3-24-2016

A Service Oriented Architecture Approach for Global Positioning System Quality of Service Monitoring

Stuart A. Everson

Recommended Citation

Everson, Stuart A., "A Service Oriented Architecture Approach for Global Positioning System Quality of Service Monitoring" (2016). *Theses and Dissertations*. 392.
https://scholar.afit.edu/etd/392

A Service Oriented Architecture Approach for Global Positioning System Quality of Service Monitoring

THESIS

Stuart A. Everson, Capt, USAF

AFIT-ENV-15-MS-16-M-149

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENV-15-MS-16-M-149

A SERVICE ORIENTED ARCHITECTURE APPROACH FOR GLOBAL POSITIONING SYSTEM QUALITY OF SERVICE MONITORING

THESIS

Presented to the Faculty
Department of Systems Engineering and Management
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Systems Engineering

Stuart A. Everson, B.S. Wireless Engineering
Capt, USAF

March 7, 2016

AFIT-ENV-15-MS-16-M-149

A SERVICE ORIENTED ARCHITECTURE APPROACH FOR GLOBAL POSITIONING SYSTEM QUALITY OF SERVICE MONITORING

THESIS

Stuart A. Everson, B.S. Wireless Engineering
Capt, USAF

Committee Membership:

Lt Col Thomas C. Ford, PhD
Chair

John M. Colombi, PhD
Member

David W. Meyer, LCDR, USN, Ret.
Member

AFIT-ENV-15-MS-16-M-149

Abstract

This research focuses on the development of a Service Oriented Architecture (SOA) for monitoring the Global Positioning System (GPS) Standard Positioning Service (SPS) in near real time utilizing a Mobile Crowd Sensing (MCS) technique. A unique approach to developing the MCS SOA was developed that utilized both the Department of Defense Architecture Framework (DoDAF) and the SOA Modeling Language (SoaML) guidance. The combination of these two frameworks resulted in generation of all the architecture products required to evaluate the SOA through the use of Model Based System Engineering (MBSE) techniques. Ultimately this research provides a feasibility analysis for utilization of mobile distributed sensors to provide situational awareness of the GPS Quality of Service (QoS). First this research provides justification for development of a new monitoring architecture and defines the scope of the SOA. Then an exploration of current SOA, MBSE, and Geospatial System Information (GIS) research was conducted. Next a Discrete Event Simulation (DES) of the MCS participant interactions was developed and simulated within AGI's Systems Toolkit. The architecture performance analysis was executed using a GIS software package known as ArcMap. Finally, this research concludes with a suitability analysis of the proposed architecture for detecting sources of GPS interference within an Area of Interest (AoI).

Table of Contents

Page

List of Figures

Figure Page

Figure Page

List of Abbreviations

Abbreviation		Page
PNT	Position, Navigation, and Timing	1
GPS	Global Positioning System	1
FAA	Federal Aviation Administration	1
DHS	Department of Homeland Security	1
USAF	U.S. Air Force	2
QoS	Quality of Service	2
SIS	Signal in Space	2
MCS	Mobile Crowd Sensing	2
AoI	Area of Interest	2
IOC	Initial Operational Capability	2
DoD	Department of Defense	2
PGM	Precision Guided Munition	2
ISR	Intelligence, Surveillance, and Reconnaissance	2
SA	Selective Availability	2
ToA	Time of Arrival	3
MEO	Medium Earth Orbit	3
SIS	Signal in Space	3
PPS	Precise Position Service	3
OCSs	Operational Control Segments	3
SPS	Standar Positioning Service	3
SMC/GP	Space and Missile Center GPS program office	3
MCS	Mobile Crowd Sensing	4

Abbreviation		Page
SOA	Service Oriented Architectures	4
DoDAF	Department of Defense Architecture Framework	5
SoaML	System Oriented Architecture Modeling Language	6
MOE	Measures of Effectiveness	6
MOP	Measures of Performance	6
FCC	Federal Communication Council	6
PCS	Personal Communication Services	6
PPDs	Personal Protection Devices	6
GNSSs	Global Navigation Satellite Systems	7
M&S	Modeling and Simulation	8
AGI	Analytical Graphics Inc.	8
STK	Systems Tool Kit	8
GIS	Geographic Information System	8
SOA	Service Oriented Architecture	9
GIS	Geo-spatial Information Sciences	10
SIS	Signal in Space	10
NGA	National Geospatial-Intelligence Agency	11
IFR	Instrument Flight Rules	12
WAAS	Wide Area Augmentation System	12
RAIM	Receiver Autonomous Integrity Monitoring	12
GNSS	Global Navigation Satellite System	13
TADA	Time Anomaly Detection Applique	14
NMEA	National Marine Electronics Association	14
UK	United Kingdom	14

Abbreviation		Page
SENTINEL	SErvices Needing Trust in Navigation, Electronics, Location, and Timing	14
IoT	Internet of Things	19
IEEE	Institute of Electrical and Electronics Engineers	20
OGC	Open Geospatial Consortium	20
ITU	International Telecommunication Union	20
COTS	Commercial Off the Shelf	21
3G	3rd Generation	22
4G	4th Generation	22
S^2aaS	Sensing as a Service	22
ESB	Enterprise Service Bus	24
MOM	Message Oriented Middleware	24
SoS	System of Systems	24
CIM	Computation Independent Model	24
PIM	Platform Independent Model	24
PSM	Platform Specific Model	24
DES	Discrete Event Simulation	25
ABEP	Architecture Based Evaluation Process	28
OSM	OpenStreetMap	29
SoaML	Service Oriented Architecture Modeling Language	34
WSDL	Web Service Description Language	38
XML	EXtensible Markup Language	38
CIO	Chief Information Officer	39
DES	Discrete Event Simulation	42

