



**NAVAL
POSTGRADUATE
SCHOOL**

MONTEREY, CALIFORNIA

THESIS

**USE OF MOBILE VIDEO TO ENHANCE SITUATIONAL
AWARENESS IN HA/DR MISSIONS**

by

Nusret Denizci

September 2012

Thesis Advisor: Gurminder Singh
Thesis Co-Advisor: John Gibson

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

| REPORT DOCUMENTATION PAGE | | | Form Approved OMB No. 0704-0188 | |
|---|--|---|--|--|
| Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503. | | | | |
| 1. AGENCY USE ONLY (Leave blank) | | 2. REPORT DATE September 2012 | 3. REPORT TYPE AND DATES COVERED Master's Thesis | |
| 4. TITLE AND SUBTITLE Use of Mobile Video to Enhance Situational Awareness in HA/DR Missions | | | 5. FUNDING NUMBERS | |
| 6. AUTHOR(S) Nusret Denizci | | | 8. PERFORMING ORGANIZATION REPORT NUMBER | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000 | | | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER | |
| 9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A | | | 11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number: N/A. | |
| 12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited | | | 12b. DISTRIBUTION CODE | |
| 13. ABSTRACT (maximum 200 words) There are several incident report systems designed for humanitarian assistance/disaster relief operations. The Smartphone Assisted Readiness, Command, and Control System, SPARCCS, is one which brings the availability of cloud computing and mobility of smartphone/tablet together. It provides a clear common operational picture in order to create situational awareness among participants. Initially, responders are not able to share video clips between each other and headquarters. This thesis proposes to add video share, capture and display capability to the system. | | | | |
| 14. SUBJECT TERMS Android Programming, Cloud Computing, Common Operating Picture, Humanitarian Assistance, Disaster Relief, Situational Awareness | | | 15. NUMBER OF PAGES 59 | |
| | | | 16. PRICE CODE | |
| 17. SECURITY CLASSIFICATION OF REPORT Unclassified | 18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified | 19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified | 20. LIMITATION OF ABSTRACT UU | |

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**USE OF MOBILE VIDEO TO ENHANCE SITUATIONAL AWARENESS IN
HA/DR MISSIONS**

Nusret Denizci
Lieutenant Junior Grade, Turkish Navy
B.S., Turkish Naval Academy, 2006

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

**NAVAL POSTGRADUATE SCHOOL
September 2012**

Author: Nusret Denizci

Approved by: Gurminder Singh
Thesis Advisor

John Gibson
Thesis Co-Advisor

Peter Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

There are several incident report systems designed for humanitarian assistance/disaster relief operations. The Smartphone Assisted Readiness, Command, and Control System, SPARCCS, is one which brings the availability of cloud computing and mobility of smartphone/tablet together. It provides a clear common operational picture in order to create situational awareness among participants. Initially, responders are not able to share video clips between each other and headquarters. This thesis proposes to add video share, capture and display capability to the system.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

| | | |
|------|---|----|
| I. | INTRODUCTION | 1 |
| A. | MOTIVATION | 1 |
| B. | BACKGROUND | 3 |
| C. | PURPOSE | 4 |
| D. | ORGANIZATION | 5 |
| II. | EVOLUTION OF HUMANITARIAN ASSISTANCE/DISASTER RELIEF SYSTEM | 7 |
| A. | SIReS—SITUATION AND INCIDENT REPORT SYSTEM | 7 |
| B. | NICS—NEXT GENERATION INCIDENT COMMAND SYSTEM | 8 |
| C. | AGIS—ADVANCED GROUND INFORMATION SYSTEMS | 9 |
| D. | SPARCCS—SMARTPHONE ASSISTED READINESS, COMMAND, AND CONTROL SYSTEM | 10 |
| E. | SUMMARY | 12 |
| III. | INFRASTRUCTURE BACKGROUND | 13 |
| A. | CLOUD COMPUTING | 13 |
| B. | GOOGLE APP ENGINE AND GOOGLE WEB TOOLKIT | 15 |
| C. | ANDROID OPERATING SYSTEM | 16 |
| D. | WEB SERVICES | 17 |
| E. | BST PLAYER | 17 |
| F. | ANDROID VIDEO FEATURE | 18 |
| 1. | VideoView | 19 |
| 2. | Camera | 20 |
| G. | SUMMARY | 22 |
| IV. | EXTENSIONS OF SPARCCS | 23 |
| A. | SPARCCS WEB-CLIENT | 23 |
| 1. | Initial Screen | 23 |
| 2. | Video Panel | 24 |
| B. | ANDROID CLIENT | 29 |
| 1. | Menu Button | 29 |
| 2. | Videos | 31 |
| C. | CONCLUSIONS | 34 |
| V. | CONCLUSIONS AND FUTURE WORK | 35 |
| A. | CONCLUSIONS | 35 |
| B. | FUTURE WORK | 36 |
| | LIST OF REFERENCES | 39 |
| | INITIAL DISTRIBUTION LIST | 43 |

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

| | | |
|------------|--|----|
| Figure 1. | A typical VideoView example..... | 19 |
| Figure 2. | Existing Android Camera application..... | 21 |
| Figure 3. | Main screen..... | 24 |
| Figure 4. | Video panel..... | 25 |
| Figure 5. | Upload video flow..... | 26 |
| Figure 6. | Video from Mission..... | 26 |
| Figure 7. | Video display after clicking on an icon..... | 27 |
| Figure 8. | Video on map..... | 28 |
| Figure 9. | Expanded video information..... | 28 |
| Figure 10. | Video gallery of all mission videos..... | 29 |
| Figure 11. | Main screen with options menu..... | 30 |
| Figure 12. | Submenu options..... | 30 |
| Figure 13. | Video gallery..... | 31 |
| Figure 14. | VideoForm..... | 32 |
| Figure 15. | VideoView..... | 33 |
| Figure 16. | Save, move or cancel..... | 34 |

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|----------|---|
| AGIS | Advanced Ground Information Systems |
| API | Application Programming Interface |
| CAL FIRE | California Department of Forestry and Fire Protection |
| COTS | Commercial Off-the-Shelf |
| COP | Common operational picture |
| XMPP | Extensible Messaging and Presence Protocol |
| XML | Extended Markup Language |
| GIS | Geographical Information Science |
| GCCS | Global Command and Control System |
| GPS | Global Positioning System |
| GAE | Google App Engine |
| GWT | Google Web Toolkit |
| GUI | Graphical user interface |
| HTML5 | Hyper Text Mark-up Language 5 |
| HTTP | Hyper Text Transport Protocol |
| IT | Information Technology |
| IaaS | Infrastructure as a Service |
| HA/DR | Humanitarian Assistance/Disaster Relief |
| LDDRS | Lincoln Distributed Disaster Response System |
| NICS | Next Generation Incident Command System |
| OS | Operating system |

| | |
|---------|--|
| PaaS | Platform as a Service |
| POI | Point of Interest |
| SPARCCS | Smartphone Assisted Readiness, Command, and Control System |
| SIReS | Situation and Incident Report System |
| SA | Situational awareness |
| SITREP | Situational Reporting |
| SaaS | Software as a Service |
| SDK | Software Development Kit |
| TCP | Transmission Control Protocol |
| URL | Uniform resource locator |

ACKNOWLEDGMENTS

I would like to acknowledge the assistance and support of individuals who made this thesis possible. My advisors, Prof. Gurminder Singh and Prof. John Gibson helped me a lot with their valuable guidance and contributions. They made this thesis enjoyable and precious for me. Thanks a lot.

