users in contributing data to complete the task. They run on different platforms, such as Ushahidi, GMM or OSM. In short, geoportal plays a role to integrate products of these platforms.

Since these platforms have applied a crowdsourcing system, products integrating of these platforms will generate a number of advantages, namely:

1. It saves time, instead of building new ones,
2. It provides good data quality because the data have passed a crowdsourcing mechanism as verification,
3. It can generate data in interchangeable formats, such as KML/KMZ and API, which complies with OGC standards, thus enabling the integration.

On the other hand, this integration can also breed problems. Each platform, such as GMM and OSM has their own systems that are impossible for manipulation. This situation disable some functionalities to run (i,e., status map and semantic web).

Rapidness of the mapping process is possible in this geoportal, it can be seen from the simulations of the created scenarios have been ran on the geoportal. If the rapidness is tested to the people who are really defined in the scenarios, it will give more reliable the speed of the damage mapping is able to be done.

In the designed architecture, we plan to use a certain format file. After we implement it, we find that the format file which connects to external database will not run as what we have designed. Even though, this problem can be solved by recreating the file manually or converted it into static type but it will consume longer time in the working process.

According to Doan et al. (2011), in employing crowdsourcing to accomplish certain tasks, we need to clarify such tasks, provide the proper tools, guideline, and medium. In the context of this research, we have clarified the main task in creating rapid post-disaster damage map. To produce the rapid post-disaster damage map, we also have divided the task into smaller portions, namely reporting damage, creating base maps, and evaluating the two assignments mentioned earlier. Tools needed for reporting damage are already in place, produced by Ushahidi. In creating base maps, we provide two options, i.e., Google Map Maker and OpenStreetMap. While for the aspect of evaluation, Ushahidi, Google Map Maker, and OpenStreetMap have made the necessary mechanism available, such as "ranking," and "commenting," the contributed data. Our geoportal functions as the medium for integrating data, and serves to present guidelines, i.e., "request map," and "pop-up information." Our results have agreed with statements uttered by Doan et al. (2011) in accomplishing the tasks. However, in the implementation, not all the functionalities are working as the designed because of the integration of the systems into the crowdsourcing geoportal. To solve it, we need to do manually of processing instead of automatically as what we have designed. For a short term, it is solved but for a longterm if the systems or platforms can fix the problems such as reading KML/KMZ file format it will strengthen crowdsourcing geoportal capabilities.

Chapter 7
Conclusions and recommendations

7.1 CONCLUSIONS

From the results, we would like to say that the crowdsourcing geoportal that we designed is able to accommodate users' requirements in creating rapid post-disaster damage mapping. Nevertheless, there are some limitations that we need to improve such as status map, semantic web and some workflow that do not run. In general, the result of this research will help us to answer the following research questions.

1. What type of users can benefit the geoportal for rapid post-disaster damage mapping? According to the study, there are two types of users who obtain the advantages from this activity. They are,
   (a) Users who are familiar with relief actions. They need as much information as possible pertaining to the situation in the field. This kind of information is very important for them in designing relief activities. In addition, several other organization or groups whose expertise is closely related to rapid post-damage mapping. These people will have their needs fulfilled through geoportals, where they can meet other users who have similar interest to deliver help.
   (b) Public who are willing to supply information of the damaged-area(s). These people have the urge to help or participate. Geoportals will assist them on helping the above-mentioned users.
2. What problems will arise on integrating map sources? Our findings suggest that:
   - Different map projections will create difficulty or cannot be integrated with maps on OpenLayers.
   - Another problem that may occur is in the aspect of directing users to observe certain damage classifications because the result may vary or unreliable due to the poorly informed users in making their choices.
   - Even though the maps are the same data format with criteria of add layer tools in the geoportal components, i.e, KML/KMZ files, we find these tools are not able to read maps containing dynamic layer (see section 5.10).
   - Utilizing the different platforms (i.e., Ushahidi, GMM, and OSM) will bring us to face different policies to access directly their databases. In term of our case, we cannot create functionalities (status map and semantic web) because of restriction to have real-time access to their databases.
3. How to solve these problems so that integrating process can be done? We provide the solutions below in order, following the sequence in which the issues discussed above.
   - Determine what map projection to be used. For example, in OpenLayers we type a command line to re-project a certain map projection onto another. (see in appendix A.1 under *function init()*)

- Improve the Ushahidi, GMM and OSM capability to read dynamic KMZ/KML layers.
- Provide clear instruction for users and prevent users from choosing more than one options in categories (e.g. radio button instead of checklist) in Ushahidi.
- Suspend some functions that cannot be applied because of restriction in accessing directly databases of different platforms.