Abbreviation		Page
ABM	Agent Based Modeling	42
SOAP	Satellite Orbit Analysis Program	43
SEAS	System Effectiveness Analysis Simulation	43
AFSMC	Air Force Space and Missile Command	43
RF	Radio Frequency	44
JSPoC	Joint Space Operations Center	44
OOSD	Object Oriented Software Development	46
WGS 1984	World Geodetic System 1984	55
SQL	Structured Query Language	56
JCA	Joint Capability Area	62
UML	Unified Modeling Language	67
RDF	Resource Description Framework	103
LARIAC	Los Angeles Region Imagery Acquisition Consortium	104
GIANT	GPS Interference and Navigation Tool	107
CONUS	Continental United States	108

A SERVICE ORIENTED ARCHITECTURE APPROACH FOR
GLOBAL POSITIONING SYSTEM QUALITY OF SERVICE MONITORING

I. Introduction

1.1 Purpose

According to the National Space-Based Position, Navigation, and Timing (PNT) Advisory Board to the National Executive Committee there were an estimated one billion devices that rely or utilize the Global Positioning System (GPS) capability. In their report the advisory board highlights the dependency of the United States (U.S.) on the GPS infrastructure and cites dependencies ranging from cell phone towers to the Federal Aviation Administration (FAA) NextGen Air Traffic Control System. The report also advocates, due to the number of GPS PNT dependent infrastructures and the critical nature of these infrastructures to the United States that the GPS service be declared a part of the U.S. abbreviationFull[Critical Infrastructure]CI [12].

"Critical Infrastructure," as defined by the Department of Homeland Security (DHS) public website, is:

> "assets, systems, and networks, whether physical or virtual, so vital to the United States that their incapacitation or destruction would have a debilitating effect on security, national economic security, national public health or safety, or any combination thereof." [13]

At a congressional educational event on GPS modernization in June 2015 Caitlin Durkovich, Assistant Secretary DHS, highlighted the dependency of our current CI on the GPS service. She also acknowledged the need to strengthen the protections around the GPS capability to preserve our nations CI.

Given the importance of GPS to our national security, it is imperative that the U.S. Air Force (USAF) develop a robust, flexible, real time, and widespread monitoring capability for the Quality of Service (QoS) of the GPS Signal in Space (SIS). The focus of this thesis is to develop a system architecture based on the principles of Mobile Crowd Sensing (MCS) for monitoring the GPS QoS in an Area of Interest (AoI).

1.2 Background

The GPS system achieved Initial Operational Capability (IOC) in 1993. The initial need for the GPS capability was born out of the inherent inaccuracy of the inertial guidance systems that were installed in Navy ballistic missile submarines. In the 1980's microchip technology had evolved to a point that allowed for the commercial development of GPS chip-sets. During the first Gulf War GPS technology was tightly integrated into both land and air military operations. This public demonstration of the success of GPS provided a large push for continued development of GPS enabled systems. The GPS program has continued to develop and integrate new capabilities into its service. Today the Department of Defense (DoD) relies on the GPS infrastructure for missions ranging from Precision Guided Munition (PGM) strikes to time tagging Intelligence, Surveillance, and Reconnaissance (ISR) data [14].

In conjunction with the U.S. military dependency on the GPS service there are countless civilian applications around the globe that rely on GPS PNT information. Civilian use of the GPS service was first made possible under President Ronald Reagan. In 1983 President Reagan announced that GPS would be made available to the public at no additional cost, but the service would not be as accurate as that provided to the military. This active degradation was known as Selective Availability (SA). However, in 2000 President Bill Clinton directed that the civilian and military signal accuracy be made equivalent and SA be removed from future GPS systems. With the SA intentional degradation disabled, civilian GPS accuracy was improved ten fold, and is now a part of daily life for users around the world [15]. Today the GPS service is specified to provide accuracies of better than thirteen meters in the horizontal plane and twenty-two in the vertical plane [16]

GPS PNT capability is made possible by the use of the concept of Time of Arrival (ToA). This concept centers around measurement of the time it takes a signal to propagate from a known location, or the case of GPS, several known locations to a single received point [16]. GPS capability makes us of the TOA information from the signal transmitted from a constellation of 24 satellites in Medium Earth Orbit (MEO). A GPS receiver must receive the GPS Signal in Space (SIS) from a minimum of four satellites in order to perform the 3D multilateration calculation to determine its position. While most people are familiar with the basics of GPS it is worth noting that the receivers do not communicate back to the GPS space element. This is especially relevant for this research as there is no direct feedback from a receiver on how well the information used to process the TOA data was received. The DoD monitors the Precise Position Service (PPS), which is used primarily for military operations, via Operational Control Segments (OCSs) around the globe. The primary difference between the Standar Positioning Service (SPS) and PPS signals is that the PPS utilizes two separate frequencies which allows for the correction of effects due to the earth's atmosphere [15]. Officially the DoD does not have a means to monitor or assess directly the performance of the SPS [1].