My love, Esra; the person who makes my life meaningful. Without her patience, intimacy and love, I could not overcome difficulties I faced. I love you, honey.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MOTIVATION

Destructive earthquakes, wide ranging environmental catastrophes, gigantic wildfires or large scale terrorist attacks may result in overwhelming consequences that neither small groups of people or a single nation can respond to alone. Such crises may require region-wide, nationwide or even international scale collaboration. However, this collaboration is inherently complex in itself. Routinely, creating situational awareness (SA) among the participants and evolving a general picture of the cooperative operation become two main concerns that affect effective decision making, which will save more lives with relatively less cost. Fortunately, by means of commercially available off-the-shelf (COTS) products, particularly with video capability, it is possible to address these challenges.

Although many definitions have been proposed so far, Mica R. Endsley describes "situational awareness" as:

[T]he perception of the elements in the environment within a volume of space and the time, the comprehension of their meaning, and the projection of their status in the near future [1].

She proposes that situational awareness is comprised of three hierarchical phases [2]. The first step is the perception of the elements of a given environment. Here, status, attributes and dynamics of the elements should be perceived. The second step is comprehension of the current situation. This involves the synthesis of disjointed

elements and understanding their significance. The third step is projection of the elements' future actions. Knowing the future possibilities will ease the decision making process [1].

Maintaining SA is a critical factor in successfully conducting a relief operation, because the situational parameters likely determine the ability of the decision maker to select an effective problem solving strategy [3].

The term common operational picture (COP) is defined by the U.S. Department of Defense as:

a distributed data processing and exchange environment for developing a dynamic database of objects, allowing each user to filter and contribute to this database, according to the user's area of responsibility and command role [4].

By definition, it is very interrelated with situational awareness. A common operating picture is the leading element in enhancing SA. It helps decision makers find answers for the questions, among many: what is happening? where is it happening? and who may contribute most?

As a consequence, both situational awareness and common operating picture are inevitable requirements for better response operation. Both shape the decision making process profoundly. Therefore, it is an urgent requirement to have gadgets in responders' hands which enable faster information sharing.

Today, the use of off-the-shelf hardware, such as smartphones and tablets, makes it possible to attain and maintain faster SA and provide real-time COP for relief

operations. Smartphones and tablets are the most recognized examples of the COTS devices used in disaster response and relief operations nowadays. Their availability and vast features make them indispensable to the relief operations. The popularity of smartphones/tablets among people continues to increase. According to studies, people increasingly want to have a smartphone/tablet. This makes every smartphone/tablet owner a potential responder in a relief operation. Moreover, smartphones/tablets help maintain quick and real-time SA and COP with their GPS, accelerometer and built-in camera functions. Their multimedia capabilities, like video and image capture, provide especially accurate and detailed information of benefit to SA and COP development and maintenance.

B. BACKGROUND

As smartphones/tablets continue to grow in their features, research into the capabilities of their usage increases. Today, studies based on usage of smartphones/tablets range from activity recognition [5] to coal mining [6]. One of the new applications of smartphones/tablets is in crisis management.

There are several systems in the market built for crisis management, which take advantage of benefits of smartphones/tablets in the operations field. The Smartphone Assisted Readiness, Command, and Control System (SPARCCS), is one of them, initially developed by two students from the Naval Postgraduate School for their thesis. This system leverages the resource-availability of cloud-computing and the utility of the Android smartphones/tablets to form a

complete system. The system offers two different user roles, web-client and Android-client. In both roles, responders are able to create a mission, establish a point of interest, attach relevant images for a mission or a point of interest and see these represented on a map. Although SPARCCS leveraged many of the smartphone/tablet capabilities, unfortunately, the system did not initially support video sharing among responders. For the video feature, extra coding remained to be added by the developers.

C. PURPOSE

SPARCCS serves two fundamental roles, with respect to these efforts. First, it provides a context within which responders explore the domain of command and control so as to understand how technologic and procedural tools can be leveraged to generate timely, context-based decisions that enable effective operations by military or disaster response organizations. Second, it provides a context within which the responders explore the architecture and implementation of mobile devices and their infrastructure necessary to support their operation. We believe that utilizing mobile video in any command and control system will help to create faster, more accurate SA and COP in a relief operation. It will provide detailed information about an incident to a decision maker. Thus, it will help a decision maker to choose an appropriate problem solving strategy. As a command and control system, SPARCCS should not be devoid of a mobile-video feature. In this thesis, we propose adding mobile-video capability to SPARCCS.

Therefore, we aim to enable both web-client and Android-client users to capture, share and display video.

D. ORGANIZATION

This thesis is organized into the following chapters:

- Chapter I provides the introduction and overview of the thesis.
- Chapter II describes existing command and control technologies built for a crisis management. The chapter highlights those features and their strengths and weaknesses.
- Chapter III summarizes the backbone structure of Smartphone Assisted Readiness, Command and Control System, and it describes the technology behind video feature.
- Chapter IV summarizes the earlier features of SPARCCS application, then describes the video capability of the system as an additional feature.
- Chapter V provides an objective summary of the video feature of the system and highlight possible improvements that can be made as further proposals.

THIS PAGE INTENTIONALLY LEFT BLANK

II. EVOLUTION OF HUMANITARIAN ASSISTANCE/DISASTER RELIEF SYSTEM

In the past several decades, the world has witnessed very destructive disasters, such as a tsunami in Japan (2011), an earthquake in Haiti (2010) and Hurricane Katrina in New Orleans (2005). They caused thousands of deaths and billions of dollars in property damage. To help keep both death tolls and the cost of damages low, an efficient response system should be deployed to such disaster areas. Thanks to technological improvements, systems are emerging which can be used in a Humanitarian Assistance/Disaster Relief (HA/DR) situation, to alleviate some of the problems of these tragedies. The Situation and Incident Report System (SIReS), Next Generation Incident Command System (NICS), Advanced Ground Information Systems (AGIS) and Smartphone Assisted Readiness, Command, and Control System (SPARRCS) are among those systems which are currently available for HA/DR missions.

In this chapter, we provide an introduction to these systems and related technologies.

A. SIReS—SITUATION AND INCIDENT REPORT SYSTEM

One implementation of an incident-report system is Northrop Grumman's Situation and Incident Report System. SIReS utilizes smartphone technology to provide real-time reports from on-site personnel to automatically populate command center databases [7]. Along with real-time situational reporting (SITREP) with embedded pictures and video, the SIReS system has Blue Force tracking and barcode scanning, which may be used for evacuee tracking

capabilities. However, the system has some vulnerability from a structural point of view. Using existing cellphone infrastructure for data flow and a server-based approach are two such vulnerabilities.

SIReS utilizes the commercial cell-phone infrastructure to create SA and COP. However, existing infrastructure can be rendered useless after severe disasters. This happened, for example, when a tsunami hit Japanese coastal cities in May, 2011. After the tsunami, the infrastructure of cities in that area of Japan was completely damaged. Therefore, it is important to use a backup wireless data service in conjunction with existing ones.

Another weakness of SIReS lies in its server-based structure. SIReS uses Extensible Messaging and Presence Protocol (XMPP), which supplies messaging, presence and Extended Markup Language (XML) routing features, and a Geographical Information Science (GIS) database in order to manipulate geographical data. Hence, the effectiveness of SIReS is restricted with customization of server and database. In contrast to server-based structure, cloud-based structure may increase the availability of system.