4. What is the principal concept of crowdsourcing for rapid post-disaster damage mapping? In this case, the first step is to invite people to participate in creating a damage map, following the earthquake. Then, provide the participants with appropriate tools, clear instruction, and a mechanism to verify data.
5. How to combine crowd-sourced information based on different attribute classification?

   - We determine the classification to be used and create rules to maintain the precision of users' entries. Then, combine the maps on OpenLayers. For example, we follow EMS 1998 as the damage scale.
   - Because we utilize crowdsourcing system that mostly the contributors are on the ground, we distinguish damage classification from ground observation and satellite imageries as seen in Figure 5.5.' Ground observation' has five classes of damage and 'satellite imageries' only use four because the second class is difficult to determine through imagery. If the difference is known, it will be easy for us to combine the different attribute categorization.
   - We also provide a function to inform contributors what the appropriate classification is demanded by requestors (as seen in Figure 5.3). This function, popup info, is intended to prevent misleading in choosing a classification.

6. What is the design framework for the geoportal for post-disaster damage mapping? We describe the designing of geoportal in Chapter 4. The objective of this geoportal is to create a rapid post-disaster damage map from different users through the implementation of crowdsourcing concepts. To do so, the geoportal should have tools for reporting damage, map editor, request map, status map, and database to store information. These information are (semi-)automatically integrated to be the intended map and provided for users.
7. What is the concept software stack for the prototype of the geoportal client/server architecture? In term of creating a rapid post-disaster damage map, we need a set of programs that works together to accomplish it. From the users' requirements, we find that some functional tools are needed to complete the task. For instance, creating a request map, showing the status of current works, reporting damage, and creating base maps. Each of functionalities has their own system but to organize these functionalities in completing the task, request map is used as guidance. To make sharing data and integrating data possible, there are some standardizations should be fulfilled (i.e., OGC standards).
8. How to implement proof of concept software stack for the geoportal client/server architecture? To prove that our concept software stack can accomplish creating a rapid damage map, we should identify software that can fulfilled users' needs and interoperability standards. For creating a request map, we utilize OpenLayers, GeoServer and PostGIS/PostgreSQL to develop this tool. For status map, we cannot implement it as the designed because of technical problem. For reporting damage, we use Ushahidi. For creating base maps, we employ two online map editors, Google Map Maker and OpenStreetMap. Subsequently, for evaluation, we run the geoportal prototype based on the created scenarios and the users' requirements.

7.2 RECOMMENDATIONS

We are aware that limitations do exist in this research and that they cannot fully answer the research questions. However, for future works, we would like to recommend the following points

1. For request map, we recommend the application of WFS-T, because it has complete functionalities for editing that will enable us to compare its product with the request map implemented in this research.
2. For Ushahidi, there should be an improvement by adding WMS and KML layers, which contain dynamic link to other servers. It can start with

   ```
   public function layers()
   ```

   in manage.php under this root directory "..\application\controllers\admin\". Additional layers, used as background layers, should also be provided in the 'submit report' menu to enable users in placing their reports with base layers. We cannot manage this issue because of limited time to explore the codes, but file reports_submit.php in the root directory "..\themes\default\view\" and reports_submit_edit_js.php in the root directory "..\application\views\\" is sufficient as a starting point for improvement.
3. For Google Map Maker, the KML/KMZ file of request map cannot be shown following the over-laying on Google Map Maker because it contains links to other servers (Geoserver, Post-greSQL/PostGIS). In Google Earth, KML/KMZ file can be shown and connected in real time to Geoserver and PostgreSQL/PostGIS. In my opinion, this can also be implemented on Google Map Maker.
4. Similar to Google Map Maker, OpenStreetMap should improve their engine to understand the dynamic KML/KMZ format file.
5. For reporting damage, the improvement for Ushahidi should be emphasized on how it manages different categories that will enable users to select only once from category lists.
6. We have two functionalities that are not successful to be applied. They are status map and semantic web. The status map is one of the main functionalities that are designed and semantic web is the additional one planned for users to search data in semantic level. Both have a similar problem, difficulties to access in real-time raw data from different platforms. If it can be solved, it will give users an ability to create and search data of damage in the crowdsourcing geoportal easier.
7. For accessing database, we suggest to make a temporary database to store data from platforms's databases that cannot be access directly or propose to the developers of the platforms to be improved.