The GPS program is currently managed by the USAF Space and Missile Center GPS program office (SMC/GP). SMC/GP is the sponsoring organization for this research. This research is intended to support SMC/GP's desire to develop a new and novel architecture for monitoring the GPS SPS in near real time around the globe. SMC/GP's interest in GPS monitoring aligns with the 2010 National Space-Based PNT Advisory Board report which included a similar recommendation for development of a national GPS interference detection architecture:

> "The NATIONAL EXECUTIVE COMMITTEE should establish and sponsor a GPS Interference Locating, Reporting, and Elimination System;... No such National (or International) Real-Time System exists today or is even currently planned [17]"

Clearly there is a need for monitoring of the GPS SPS. SMC/GP's direction for this research is to explore development of a monitoring architecture that utilizes sensors of opportunity that are already fielded to reduce development and fielding cost that traditionally accompany acquisition of new systems. In order to meet these objectives this research will focus application of a concept known as Mobile Crowd Sensing (MCS) to monitor the GPS SPS.

MCS is an emerging phenomenon implemented in many industries for monitoring sets users' behaviors, decisions, and status at any given time through various open source media. This monitoring technique is made possible by the continued growth of wireless networks, social media, and capabilities of mobile phone technology. MCS is a concept closely tied in with the modern data rich concepts of cloud computing and Service Oriented Architectures (SOA). Both of these architecture concepts utilize QoS as a key metric for measuring its effectiveness. A MCS cloud computing and/or SOAs can accommodate both passive and active users [18]. Additionally, due to the global nature of mobile phone technology using mobile phone users as inputs to the MCS architecture provides the opportunity for wide spread localized monitoring of the GPS SPS signal which aligns with the SMC/GP objectives.

SMC/GP has also expressed a desire for the GPS SPS monitoring architecture to be capable of identifying trends that indicate development and deployment of threats to GPS operations in an AoR. As GPS has evolved from a U.S. military capability to a globally accessed service so too have the systems used to disrupt and deny it. GPS threats are divided into two primary categories Radio Frequency (RF) jamming and spoofing. RF jamming works on the principle of overpowering the weaker GPS SIS with a higher power noise signal in the GPS frequency band of 1575.42 MHz, or the L1 frequency band. This type of jamming prevents the receiver on the ground from detecting the SIS from any satellites, and thus it cannot determine its position or the GPS timing data. Potentially more destructive, GPS spoofing is more sophisticated than just blocking the GPS SIS. GPS spoofing jammers interrupt the receiver's ability to detect the GPS signal briefly. Then instead of allowing the receiver to reacquire the GPS signal, it transmits a false signal that tricks the receiver into using the artificial signal from the jammer to determine the PNT information [2].

The lack of basic monitoring capability coupled with the ease of availability of threats to the GPS capability requires a novel architectural solution that is both technologically feasible and politically palatable. The application of utilizing commercial GPS receivers as sensor of opportunity for inputs into a MCS architecture is one possible solution to this capability gap and is further explored in this research. The following section details the research objective of this research. This is followed by key research questions that help further refine the topic areas this research will explore.

1.3 Research Objective

The focus of this research is to develop and simulate a MCS architecture capable of monitoring, in near real-time, the received quality of the SPS GPS SIS on a global scale utilizing receivers of opportunity in an AoR and apply predictive analytics to anticipate threats to the GPS QoS. The high level system architecture will be developed using the Department of Defense Architecture Framework (DoDAF) and supplemented with the System Oriented Architecture Modeling Language (SoaML). To assess the potential effectiveness of the selected architecture a model of the monitoring architecture will be developed and simulated. Development of Measures of Effectiveness (MOE) and Measures of Performance (MOP) will be critical for evaluating the candidate architecture, and will be developed with the aid of the research sponsor.

1.4 Research Questions

1. What current methods exist to detect jamming of the GPS SPS?
2. What are the key factors to consider when developing a MCS architecture?
3. What are the most applicable tools and techniques for modeling and simulating a MCS architecture?
4. How effective is a MCS architecture at identifying the source of a single source of GPS interference?

1.5 Justification

As highlighted above, the lack of monitoring capability and ease of access to GPS jamming technology pose a threat to the GPS SPS and the global user base that is dependent upon the PNT information it provides. According to the Federal Communication Council (FCC):

> Federal law prohibits the operation, marketing, or sale of any type of jamming equipment, including devices that interfere with cellular and Personal Communication Services (PCS), police radar, Global Positioning Systems (GPS), and wireless networking services (Wi-Fi) [19]

However, GPS jammers, or Personal Protection Devices (PPDs), can be acquired for as little as $20. These jammers are becoming increasingly sophisticated. These systems have the ability to disable not only GPS frequencies, but other Global Navigation Satellite Systems (GNSSs) and cell phone frequencies [2].

One example of these types of jammers being implemented occurred in 2013 when a New Jersey truck driver, in an attempt to block the GPS tracking device his employer required him to carry, began operating a jammer out of his vehicle. The jamming was eventually detected, reported, and the driver fined $32,000 due to the interference his jammer caused as he drove by the Newark Liberty International Airport. The jamming signal was identified when an Air Traffic Controller at the Newark Airport experienced continued interference and issues with the airport's new GPS aided navigation system. The resulting report eventually prompted a joint investigation between the FAA and FCC. After six months of investigation the source of the jamming was identified, and the driver charged [2][17].

Earlier detection, more consistent monitoring, and efficient data processing could have reduced the six months of investigation time trying to locate the source of the jamming. This is only one example of how GPS jamming threatens a system that relies on the GPS capability. According to a study conducted by Chronos Technology Limited in London England there was an average of fifty jamming events per day from February to December of 2013 [2]. If the volume of jamming events for this major city is to be considered typical then the current identification and prosecution process for GPS law breakers must be augmented to become more efficient.