B. NICS—NEXT GENERATION INCIDENT COMMAND SYSTEM

The Next Generation Incident Command System, formerly called the Lincoln Distributed Disaster Response System (LDDRS), was developed by the Massachusetts Institute of Technology in partnership with the California Department of Forestry and Fire Protection (CAL FIRE) to enhance emergency response capabilities [8] in the USA. The system aims to increase the situational awareness of first-

responders and collaboration between participants by leveraging a web-based platform and a map-based environment. However, both features carry some problems.

NICS is a web-based platform which allows participants to login via a web browser over a wireless network. Information about an incident, which can be real-time feed of airborne images, video, weather and critical infrastructure as well as terrain information, is displayed on the client's browser. However, a display format for the system is not available for a smartpone, which hinders rendering issues. Moreover, a mobile user is not able to store information in a local database. This may dramatically affect the efficiency of the system due to the fact that it causes rapid energy consumption at the mobile end.

NICS has a map-based environment. This environment provides several graphical tools. It provides georeferenced virtual whiteboards, which can be used for collaboration. This tool facilitates communication among responders. For instance, a responder can create a team, send messages to another responder, and share maps and drawings with this tool. However, the map-based environment is restricted to the California region. Therefore, NICS is not applicable throughout the nation or world.

C. AGIS—ADVANCED GROUND INFORMATION SYSTEMS

Advanced Ground Information Systems developed a server-based emergency communication system between peers, which AGIS called LifeRing [9]. The system was started in 2004 and has been under development for seven years. With

its current capabilities, LifeRing provides useful features in order to create common operational pictures.

LifeRing's capabilities offer a wide variety of options to users. Firstly, the system leverages a combination of various wireless data services in the field of operation. It utilizes cellular network, Wi-Fi, military and land mobile radio, and satellite communication. Secondly, in AGIS, communication between the responder is encrypted. The system utilizes an advanced standard algorithm for encryption. Thirdly, the system is compatible with approximately 50 different types of maps, including known military maps. It is possible to switch between maps.

LifeRing operates on Microsoft PC and Android-based smartphones/tablets. Using a PC and smartphone, the user is able to view his or her and the participants' location, declare an emergency message which appears on all participants' displays, and enter georeferenced text and images. Moreover, the user is able to send messages, open chat sessions, view video, and open whiteboards. On the other hand, these features can be limited by an administrator.

D. SPARCCS—SMARTPHONE ASSISTED READINESS, COMMAND, AND CONTROL SYSTEM

The Smartphone Assisted Readiness, Command, and Control System (SPARRCS) is an implementation of command-and-control technology described in a previous thesis done at the Naval Postgraduate School [10]. The system leverages cloud computing and commercial off-the-shelf (COTS) products. These features bring great advantages from an HA/DR operations perspective.

SPARRCS is a Google cloud-based system. It uses the Google Web Toolkit (GWT) and Google App Engine (GAE). The GWT provides a developer environment for common object-oriented code to be repurposed and optimized into web browser-independent applications. The GAE provides a platform for developing and hosting web applications in Google managed data centers. With cloud computing, SPARRCS can easily overcome the issues encountered when dealing with a client-server based system. Moreover, SPARRCS takes advantage of Google Maps as a tool in order to create a common operational picture. Since Google Maps covers the entire world map, SPARRCS can be deployed easily throughout the world without having to preload maps, which makes it completely different from other systems that we have covered. The downside of this approach is that SPARRCS clients need to download maps in real-time, as needed, which makes the bandwidth requirement high in what might be a fairly constrained networking infrastructure.

Using COTS mobile devices in order to create situational awareness among the participants in HA/DR operations is becoming common. Among those COTS phones, Android-based mobile devices such as smartphones and tablets lead the space, especially in the USA. According to results of a recent survey [11], 50 percent of 104 million subscribers in the USA are using Android-based smartphones. SPARRCS utilizes the Android-based smartphone/tablet as the target user-device for which to create missions, points of interest and display images for the sake of furthering situational awareness.

E. SUMMARY

Disasters such as tsunamis, earthquakes, hurricanes, etc. need an effective response system to be deployed in order to help manage response efforts so that more lives might be saved and assets are better managed to reduce costs. There are several systems currently being used for HA/DR, such as SIREs, AGIS' LifeRing and NICS. SPARRCS differs from the others by being designed explicitly to leverage mobile and cloud computing potential. By adding video sharing capability, the system can be even more efficient for usage.

III. INFRASTRUCTURE BACKGROUND

The efficiency of SPARRCS is based on the prominent technology it was built upon, which differs from other HA/DR implementations. In this chapter, we will describe the core infrastructure of SPARRCS which uses cloud computing, the Google App Engine and the Google Web Toolkit, and then discuss the background of mobile video capability improvement added for the purposes of the thesis, which includes BST player [12], VideoView and Camera application programming interfaces (API).

A. CLOUD COMPUTING

SPARRCS is a cloud computing implementation. The term "cloud computing" is a new style computing model that needs to be clearly defined to understand SPARRCS' advantages with regard to the HA/DR field.

According to the National Institute of Standards and Technology, cloud computing is defined as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." Cloud computing offers explicit benefits as compared to client-server computing for HA/DR operations; significant among them are centralized data storage, faster management and reliable backups or data recovery.

Cloud computing is also a popular trend in information technology (IT) and electronic commerce vendors. Major

players in this field, such as Amazon, Microsoft and Google, provide different service models to their customers. It is possible to categorize these service models in three groups.

In the Software as a Service (SaaS) model, the consumer is able to use the provider's applications which run on a cloud infrastructure. This is typically actualized either via a web browser or a special program interface. Mail services like G-mail and Hotmail are good examples of this type of service model.

In the Infrastructure as a Service (IaaS) model, the consumer is able to deploy/run arbitrary software over the provided computing resources, such as servers, storage, operating systems, networks, etc. Implementations like Amazon EC2 and Rackspace are examples.

With the Platform as a Service (PaaS) model, consumers are allowed to deploy their own applications created using specialized programming languages, tools, and libraries supported by the providers. The Google App Engine and Google Web Toolkit are two such services provided by Google allowing the consumers to develop their own applications for PaaS implementation. SPARCCS, itself, is a good example in this category. It was developed by NPS students as a thesis project and deployed on the Google Cloud service in order to respond to specific HA/DR mission needs. The application, itself, can be configured by developers for further improvement. This thesis has expended the SPARCCS functionality by adding mobile video capability, which will be discussed later.

In conclusion, cloud computing has inherent advantages for HA/DR missions, including faster management, continuity of service, and flexibility. Currently, of the three major service models in the market, SPARRCS takes advantage of the PaaS model provided by Google.

B. GOOGLE APP ENGINE AND GOOGLE WEB TOOLKIT

Google App Engine (GAE) and Google Web Toolkit (GWT) are two tools, provided by Google, used by NPS thesis students to develop and configure SPARCCS to operate on the Google infrastructure.

Google App Engine is a platform which lets the developers run web applications on the Google Cloud service. With GAE, the processes of building, maintaining and scaling user created applications are easy. It supports applications written in the Java, Python and Go programming languages, the latter being a Google sponsored open-source language [13]. The Java programming language is used for the web service side of SPARRCS; the Java runtime environment used is Java 6. The Google App Engine SDK for Java supports developing applications using either Java 5 or 6. From a cost perspective, 1 GB of storage and 5 million page views a month are provided for free.