Bibliography

Aardt, J. A., McKeown, D., Faulring, J., Raqueno, N., Casterline, M., Renschler, C., Eguchi, R., Messinger, D., Krzaczek, R., Cavillia, S., John Antalovich, J., Philips, N., Bartlett, B., Salvaggio, C., Ontiveros, E., and Gill, S. (2011). Geospatial disaster response during the haiti earthquake: A case study spanning airborne deployment, data collection, transfer, processing, and dissemination. *Photogrammetric Engineering and Remote Sensing*, 77(9):923–931.

Arlow, J. and Neustadt, I. (2005). *UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design (2nd Edition).* Addison-Wesley.

Berners-Lee, T., Hendler, J., and Lassila, O. (2002). The semantic web. *Scientific American Special Online Issue*, pages 24–30.

Broek, v. d. B., Kiefl, R., Riedlinger, T., Scholte, K., Granica, K., Gutjahr, K., Stephenne, N., Binet, R., and de la Cruz, A. (2009). Rapid mapping and damage assessment. In Jasani, B., Pesaresi, M., Schneiderbauer, S., and Zeug, G., editors, *Remote Sensing from Space*, pages 261–286. Springer Netherlands.

Doan, A., Ramakrishnan, R., and Halevy, A. Y. (2011). Crowdsourcing systems on the world-wide web. *Communications of The ACM*, 54(4):86–96.

Duda, K. A. and Jones, B. K. (2011). Usgs remote sensing coordination for the 2010 haiti earthquake. *Photogrammetric Engineering and Remote Sensing*, 77(9):899–907.

Feeney, M.-E., Rajabifard, A., and Williamson, I. P. (2001). Spatial data infrastructure frameworks to support decision-making for sustainable development. In *Statement on the Cadastre. Report Prepared for the International Federation of Surveyors by Commission 7 (Cadastre and Land Management)*, pages 21–25.

Giuliani, G. and Peduzzi, P. (2011). The preview global risk data platform: a geoportal to serve and share global data on risk to natural hazards. *Natural Hazards and Earth System Sciences*, 11(1):53–66.

Goodchild, M. F. and Glennon, J. A. (2010). Crowdsourcing geographic information for disaster response: a research frontier. *International Journal of Digital Earth*, 3(3):231–241.

Google (2011). Support disaster relief in haiti. `http://www.google.com/relief/haitiearthquake/`. Accessed: 5 December, 2011.

Heinzelman, J. and Waters, C. (2010). Crowdsourcing crisis information in disaster affected haiti. Technical report, United States Institute of Peace, Washington, DC.

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

Howe, J. (2006). The rise of crowdsourcing. `http://www.wired.com/wired/archive/14.06/crowds.html?pg=1&topic=crowds&topic_set=`. Accessed: 5 December, 2011.

Imagecat (2010). Virtual disaster viewer: 2010 haiti earthquake. http://www.virtualdisasterviewer.com/vdv/index.php?selectedEventId=7. Accessed: 20 November, 2011.

Kerle, N. (2010). Satellite-based damage mapping following the 2006 indonesia earthquake-how accurate was it? *International Journal of Applied Earth Observation and Geoinformation*, 12(6):466 – 476.

Kerle, N. (2011). Remote sensing based post-disaster damage mapping – ready for a collaborative approach? *Earth Zine, Fostering Earth Observation & Global Awareness*.

Kohana (2011). Kohana development. http://dev.kohanaframework.org/. Accessed: 12 December, 2011.

Maguire, D. J. and Longley, P. A. (2005). The emergence of geoportals and their role in spatial data infrastructures. *Computers, Environment and Urban Systems*, 29(1):3–14.

Maiyo, L., Kerle, N., and Köbben, B. (2010). Collaborative post-disaster damage mapping via geo web services. In Konecny, M., Bandrova, T. L., Zlatanova, S., Cartwright, W., Gartner, G., Meng, L., and Peterson, M. P., editors, *Geographic Information and Cartography for Risk and Crisis Management*, Lecture Notes in Geoinformation and Cartography, pages 221–231. Springer Berlin Heidelberg.