1.6 Scope

To better focus this research the MCS architectures under investigation will be constrained to capability analysis, functional decomposition, and service based architecture products, or views. This research does not attempt to arrive at the actual technical solution for implementation of the architecture or the development of jamming detection technology. This research assumes that the sensors supplying inputs to the monitoring architecture are utilizing modern open source interfaces, and can provide data as required to the MCS architecture for processing. This research also assumes that the GPS SPS is not able to be augmented, and will only be able to provide the standard level of signal quality for PNT operations.

The MCS architecture investigated in this research is intended to consider monitoring of the SPS, and does not consider the military PPS. To understand the effects on the MCS architecture from including military receivers the additional capabilities afforded to military receivers would need to be included in the architecture and simulation. The MCS monitoring solution could be applicable to military units operating with civilian based GPS systems in an AoR, but would not provide a holistic analysis of actual military monitoring capability without inclusion of appropriate military systems and capabilities.

1.7 Methodology Preview

In support of the SME/GP objectives, the methodology section of this document begins by explaining the development process of the select DoDAF/SoaML products. The architecture development focuses on decomposition of the capabilities, operational activities, service interfaces, and functions for the MCS architecture. The architecture products are developed in Sparx Systems Enterprise Architect 10.0. The section continues with a description of how the Modeling and Simulation (M&S) of the MCS architecture was executed. This section provides details regarding how the Analytical Graphics Inc. (AGI) Systems Tool Kit (STK) was driven by external scripts developed in the computer programming language Python to create the MCS model, and then how a combination of python and a Geographic Information System (GIS) tool known as ArcGIS by ESRI was used to process the simulation outputs.

Finally, the section concludes with how the MOE were extracted from the ArcGIS program.

1.8 Thesis Overview

This thesis is comprised of five chapters; the Introduction, Literature Review, Methodology, Data Discussion, and Conclusion/Results. The literature review section provides additional information on the current GPS monitoring capability, and it explores the current state of research in both MCS and Service Oriented Architecture (SOA). The methodology section describes the process used for developing the monitoring architecture, a description of MCS architecture products, and an overview of the M&S activities used to evaluate the effectiveness of the MCS architecture. The analysis section provides insight into the effectiveness of how well the proposed MCS architecture performs the monitoring capability, and includes information on which MOPs are critical factors of the architecture for achieving the MOEs identified by the SMC/GP office.

II. Literature Review

2.1 Overview

The following section provides insight into the current GPS architecture, and a review of current research in the area of GPS QoS monitoring. Additionally the topics of service based system architecture, MCS, and Geo-spatial Information Sciences (GIS) are explored in this section. The information in this section was used as the basis for development and analysis of the GPS QoS MCS architecture proposed as a part of this research effort.

2.2 GPS Monitoring Capabilities

The current GPS SPS performance standard maintained by the DoD GPS directorate is the official source document for describing SPS "broadcast signal parameters and GPS constellation design." The GPS SPS performance specification defines the SPS as:

> The SPS is a positioning and timing service provided by way of ranging signals broadcast at the GPS L1 frequency. The L1 frequency, transmitted by all satellites, contains a course/acquisition (C/A) code ranging signal, with a navigation data message, that is available for peaceful civil commercial, and scientific use. [1]

The GPS satellites that provide the SPS service are controlled via the Master Control Station located at Schriever Air Force Base, Colorado. The Master Control Station is responsible for ensuring that the GPS transmission, known as the GPS Signal in Space (SIS), is broadcasting in accordance with the GPS system performance specification. Additionally, the MCS performs any satellite routine maintenance processes such as software maintenance. The Master Control Station is supported by a network of globally distributed monitoring sites. When the monitoring sites are used in conjunction with the National Geospatial-Intelligence Agency (NGA) GPS monitoring stations 100% global monitoring of the GPS SIS is possible [1].

At first glance it would appear that the GPS SPS performance is well managed and monitored. However, 100% global monitoring is not as inclusive as one might think. Below in Figure 1 is an image of the approximate coverage area for a single GPS satellite.

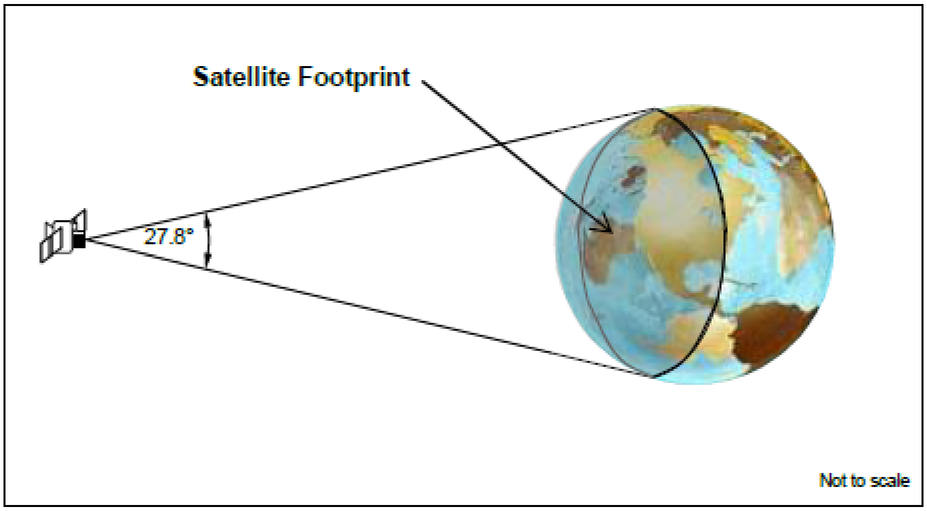


Figure 1. GPS Satellite Footprint [1]

As depicted in the Figure 1 each satellite in the 24 slot constellation provides roughly 38% coverage of the Earth's surface [1]. This equates to 74 million square miles of coverage. Quickly it becomes apparent that the current monitoring architecture is not capable of providing insight into the GPS QoS across such vast distances. The current monitoring system is used to insure that if uninterrupted across the 74 million square miles that the SIS is reaching the intended area in accordance with the SPS.