Google Web Toolkit is a development toolkit for building and optimizing complex, browser-based applications [14]. It has four elements. Firstly, a software development kit (SDK) that eases the problem that developers run into while writing web apps for multiple browsers. Secondly, Speed Tracer that helps diagnose problems in the web applications by visualizing metrics. Thirdly, Eclipse

Plugins that enable developers to quickly design, build, optimize and deploy cloud-based applications written in the Java Environment [15]. Lastly, GWT Designer, a bi-directional Java graphical user interface (GUI) that saves time by using an auto-generating Java code feature. However, from the thesis perspective, the developers of SPARCCS did not use this final feature.

C. ANDROID OPERATING SYSTEM

SPARRCS leverages COTS product advantages to rapidly develop and deploy application capabilities for the HA/DR operation field. Currently, the system utilizes the Android-based mobile operating system (OS) on a smartphone/tablet for the mobile end-user host platform.

Although Android's history dates back to 2003, it earned its existing fame as a mobile OS when Google stepped into this market with its company, Open Handset Alliance [16]. Currently, Android-based smartphones share 36% of the market and trends indicate that this share will significantly increase in the near future. This brings visible efficacy to SPARCCS from the responder's perspective. It means responders will immediately be ready to join the HA/DR mission because of existing Android-based smartphones in their hands.

Android-based applications may contribute HA/DR operations with Android's key components. It provides various methods for information input. With its media capture capability, it allows video, audio and picture input from its built-in camera and microphone. We will discuss the Android video feature in following sections.

Also, GPS and accelerometer hardware provide crucial data, such as location information and real-time actions of responders. Moreover, Android supports common wireless data services including: Bluetooth, Wi-Fi, GSM, EDGE, 3G and 4G. By means of these services, Android supplies broad alternatives for transferring collected information. Further, Android supplements any database implementation with its own database, SQLite. In SPARCCS, SQLite is used to sync information on the mobile-end with information in the cloud.

All in all, the Android OS presents reasonable benefits to SPARCCS developers for meeting the basic situation awareness and analysis requirements of the HA/DR field. Detailed information about the Android video feature will be described in following sections.

D. WEB SERVICES

Web Service is a software system allowing machine-to-machine interaction over a network. In SPARCCS, interaction between an Android-based smartphone and a server is implemented using HTTP Posts and HTTP Requests using HTTP Servlets. On the other hand, for web-client to server interaction, Remote Procedure Calls are used.

E. BST PLAYER

Originally, SPARCCS was devoid of video capture and sharing/displaying features. We believed that adding the video capability to the current system would multiply its efficacy and make it more useful from a situational awareness perspective. For this purpose, on the web-client

end, we chose the third party API built by Apache Maven, which is a software project management and comprehension tool, BST Player, to add video playback capability.

Currently, the video API provided in GWT SDK, is under development and subject to change. As a third party API, the BST Player fills this gap with some unique features. The BST Player API makes most common media player plug-ins available to GWT applications [17]. It supports Windows Media Player, QuickTime, Flash Player, VLC Media Player, HTML 5 Video and DivX Web Player. This feature helps overcome a usual handicap that developers encounter as a result of conflicting video format. Moreover, the API allows developers to customize an HTML-based player and use it as a cross-browser media player. Therefore, developers are able to build their own skin interface for general-purpose application.

Consequently, the BST Player API provides necessary features for video support. It allows users to watch common video formats and create customized interfaces.

F. ANDROID VIDEO FEATURE

As previously mentioned, the Android OS video framework supports the display and capture of video in different formats. Therefore, it is easy to integrate its video feature into any application. By using the MediaPlayer or VideoView APIs, Android allows playing video from various sources, such as media files stored in an application's resources, files in a file system or even a data stream arriving over a network connection [18]. Additionally, by utilizing MediaRecorder or Camera APIs,

Android allows the recording of video clips, of duration limited by the device hardware support. For the SPARCCS project, we have implemented VideoView and Camera APIs. Therefore, we will discuss the features of these two APIs.

1. VideoView

The VideoView (Figure 1) is a class which displays a video file. It can load videos from designated sources, such as resources or content providers. It can be used in any layout manager and provides various display options, such as scaling [19].



Figure 1. A typical VideoView example

To display video, the string format of the path for the designated video source should be configured into `setVideoPath`, a submethod of the `VideoView` class. In the SPARCCS project, the path is typically a uniform resource locator (URL) of a video source located in the Google Cloud infrastructure.

The Display feature provided by the `VideoView` class offers adequate options for simple video implementation. Its `onMeasure` method determines the width and the height of the video and lets the developers resize the video dimensions so that the video is properly adjusted for any size screen.

In conclusion, `VideoView` class is a simple solution for displaying video on Android-based smartphones in the SPARCCS project. With methods provided under the `VideoView` class, any obstacles based on video dimensions can be resolved for any screen size.

2. Camera

Android supports various cameras and camera-features available on differing devices. Primarily, the API that controls device cameras is provided under the name of "Camera." Developers can either customize their application through Camera API or utilize the existing Camera application provided by default. For our purposes, we take advantage of the existing Android Camera application.

Using the existing Android Camera application (Figure 2) is a quick way to capture video with minimal coding. Invoking the camera requires initiation of an abstract operation, called "intent." To access the device's

camera and its camera features, "Camera Permission" and its elements should be declared in the manifest file. By default, Android security architecture design does not allow any application to impact other applications [20]. In order to do that "permission" should be declared explicitly in the manifest file. The manifest file presents the essential information about the application to the Android system [21] such as describing components of the application, determining which process will host the application and declaring the minimum level of the Android API that the application requires.



Figure 2. Existing Android Camera application

G. SUMMARY

This chapter reviewed key technologies supporting the earlier SPARCCS project and then discussed further technologies available to improve the system by incorporating a video capture and video display capability.

Cloud computing, the Google App Engine, the Google Web Toolkit web services, and the Android OS are the keystone technologies that provide a variety of capabilities and rapid prototyping to SPARCCS. Using the cloud model, particularly the PaaS model, offers faster management, continuity of operations and flexibility. Moreover, GAE and GWT simplify creating and maintaining a web application in the Google Cloud infrastructure. Furthermore, HTTP web services provides secure machine-to-machine communication in SPARCCS. Finally, the Android OS meets the general requirements of HA/DR missions with its versatility.

As for improvements made to the system, key application programming interfaces such as BST player, VideoView and Camera, which make video capture, sharing and display available in SPARCCS, are discussed. BST player and VideoView help display common video formats for web-client and Android smartphone entities, respectively. Finally, Camera API enables video capture through Android smartphone. For next chapter, we will provide detailed information about these new additions from user interface perspective.

IV. EXTENSIONS OF SPARCCS

In previous chapters, we have discussed current HA/DR implementations in the market, and the technology in SPARCCS. As earlier features of SPARCCS have been explained by developers in their theses, here we are going to explain new features of SPARCCS for the purpose of this thesis. Since SPARCCS offers two different client models, web-client or mobile client, we will explain new additions on both sides, one at a time.

A. SPARCCS WEB-CLIENT

On the web application side of SPARCCS, we have adhered to the original user interface flow. Therefore, while we added our extensions, we followed a similar structural concept used for the image module. Hence, integrity of the web-browser-based version has been preserved.

1. Initial Screen

The user is taken to the main interaction screen (Figure 3), after the login process. The main screen consists of two principal panels, a map panel and a navigational panel. On the right side of the screen is the user map, centered on the user's current location. As an additional feature, video icons are now displayed on the map, along with the mission, points of interest, responder and image icons. Thus, the user is now aware of video captures sent from responders.