Molina, M. and Bayarri, S. (2011). A multinational sdi-based system to facilitate disaster risk management in the andean community. *Computers & Geosciences*, 37(9):1501–1510.

Nebert, D. D. (2004). Developing spatial data infrastructures: The sdi cookbook. http://www.gsdi.org/docs2004/Cookbook/cookbookV2.0.pdf. Accessed: 18 August, 2011.

Nikolaos, A., Kostas, K., Michail, V., and Nikolaos, S. (2005). The emerge of semantic geoportals. In Meersman, R., Tari, Z., and Herrero, P., editors, *On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops*, volume 3762 of *Lecture Notes in Computer Science*, pages 1127–1136. Springer Berlin Heidelberg.

Nikolaos, A., Kostas, K., Michail, V., and Nikolaos, S. (2009). Towards a semantics-based approach in the development of geographic portals. *Computers & Geosciences*, 35(2):301–308.

Obe, R. O. and Hsu, L. S. (2011). *PostGIS in action*. Manning Publications Co., Stamford.

Open Geospatial Consortium Inc, . (2004). Geospatial portal reference architecture. http://portal.opengeospatial.org/files/?artifact_id=6669. Accessed: 2 November, 2011.

OpenStreetMap (2010). Haiti crisis map. http://haiti.openstreetmap.nl/. Accessed: 20 November, 2011.

OpenStreetMap (2011). Wikiproject haiti. http://wiki.openstreetmap.org/wiki/WikiProject_Haiti. Accessed: 5 December, 2011.

Putra, T. Y., Aditya, T., and de Vries, W. (2010). Making sense of local spatial data infrastructure in volcanic disaster risk management; a case study at sleman regency, indonesia. In *GSDI 12 World Conference*, Singapore. GSDI 12 World Conference.

Saito, K. and Spence, R. (2004). Rapid damage mapping to support post-disaster recovery. Number 2906, Vancouver, B.C., Canada. World Conference on Earthquake Engineering, 13th World Conference on Earthquake Engineering.

UNISDR (2009). Terminology of disaster risk reduction. http://www.unisdr.org/files/7817_UNISDRTerminologyEnglish.pdf. Accessed: 9 September, 2011.

UNOSAT (2010). Poster: Building damage assessment, haiti. united nations institute for training and research (unitar) operational satellite applications programme (unosat). http://maps.unosat.org/HT/EQ20100114HTI/PDNA_HTI_EQ2010_BuildingDamagePosterA0_v1_LR.pdf. Accessed: 10 November, 2011.

USGS (2011a). Earthquake summary: Magnitude 7.0 - haiti region. http://earthquake.usgs.gov/earthquakes/eqinthenews/2010/us2010rja6/. Accessed: 12 November, 2011.

USGS (2011b). Hazards data distribution system (hdds). http://hdds.usgs.gov/hdds2/. Accessed: 12 November, 2011.

Ushahidi (2010). Haiti: the 2010 earthquake in haiti. http://haiti.ushahidi.com/. Accessed: 5 December, 2011.

Ushahidi (2011). Coding standards. http://wiki.ushahidi.com/doku.php?id=coding_standards. Accessed: 20 December, 2011.

Voigt, S., Schneiderhan, T., Twele, A., Gähler, M., Stein, E., and Mehl, H. (2011). Rapid damage assessment and situation mapping: Learning from the 2010 haiti earthquake. *Photogrammetric Engineering and Remote Sensing*, 77(9):923–931.

Westen, C. J. V. (2011). Remote sensing and gis for natural hazards assessment and disaster risk management. Paper in module 12: Spatial data for disaster risk management at ITC-University of Twente.