There has been a concerted effort by certain industries, in particular the FAA, to enhance the GPS SPS monitoring capability. Since 1993 the FAA has monitored the GPS SPS performance data in order to determine GPS viability for Instrument Flight Rules (IFR) operations. The system approved by the FAA is known as the Wide Area Augmentation System, or WAAS. WAAS is a network of twenty eight precisely surveyed locations that provide GPS correction and augmentation data for more precise navigation during aircraft takeoff and landing. The FAA GPS product team performs a quarterly analysis report that documents the GPS performance as collected by the WAAS reference stations [20]. These reports uses a 24 hour position accuracy value based on a one second sample time interval. These sample sets of data ”give a relative idea of constellation health for both the current and combined history of past quarters” [20]. The information provided by WAAS is intended to remove any anomalies presented by the SIS due to spacecraft error or natural interference phenomenon during transmission. WAAS is not intended to detect jamming events nor provide any feedback to a monitoring station as to a potential source of any interference. WAAS does have the capability to broadcast correction data but the data recorded by WAAS focuses on reliability of the GPS SIS not identification of jamming occurrences [20]. In fact according the WAAS performance analysis report from the third quart of 2013 the WAAS system reported no availability outages during the third quarter for Newark Liberty International Airport, New Jersey even though there was a recorded jamming event that was disrupting airport activities [21][22]. Ultimately WAAS provides a pivotal service to the FAA, but it is not a viable solution on its own to monitor the GPS QoS.

In addition to the WAAS, the FAA utilizes Receiver Autonomous Integrity Monitoring (RAIM) to enhance their GPS capability. RAIM is a GPS signal processing technique that uses a minimum of five GPS satellite signals to compute a set of PNT solutions. The receiver then compares the set of navigation solutions and is able to identify and reject any faulty GPS signal data [23]. Included as a part of the quarterly GPS SPS performance report to the FAA is a RAIM coverage and performance analysis. The RAIM receivers are collocated with the WAAS sites, and the FAA utilizes RAIM enabled GPS receivers on board aircraft to notify the pilot if the GPS system is unreliable [20]. However, RAIM receivers are very specific to the aviation community and are not deployed with enough volume to provide an actionable service to consumers. Additionally RAIM focuses on GPS SIS reliability and not identifying the root cause of the error.

At their core these current monitoring capabilities are all focused on understanding the GPS SPS under normal operations. Unfortunately these systems are not designed to differentiate between a jamming attack and a malfunction of the GPS satellite or distortion of the signal due to atmospheric effects. Additionally, OCS, NGA, and WAAS monitoring systems are limited to the precisely surveyed locations. It is apparent that a dedicated, more mobile, and sustainable detection architecture is required to enhance the current GPS monitoring capability.

2.3 GPS Jamming and Detection

The capability to detect and identify GPS jammers is not a completely unexplored area of research. This section focuses on two industry leaders in PNT technology that have been focused on in the area of GPS jamming event detection and mitigation.

The first of these industry leaders is The MITRE Corporation. The MITRE Corporation, henceforth referred to as MITRE, is a federally funded research and development center that assist the United States government in the areas of science and technology research and development. MITRE has a dedicated PNT research focus which provides support directly to SMC/GP directorate. One of the technologies highlighted in discussions with members from the MITRE PNT team was a smart phone application that could perform Global Navigation Satellite System (GNSS) jamming and spoofing detection directly on the host device. This application is known as the Time Anomaly Detection Applique (TADA). TADA utilizes the native android operating system location service interfaces to observe the GPS National Marine Electronics Association (NMEA) formatted data. TADA is used to provide feedback to a command and control node when potential jamming is detected, and it provides the user with situation awareness of other potential jamming events in the area[24]. MITRE has demonstrated that the TADA application's ability to detect GPS jamming entities both in the laboratory and during field testing. Additional information regarding the TADA application is available through request to MITRE. MITRE's research and development of the TADA application proves that it is possible to utilize commercial smart phone technology as a sensor for GPS quality. This validates a large assumption critical to this research that smart phones could act as inputs to the GPS MCS architecture.

Another industry leader in GPS jamming detection and identification is Chronos Technologies. Chronos Technologies, founded in 1986, is a European based company whose focus is in timing and monitoring systems for network systems [25]. Starting in 2008, at the request of the United Kingdom (UK) Technology Strategy Board, Chronos Technologies launched a program known as the SErvices Needing Trust in Navigation, Electronics, Location, and Timing (SENTINEL). The main goal for the SENTINEL program was to develop a "national network of GPS interference and jamming sensors" [2]. The SENTINEL program utilized GPS receivers augmented with jamming detection at key mission and safety critical sites that relayed GPS jamming event detection to a server at Chronos Technology. The SENTINEL program was able to gather months of data at various sites around the UK. For the month of February 2013 a single sensor from Chronos Technology detected over 100 jamming events within the city of London. Figure 2 is taken from the SENTINEL project final report showing the cumulative number of jamming events per day of the week for the month of October 2012 to December 2013.