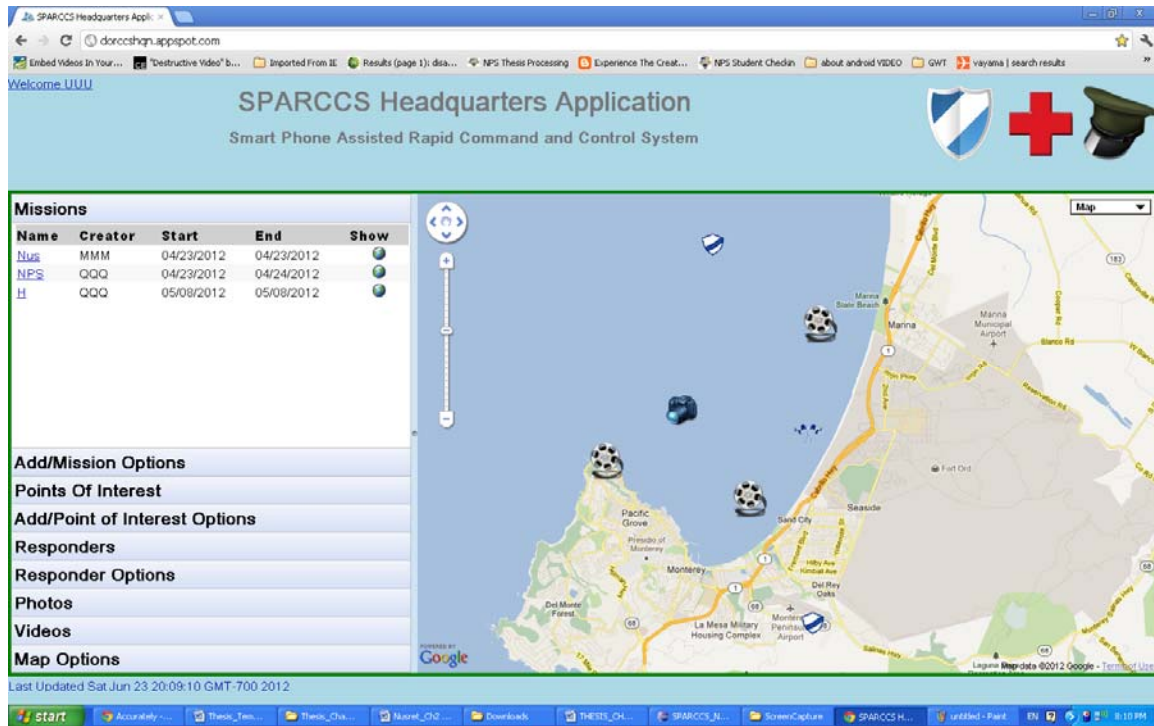


Figure 3. Main screen

2. Video Panel

Like other panels in the navigational panel, the Video panel (Figure 4) consists of three main sections: the Mission section, the Points of Interest section and the Responders section.

The Mission section contains three radio buttons and a list box. The radio buttons enable the user to add a video capture to a selected mission from the list box, to view the video capture from a selected mission from the list box or to view all the videos from all missions.

| |
|---|
| Missions |
| Add/Mission Options |
| Points Of Interest |
| Add/Point of Interest Options |
| Responders |
| Responder Options |
| Photos |
| Videos |
| Missions |
| <input type="radio"/> Add Video to a Mission <input type="radio"/> See Videos From a Mission <input type="radio"/> See All Mission Videos |
| MISMMM11153604232012 Nus MISCQQ185204232012 NPS MISMQQ12115205082012 H |
| Points of Interest (POI's) |
| <input type="radio"/> Add Video to a POI <input type="radio"/> See Videos From a POI <input type="radio"/> See All POI Videos |
| POICEE01042204292012 map poi MISMMM1115360423201 |
| Responders |
| <input type="radio"/> See Videos From a Responder |
| MMM QQQ EEE UUU |
| Map Options |

Figure 4. Video panel

In order to add a video capture to a selected mission, first, the user should select one of the current missions in the list box, and then select the "Add Video to a Mission" radio button. This will bring up a video form. To successfully upload a video, the user must fill in the "Name," "Description," and "Latitude" and "Longitude" text fields, and then click on the "Submit" button. This flow is illustrated in Figure 5.

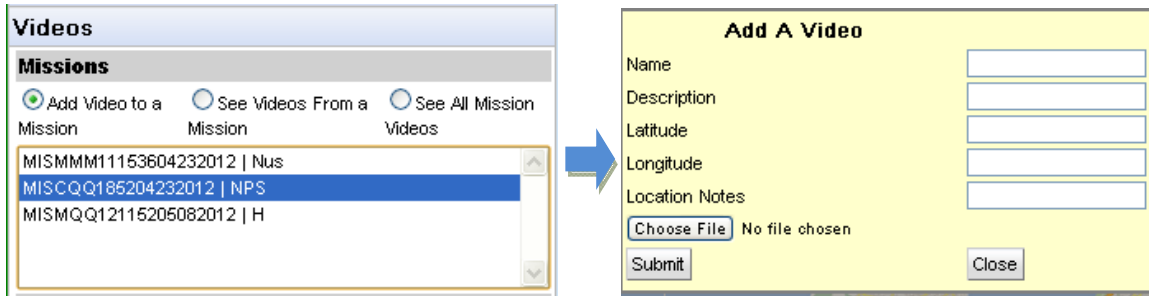


Figure 5. Upload video flow

With respect to viewing videos from a particular mission, a similar process is followed. First, the user selects a mission from the list box, and then selects the "See Videos From a Mission" radio button. This will bring up a gallery of standard video icons that represent, individually, unique video captures from the mission (Figure 6). Descriptions of the videos will be displayed in a popup window if a related icon is "moused-over."