Appendix A

Appendix

A.1 SCRIPTS OF REQUEST MAP INTERFACE

```
<html>
<head>
<!--load Openlayers JS API-->
<script src="http://openlayers.org/api/OpenLayers.js" type="text/
    javascript"></script>

<!--load Google v3 JS API-->
<link rel="stylesheet" href="http://trac.osgeo.org/openlayers/
    browser/trunk/OpenLayers/theme/default/google.css" type="text/
    css">
<script src="http://maps.google.com/maps/api/js?v=3.5&amp;sensor=
    false" type="text/javascript"></script>
<style type="text/css">
     div.olControlScaleLine {openlayers.org; color: black;
        bottom: 60px;     height: 10px;}
</style>

<script type="text/javascript">
 var drawControls;
 function toggleControl(element) {
     for(key in drawControls) {
         var control = drawControls[key];
             if(element.value == key && element.checked) {
                         control.activate();
                     } else {control.deactivate();}}}

 function allowPan(element) {
                var stop = !element.checked;
                 for(var key in drawControls) {
                     drawControls[key].handler.stopDown = stop;
                     drawControls[key].handler.stopUp = stop;}}

 function init(){
      var latlonProj = new OpenLayers.Projection("EPSG:4326");
      var googleProj = new OpenLayers.Projection("EPSG:900913");
 // world extent in google projection coords:
```

```
        var worldExtent = new OpenLayers.Bounds(-20037508,
            -20037508,20037508, 20037508);
 // center of Haiti
            var latlonCenter = new OpenLayers.LonLat(-72.5,18.9);
              var xyCenter = latlonCenter.clone().transform( //
                 reprojected into Google proj
                  latlonProj, //from
            googleProj //to
              );
        var map = new OpenLayers.Map("map",
                    {   projection: googleProj,
                        displayProjection: latlonProj,
                        units: "m",
                        maxResolution: 156543.0339,
                        maxExtent: worldExtent } );

        var gphy = new OpenLayers.Layer.Google(
          "Google␣Terrain",
          {type: google.maps.MapTypeId.TERRAIN, 'sphericalMercator':
              true, numZoomLevels: 22});
            var gmap = new OpenLayers.Layer.Google(
          "Google␣Map", // the default
          {'sphericalMercator': true, numZoomLevels: 22} );
            var ghyb = new OpenLayers.Layer.Google( "Google␣Hybrid",
          {type: google.maps.MapTypeId.HYBRID, 'sphericalMercator':
             true, numZoomLevels: 22});
            var gsat = new OpenLayers.Layer.Google( "Google␣
               Satellite",
          {type: google.maps.MapTypeId.SATELLITE, 'sphericalMercator
             ': true, numZoomLevels: 22});

// create OSM layers
            var mapnik = new OpenLayers.Layer.OSM();
        var osmarender = new OpenLayers.Layer.OSM("OpenStreetMap␣(
           Tiles@Home)",
                                                  "http://tah.openstreetmap.
                                                     org/Tiles/tile/${z}/${
                                                     x}/${y}.png" );

//creating a polygon
            var polygonLayer = new OpenLayers.Layer.Vector("Polygon␣
               Layer");
            var roadLayer = new OpenLayers.Layer.WMS ("Roads","http
               ://localhost:8181/geoserver/haiti/wms?",{layers:'
               haiti:haiti_road', transparent: "true", format: '
               image/png'});
        var boundLayer = new OpenLayers.Layer.WMS ("Haiti␣Boundary",
           "http://localhost:8181/geoserver/haiti/wms?",{layers:'
```

```
        haiti:haiti_bound', transparent: "true", format: 'image/
        png'});
    var requestLayer = new OpenLayers.Layer.WMS ("Request␣Map","
        http://localhost:8181/geoserver/haiti/wms?",{layers:'
        haiti:extent', transparent: "true", format: 'image/png
        '});

         map.addLayers([gphy, gmap, ghyb, gsat,mapnik, osmarender
            ,boundLayer,roadLayer, requestLayer , polygonLayer])
            ;
    map.setCenter(xyCenter, 8); //zoom in level  to Haiti
    map.addControl(new OpenLayers.Control.LayerSwitcher());
        map.addControl(new OpenLayers.Control.ScaleLine());
    map.addControl(new OpenLayers.Control.MousePosition());

         drawControls = { polygon: new OpenLayers.Control.
            DrawFeature(polygonLayer, OpenLayers.Handler.Polygon
            , {featureAdded: function(evt){

//capturing delineated coordinate in WKT
                    var geom = evt.geometry;
                    var x = geom.toString();

document.myform3.formvar1.value = x;}})};

        for(var key in drawControls) {
                    map.addControl(drawControls[key]);}}

</script>
</head>
<body onLoad="init();␣loadtime();">
<div id="map" style="position:absolute;␣width:400px;␣height:500px;
    ␣z-index:1;␣left:␣5px;␣top:␣6px;␣overflow:␣hidden;␣border:␣1px
    ␣solid␣red;"></div>
<div id="legend" style="position:absolute;␣width:250px;␣height:218
    px;␣z-index:2;␣left:␣411px;␣top:␣288px;␣overflow:␣auto;␣border
    :␣1px␣solid␣red;">

<!--map legend-->
<b>Legend</b><br/>
<img src="http://localhost:8181/geoserver/wms?&SERVICE=WMS&REQUEST
    =GetLegendGraphic&VERSION=1.1.1&FORMAT=image/png&LAYER=
    haiti_road"> Road <br />
<img src="http://localhost:8181/geoserver/wms?REQUEST=
    GetLegendGraphic&VERSION=1.1.1&FORMAT=image/png&LAYER=haiti:
    haiti_bound"> Boundary<br />
<img src="http://localhost:8181/geoserver/wms?REQUEST=
    GetLegendGraphic&VERSION=1.1.1&FORMAT=image/png&LAYER=haiti:
```

```
   extent"> Request<br />
 </div>

<!--activate creating a polygon-->
<div style="position:absolute;␣width:250px;␣height:74px;␣z-index
   :2;␣left:␣411px;␣top:␣6px;␣overflow:␣auto;␣border:␣1px␣solid␣
   red;">
  <ul id="controlToggle">
     <li>
       <input type="radio" name="type" value="none" id="
          noneToggle" onclick="toggleControl(this);" checked="
          checked" />
       <label for="noneToggle">navigate</label>
     </li>
     <li>
       <input type="radio" name="type" value="polygon" id="
          polygonToggle" onClick="toggleControl(this);" />
       <label for="polygonToggle">draw polygon</label>
     </li>
  </ul>
</div>

<!--A form for requesting help-->
<div style="position:absolute;␣width:250px;␣height:200px;␣z-index
   :2;␣left:␣411px;␣top:␣84px;␣overflow:␣auto;␣border:␣1px␣solid␣
   red;">
 <form name= "myform3" action="test_form3action.html" method="post
    ">
    WKT: <input type="text" size="40" length="40" name="formvar1"
       ><br>
    Name : <input type="text" name="users" size="40" length="40" >
       <br>
    Note : <input type="text" name="note" size="40" length="40" ><
       br>
        Email Address : <input type="text" name="email" size="40"
          length="40" ><br>
    <input type="submit" name="submit" value="Submit">
    <input type="reset" name="reset" value="Clear␣It">
 </form>
</div>
</body>
</html>
```