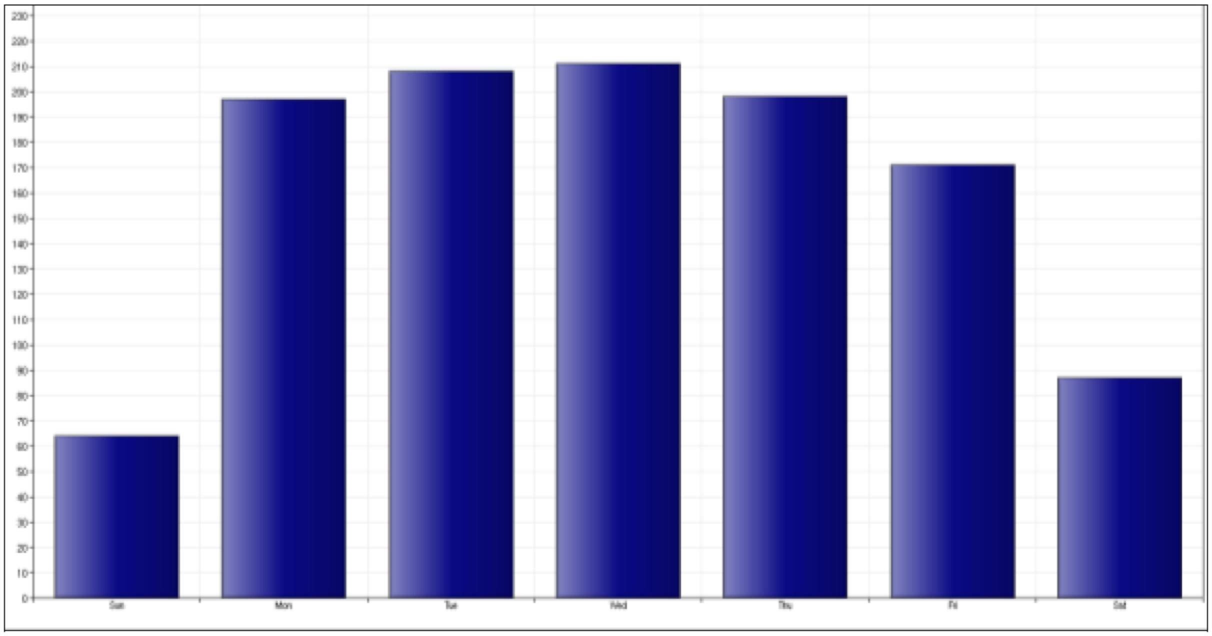


Figure 2. Total jamming events per day from Oct 2013 to Dec 2015 [2]

Detection data from the SENTINEL program was also used by Chronos Technology and law enforcement agencies to identify and apprehend a GPS jamming perpetrator in as little as three weeks, which is a stark contrast to the six month Newark airport jamming investigation. The ability of the SENTINEL program to close the loop from detection to elimination of a jamming threat with such efficiency serves as proof of concept that distributed sensors are a viable method for GPS jamming monitoring and defeat [2].

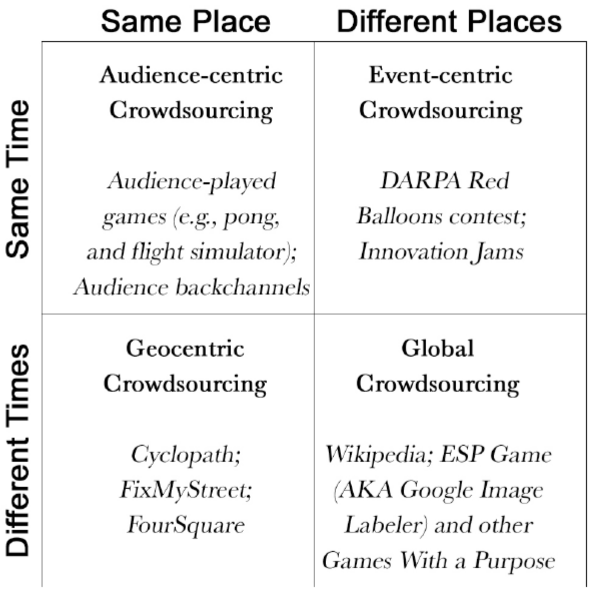
It is clear that monitoring the QoS of the GPS SPS is possible via remote systems. And that actionable intelligence can be produced by the system for the removal of a GPS interference source. However, to date the systems used for detection of GPS jammers are still limited to pre-placed locations. This limiting factor makes it hard to proactively look for sources of GPS interference and to locate the source of the GPS jamming or spoofing threats once the threat has left the area of interest. Additionally, if the jamming detection device locations become known, then the perpetrator can simply avoid them.

2.4 Mobile Crowd Sensing

One possible solution to the issue of non-mobile detection devices utilized in the SENTINEL program is to expound upon the MITRE TADA application through the use of MCS. MCS is a capability that has been made possible by the pervasiveness of small mobile devices and powerful networks capable of passing gigabytes of data between users. MCS is essentially the utilization of mobile devices to perform crowdsourcing. In a 2011 workshop on crowdsourcing and human computation Thomas Erickson of the IBM T.J. Watson Research Center released a paper defining crowdsourcing as "Tapping the perceptual, cognitive, or enactive abilities of many people to achieve a well defined result such as solving a problem, classifying a data set, or producing a decision." The IBM team also explored the concept of crowdsourcing across four space and time analysis domain as depicted in Figure 3 below.