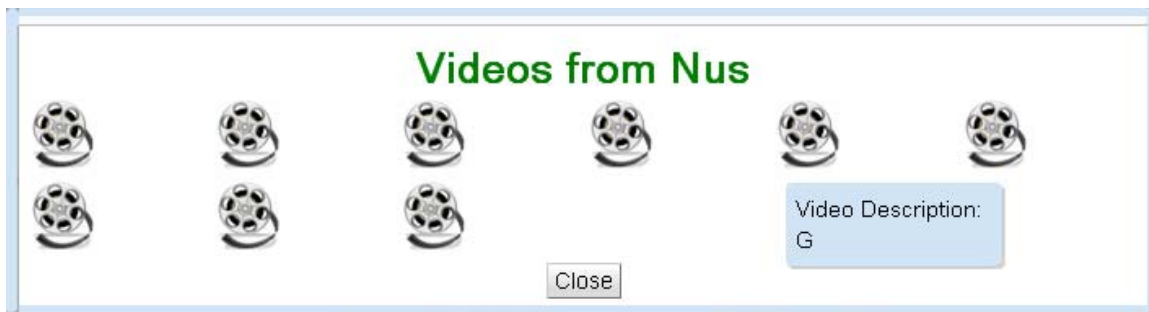


Figure 6. Video from Mission

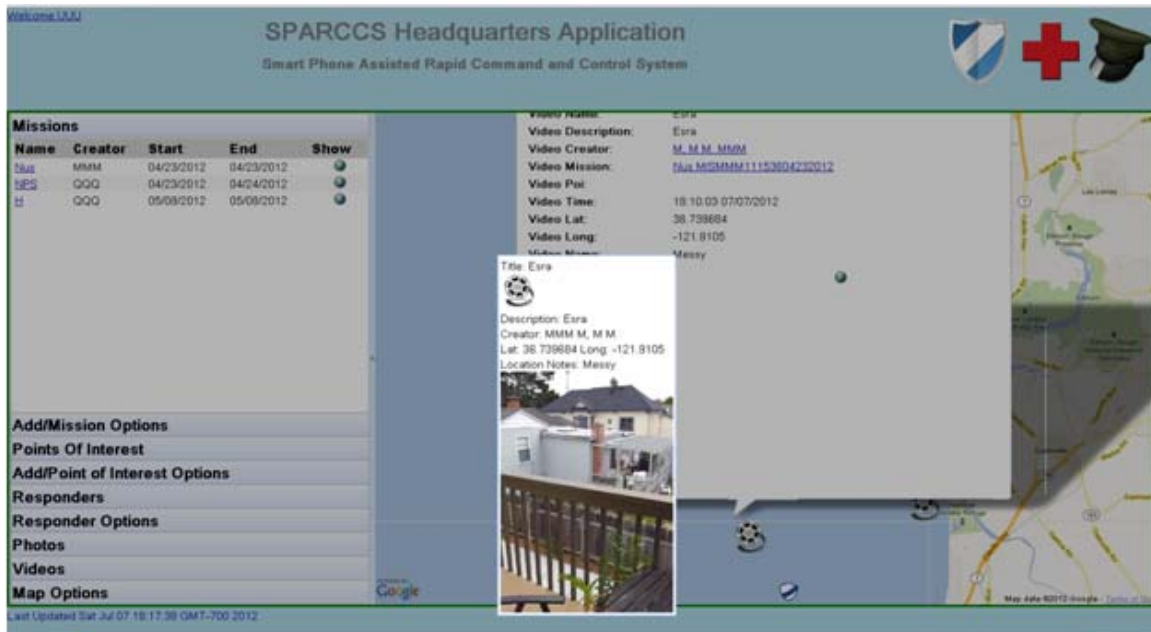


Figure 7. Video display after clicking on an icon

If an icon is clicked on, the video will be displayed in a Simple panel (Figure 7). If the current user is the creator of the mission, then the user has the option of deleting the video.

There is also a globe button on this panel. This button directs the user to the video's location on the map and allows the user to see essential information about the video within a bubble (Figure 8). If this bubble is expanded, all information about the video will be displayed, as shown in Figure 9. The user can view the video by clicking the video icon.

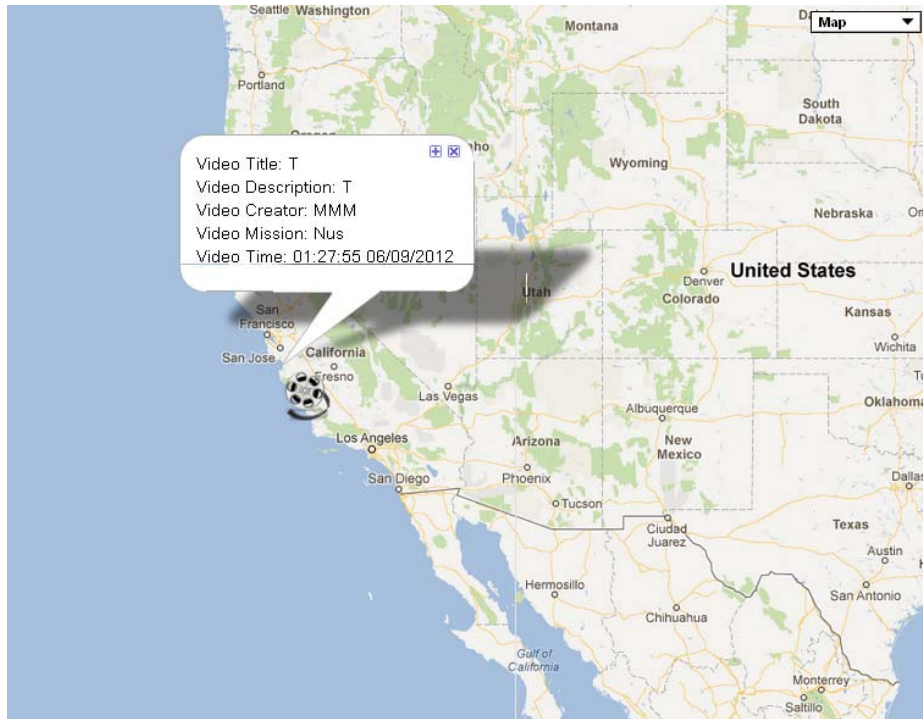


Figure 8. Video on map

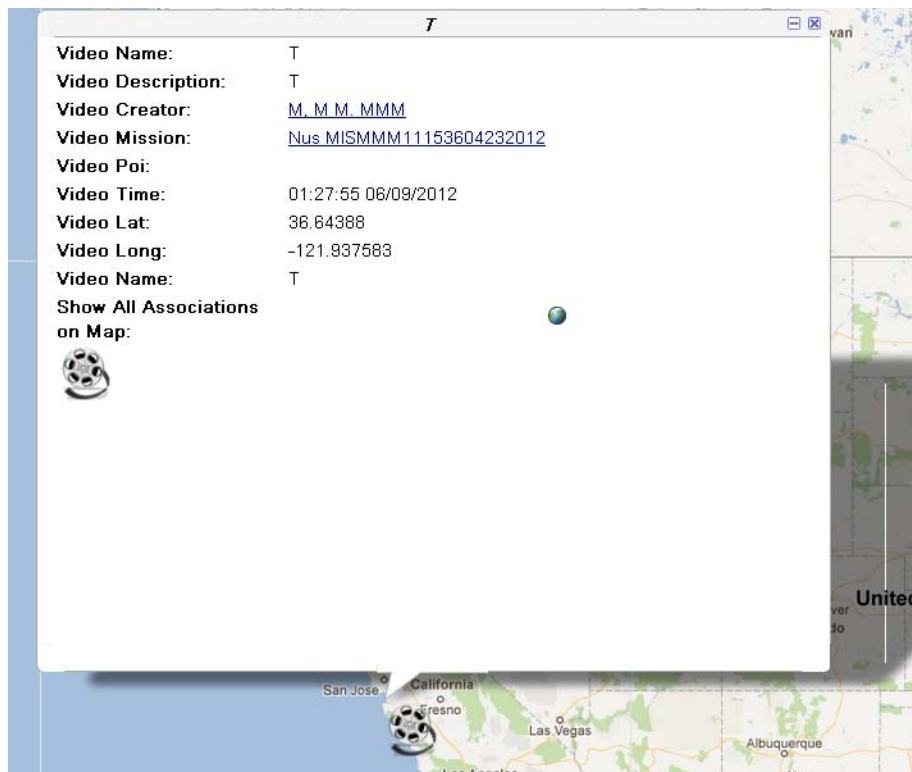


Figure 9. Expanded video information

To see all videos from all missions, the user simply clicks on the radio button "See all Mission Videos" (Figure 10), following the same actions from within the video gallery.



Figure 10. Video gallery of all mission videos

B. ANDROID CLIENT

1. Menu Button

Due to its limited screen size, there is no navigational panel on the main screen of the SPARCCS' Android addition; all functionalities are accessed by pressing the "Menu" button. In an earlier version, user options were limited to: "POI" (points of interest), "Users" (responders), "Missions" and "Image." We added "Video" as a fifth option for the user, as shown in Figure 11.