A.2 STORING REQUEST DATA TO POSTGIS/POSTGRESQL

```
<html>
    <body>
<!--Define variables-->
<?php
```

```
        $db = pg_connect('host=localhost dbname=opengeo user=
            postgres password=aq');
        $users = pg_escape_string($_POST['users']);
        $note = pg_escape_string($_POST['note']);
        $email = pg_escape_string($_POST['email']);
                $wkt = pg_escape_string($_POST['formvar1']);

//generate id
$eqid = pg_query($db,"select ex_id FROM extent " );
$eqid_last =MAX( pg_fetch_all_columns($eqid));
$eqids = $eqid_last + 1;

//inserting data to table 'extent' in PostGIS
$query = "INSERT INTO extent(ex_id, the_geom, users, note, email)
  VALUES('" . $eqids . "',ST_GeomFromText('$wkt',900913), '" .
  $users . "', '" . $note . "', '" . $email . "')";
$result = pg_query($query);
        if (!$result) {$errormessage = pg_last_error();
            echo "Error with query: " . $errormessage;
            exit();}
       printf ("%s, your request was accepted", $users);
       pg_close();
?>
<br> <a href="test_form3.html">Back to home</a>
    </body>
</html>
```

A.3 SCRIPTS FOR TWITTER WIDGET ON GEOPORTAL

```
<script src="http://widgets.twimg.com/j/2/widget.js"></script>
<script>
new TWTR.Widget({
  version: 2,
  type: 'search',
  search: '#haiti_eq',
  interval: 30000,
  title: 'Earthquake',
  subject: 'Haiti',
  width: 200,
  height: 300,
  theme: {
    shell: {
      background: '#8ec1da',
      color: '#ffffff'
    },
    tweets: {
      background: '#ffffff',
      color: '#444444',
      links: '#1985b5'
```

```
    }
  },
  features: {
    scrollbar: true,
    loop: true,
    live: true,
    behavior: 'default'
  }
}).render().start();
</script>
```