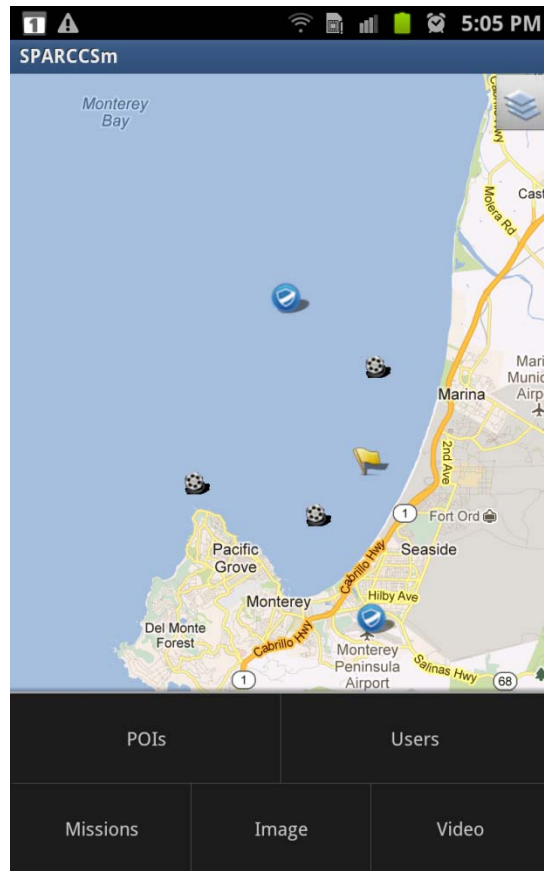


Figure 11. Main screen with options menu

As with other options, pressing the "Video" option will bring up a submenu (Figure 12). This submenu offers two different methods of action, "From Gallery" and "Take." "From Gallery" is self-explanatory; it takes the user to the video gallery, a default location for video captures (Figure 13). "Take" opens the device's camera application.

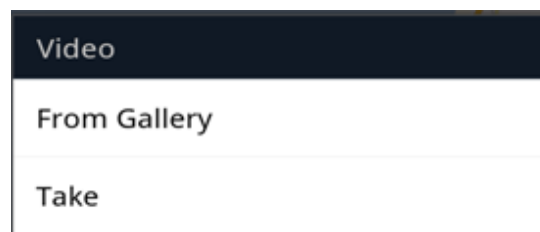


Figure 12. Submenu options



Figure 13. Video gallery

2. Videos

As previously mentioned, there are two ways for video to be entered into the system; either by using the device's native camera or by selecting "Video Capture" in the video gallery.

Through the native camera, captured video is saved to the default folder, video gallery. The VideoClient class retains the path to the video until the video is uploaded to the cloud. Once it is successfully uploaded, an URL is returned for the video's newly assigned path on the server. Immediately after capturing a, VideoForm is opened.

The VideoForm screen displays video icon, form, and function buttons, from top to bottom (Figure 14). At the top is a video icon that contains a link to the video. By tapping on the icon, captured video is viewed by the VideoView class (Figure 15). Beneath the video icon there are three forms that allow related information about the video to be provided by the user, such as name, description and location notes.

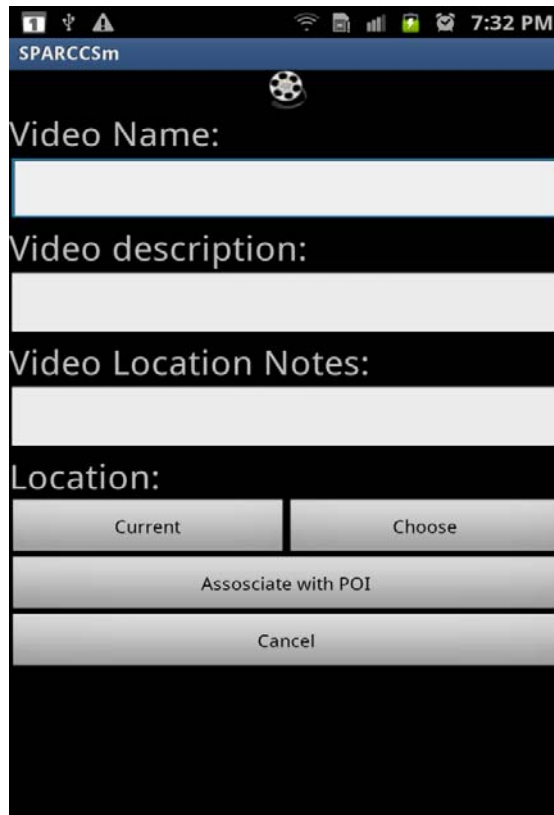


Figure 14. VideoForm



Figure 15. VideoView

At the bottom, three buttons offer different options to the user about location of the video. The user can choose either to place the video in its current location or on another location on a map, or the user can choose to associate the video with a point of interest. Distinct from the third option, either of the first two options places a video icon on the map. An icon placed on the map is itself a link to the video. The user can see information related to the video by tapping the icon on the map. This action will open the VideoForm class instance. Additionally, both buttons will open further options for the user to save, move or cancel the video (Figure 16).

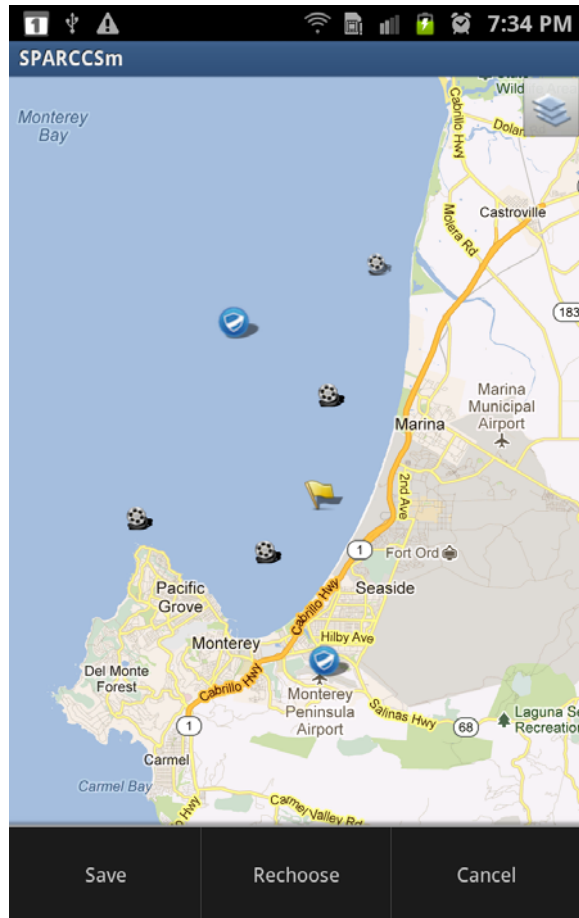


Figure 16. Save, move or cancel

C. CONCLUSIONS

With the new extensions, SPARCCS is now able to support video capture, display and sharing among responders. Responders can share video about an incident by means of either the web-based application or the Android application. With the help of video capability, SPARCCS will provide more detailed information and create a more accurate operational picture for the HA/DR missions. Thus, response to any specific incident may be conducted more properly.

V. CONCLUSIONS AND FUTURE WORK

A. CONCLUSIONS

Mobile video for HA/DR missions plays an important role in enhancing situational awareness among participants. A simple captured video from the operation field provides detailed information about an incident. Hence, it may lead to better understanding of the particular situation faced in an operation. Therefore, any command and control system designed to create a COP in order to maintain complete SA should benefit from mobile video. Although SPARCCS provides excellent features for COP, initially it was devoid of video capture, share and display features. In this thesis, we added this feature and enhanced the effectiveness of SPARCCS one step further.

While programming the video capability, we faced several difficulties. We resolved some of them, but not all. There were some trade-offs that we had to decide. First of all, SPARCCS communication infrastructure between mobile hosts and the cloud server became an obstacle for us. As previously mentioned, SPARCCS utilizes the Google Cloud platform. For security reasons, communication between responders and the cloud happens via HTTP connections. HTTP uses Transmission Control Protocol (TCP) as its underlying transport protocol. TCP provides a reliable, connection-oriented service to the invoking application [22]. However, TCP's congestion control feature may affect video transmission adversely when packet loss rate is higher than

tolerated. Therefore, when a network is in a stressed state, it may take a relatively long time to transfer or display video in SPARCCS.

Secondly, we encountered some problems when dealing with smartphone characteristics. Smartphones manufactured by different firms, even though they use the same Android OS, their characteristics differ from each other. From a video point of view, the screen size, resolution and color quality of the device became main concerns for us. A video capture with high resolution sometimes may cause problems on a receiving smartphone which has lower screen resolution or less computing power. We dealt with this problem by using factory settings as a reference for our coding. Thus, SPARCCS may continuously leverage features of upcoming smartphones in the future.

In conclusion, we enhanced the efficiency of SPARCCS by adding video capture, share and display capability in this thesis. We added our coding both on the server and on the web application and mobile application side. Both web user and mobile user are now able to capture video, share it with participants and view it. With this additional feature, SPARCCS provides faster situational awareness and more functionality in creating COP. SPARCCS is better ready to meet general requirements in HA/DR operations.

B. FUTURE WORK

Although we enhanced the utility of SPARCCS, there are several things that can possibly be done to further enhance its functionality. To that end, we have provided a list of possible future upgrades.

- 1) Picture editing—Augmenting further information to shared images will help convey intended messages and extend usefulness. For instance, drawing arrows or circles or scripting notes on an image will help draw attention of users to the point or information the sender wishes to emphasize.
- 2) Different mobile OS—SPARCCS leverages the functionality of Android-based smartphones in the operation field. However, Android is not the only operating system for the smartphones. There are several mobile operating systems such as Apple iOS, Windows 8 and Blackberry 7.1, which may provide similar functionality for SPARCCS. We believe that if mobile application of SPARCCS is enhanced through these operating systems, the number of potential responders will increase. Thus, availability of SPARCCS will be affected in a good way.
- 3) HTML5—Coding native applications for mobile users adds an extra burden from a system management perspective. As smartphone characteristics vary from screen size, resolution, and processing capability, the compatibility of the application becomes a big issue. It is possible to deal with this problem by converting SPARCCS into a device-independent application. Although it is still under development, HTML5 promises new opportunities for a device-independent application. Currently, HTML5 does not meet all

the functionality of SPARCCS. However, as it improves, it is likely to meet all functionality. At that point, an HTML5-coded version of SPARCCS will be more efficient and flexible, regardless of which device the participants have.

- 4) Real-time video sharing—Although adding video capture, share and display capability made SPARCCS more suitable for HA/DR missions, it is also useful to have real-time video streaming from the operation field. By receiving real-time feeds, decision makers at distant headquarters will be better able to follow the changing situation in the field.

LIST OF REFERENCES

- [1] M. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human Factors*, no. 1, pp. 32-64, March 1995.
- [2] C. Bolstad and T. Hess, "Situational awareness and older workers," in *Experimental Analysis and Measurement of Situational Awareness Conf.*, Daytona Beach, FL, 1995.
- [3] D. Jones and M. Endsley, "Investigation of Situation Awareness Errors," in *8th Int. Symp. on Aviation Psychology*, Columbus, OH, 1995.
- [4] Department of Defense. (2008, October 31). *Chairman of the Joint Chiefs of Staff Instruction 3151.01B Global Command and Control System Common Operational Picture Reporting Requirements* [Online]. Available: www.dtic.mil/cjcs_directives/cdata/unlimit/3151_01.pdf
- [5] M. Keally et al., "PBN: Towards practical activity recognition using smartphone-based body sensor networks," in *Proc. of the 9th ACM Conf. on Embedded Networked Sensor Systems (SenSys '11)*, Seattle, WA, 2011, pp. 246-259. doi: [10.1145/2070942.2070968]
- [6] A. Samanta et al., "PhoneGuard: A smartphone in the coalmine," in *Proc. of the 1st Int. Conf. on Wireless Technologies for Humanitarian Relief (ACWR '11)*, Kerala, India, 2011, pp. 301-308. doi: [10.1145/2185216.2185299]
- [7] R. Renner et al., "Situation and incident reporting system," in *Proc. of the 2nd Int. Conf. on Computing for Geospatial Research & Applications (COM.Geo '11)*, Washington, D.C., 2011. doi: [10.1145/1999320.1999364]
- [8] Lincoln Laboratory. (2011). *Next generation incident command system* [Online]. Available: http://www.ll.mit.edu/publications/technotes/TechNote_NICS.pdf
- [9] Advanced ground information systems. (2010). *LifeRing operators manual* [Online]. Available: http://www.agisinc.com/AGIS_Operators_manual.pdf

- [10] M. Asche and M. Crewes, "Utilizing Android and the cloud computing environment to increase situational awareness for a mobile distributed response," M.S. thesis, Dept. Computer Science, Naval Postgraduate School, Monterey, CA, 2012.
- [11] comScore Inc. (2012, April 3). *Android captures majority share of U.S. smartphone market* [Online]. Available:
<http://search.proquest.com/docview/967153160?accountid=12702>
- [12] Google. (2012). *Bst-player* [Online]. Available:
<http://code.google.com/p/bst-player/>
- [13] Google. (2012). *The Go programming language* [Online]. Available: <http://golang.org/>
- [14] Google. (2012). *Google web toolkit overview* [Online]. Available:
<http://code.google.com/webtoolkit/overview.html>
- [15] Google. (2012). *Google plugin for Eclipse* [Online]. Available:
<https://developers.google.com/eclipse/index?hl=tr>
- [16] Google. (2012). *Philosophy and goals* [Online]. Available:
<http://source.android.com/about/philosophy.html>
- [17] Bramo systems. (2012). *BST player* [Online]. Available:
<http://oss.bramosystems.com/bst-player/>
- [18] Android. (2012). *Media and camera* [Online]. Available:
<http://developer.android.com/guide/topics/media/index.html>
- [19] Android. (2012). *VideoView* [Online]. Available:
<http://developer.android.com/reference/android/widget/VideoView.html>
- [20] Android. (2012). *Permissions* [Online]. Available:
<http://developer.android.com/guide/topics/security/permissions.html>

- [21] Android. (2012). *The AndroidManifest.xml File*
[Online]. Available:
<http://developer.android.com/guide/topics/manifest/manifest-intro.html>
- [22] J. Kurose and K. Ross, "Application layer," in
Computer Networking: A Top-Down Approach, 5th ed.,
Boston: Addison-Wesley, 2010, p. 101.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California
3. Turkish Naval Academy Library
Turkish Naval Academy
Tuzla, Istanbul TURKEY
4. Middle East Technical University Library
Middle East Technical University
Cankaya, Ankara TURKEY