Application of this crowdsourcing model is useful to a GPS MCS architecture by highlighting the spatial and temporal domains in which the MCS architecture might operate in. In particular this research focuses on the Geocentric Crowdsourcing and Audience-centric Crowdsourcing since the GPS MCS architecture is focused on near real time monitoring of the GPS QoS for a set of users in a defined AoI [3].

MCS can be classified into two categories: 1) Participatory Sensing and 2) Opportunistic Sensing [26][27]. Participatory sensing occurs when a mobile user is an active member of the MCS and is knowingly providing data for processing [26]. Khan *et al* present an example of a participatory MCS program known as *PEIR*. *PEIR* utilizes GPS location data, traffic data, weather, and other data to monitor a user's



	Same Place	Different Places
Same Time	**Audience-centric Crowdsourcing** *Audience-played games (e.g., pong, and flight simulator); Audience backchannels*	**Event-centric Crowdsourcing** *DARPA Red Balloons contest; Innovation Jams*
Different Times	**Geocentric Crowdsourcing** *Cyclopath; FixMyStreet; FourSquare*	**Global Crowdsourcing** *Wikipedia; ESP Game (AKA Google Image Labeler) and other Games With a Purpose*

Figure 3. Four quadrant crowdsourcing model [3]

environmental footprint. Users then can share and compare their impact data among a network of other users. The active logging of the data over time can be used to spot trends in a *PEIR* user's habits [27]. Conversely, opportunistic sensing puts the burden of submitting sensor data onto the mobile device. The user is not aware that the data are being collected and data are automatically uploaded from the mobile user to a network for processing. Khan *et al* also present an example of an opportunistic sensing architecture known as the *Activity Recognition System*. This system automatically identifies user's activities, such as biking, driving, or walking, based on accelerometer and GPS data from the mobile user's cell phone. The designers of the *Activity Recognition System* can then use the data to generate activity profiles over time for a set of mobile user to inform on most utilized modes of transportation[27].

MCS can also be categorized via the scope of the sensing architecture. This "sensing scale" can be considered to have three different levels personal sensing, group sensing, and community sensing [26]. For the purposes of this research, community sensing is the most appropriate. Community sensing consists of a large number of independent users providing data for the good of the whole community. Often times Community sensing requires users to be willing to connect to other unknown users or groups of users. This blind sharing creates a set of unique challenges due to a user's desire to protect personnel data, such as their immediate location. To overcome this challenge, a set of opportunistic sensing users could provide either less data or data less often with a select number of participatory sensors providing a full set of required data could provide the required GPS monitoring capability, and could reduce the risk and impact on a mobile user thus incentivizing more users to participate in the service[26].

The GPS PPS is not the focus of this architecture development, but it is included here to highlight some of the considerations that would be required to implement an MCS architecture for the PPS. The PPS is the primary source for PNT data for DoD military and select government agencies. The PPS is accessed through the use of controlled cryptographic features [16]. Additionally, during military operations there are GPS signal augmentation capabilities for increased QoS that are not available to the everyday GPS SPS users. Regarding a MCS architecture for monitoring during military operation the "sensing scale" can be considered at the group level because each user is considered to have the same goal or mission. In a group sensing architecture there is an element of trust between users that facilitates faster collection and processing of data [26]. Additionally, there are trusted military networks and electronic warfare systems that the military employs during operations that could enhance the MCS architecture. However, during military operations there might not be as many mobile sensor or users from which to monitor the GPS QoS. The concepts presented in the Lane article are still applicable to potential military MCS operations, but might require a hybrid approach with commercial receivers of opportunity in or around the military AoI. The MCS architecture developed as a part of this research could be utilized during military operation. Additional research would be required for the development of a GPS QoS capability during military operations that considers the GPS PPS capabilities, number of mobile military users, and military electronic warfare tactics.

2.5 Service Oriented Architecture

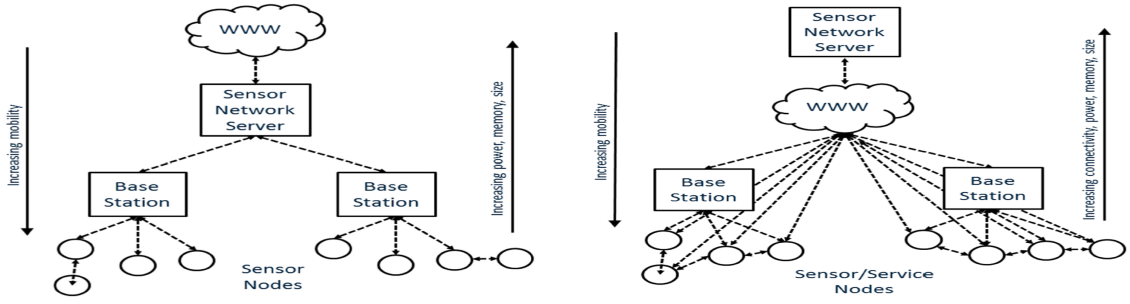
One approach to a MCS GPS monitoring solution is to use a SOA. The use of a SOA supports both opportunistic and participatory MCS solutions. The term SOA applies primarily to the design of evolutionary and flexible software systems based on encapsulation and low coupling . The goal of a SOA is to provide a set of abstract interfaces between systems, via software, that allows an end user's product to be developed independently of platform and lower levels of programming [14, p. 167]. Another definition provided by Kotsev *et al* of a SOA is:

> an environment where loosely-coupled network resources are made available as independent services, which can be accessed without preliminary knowledge of their underlying implementation platform and exist autonomously yet not isolated from each other. [4]

SOAs play a key role today in software development to enable connecting, organizing, and transferring data between devices considered to be a part of the Internet of Things (IoT). IoT devices are considered to be a worldwide network of objects that are uniquely identifiable and reachable based on standard communication protocols. In the journal article by Kotsev *et al* titled "Architecture of a Service-Enabled Sensing Platform for the Environment" the authors highlight a set of standards for enabling a distributed network of mobile devices, a service oriented sensing architecture for the environment based on these standards, and a review and evaluation of potential hardware and software solutions for implementation of the proposed SOA.

First, the authors identify the ”interdependent factors” that restrict the ability of IoT and distributed devices from being able to network and operate together. The authors highlight the lack of interoperability between sensor components which results in non-standard calibration of data from sensor to sensor is what prevents data utilization. Additionally, formatting of data is often proprietary for different environmental sensor devices which present a challenge of processing and combing data from different sensing sources.

To overcome these barriers the authors identify standards set by the Institute of Electrical and Electronics Engineers (IEEE), Open Geospatial Consortium (OGC), and the International Telecommunication Union (ITU). The authors apply these standards and guidelines to the development of a SOA that focuses on dynamic allocation of sensor nodes; with each node in the architecture possessing the ability of on-board discovery each node can essentially be thought of as an independent actor capable of being assigned to multiple networks at a time. Figure 4(a) and Figure 4(b) below show the difference of the proposed environmental sensing architecture from the traditional sensor approach.



(a) Traditional Environmental Sensing

(b) Distributed SOA Environmental Sensing

Figure 4. Environmental Sensing Architecture Comparison [4]

This proposed architecture mitigates the effects that occur due to the heterogeneous nature of the environmental sensing devices, and provides more flexible data routing capability. Hosting web interfaces directly on the sensor devices facilitates linking the devices into new networks without the need to coordinate knowledge regarding the hardware or lower level protocols of each device. The use of these open standards and dynamic network configurations creates a conducive environment for data gathering and processing.

The next section of the article details the proof of concept experiment the authors conducted to design and test a prototype environmental sensor from low cost Commercial Off the Shelf (COTS) hardware and open source software. The prototype utilized the higher level web service protocols identified earlier in the article to expose data to network participants. The authors were able to demonstrate that "multiple accesses to data without preliminary knowledge of the actual technology..." was possible [4].

The results of the Kotsev *et al* research demonstrate that it is possible to utilize low cost, network flexible, and interoperable sensors for environmental sensing. These attributes are considered highly desirable for large scale opportunistic MCS. Additionally, the proof of concept experiment demonstrates that dedicated participatory GPS sensors could be made interoperable with the opportunistic GPS sensors through the use of internationally recognized web service open standards. The use of a web service interface for devices in the MCS network also allows for interoperability without requiring a change at the lower levels of the already established IoT devices.

Mobile phones and tablet computers are the predominant members of the IoT. The continual growth of capability in these devices makes them excellent candidates for being considered the primary sensors in a GPS MCS architecture. One way to consider the use of mobile devices for sensing the GPS QoS, or environment in general, is to consider the sensing capability as a cloud computing sensor service request. Mobile phones are equipped with an array of embedded sensors, and are capable of connecting to external sensors via Bluetooth and other personal area network protocols. They are also intrinsically linked to a network, often via 3rd Generation (3G), 4th Generation (4G), or Wi-Fi. These mobile, networked, and process capable devices provide the primary inputs and requests in a Sensing as a Service (S^{2}aaS) architecture as described in the sheng *et al* article [5].

S^{2}aaS is a proposed method where cloud users or network service providers can initiate a sensing request via a web based interface. This request is then sent to a sensing server where the request is processed and released to the appropriate mobile users in the AoI. These mobile users perform the requested task in either a participatory or opportunistic sensing fashion then provide the data to a database where it can be accessed and processed by the requesting user.

Figure 5 depicts the S^{2}aaS cloud service architecture. A S^{2}aaS cloud computing service is dependent on the mobile users sacrificing their own battery and computing power to perform the sensing task and rudimentary processing before forwarding a response to the server. As identified above there are certain risks to the user associated with a community sensing architecture; particularly the risk of a security or privacy breach. There are four primary capabilities that a S^{2}aaS should support in order to be considered an effective system [5].

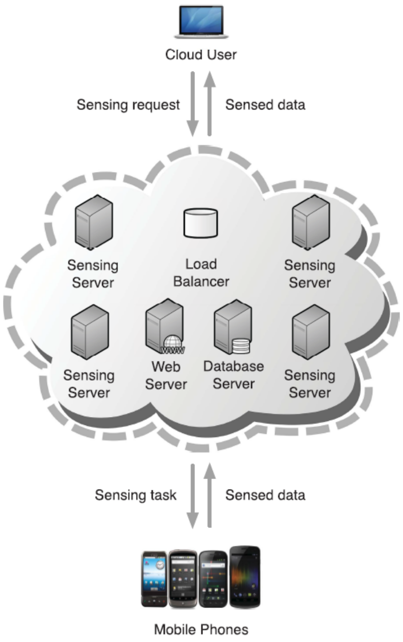


Figure 5. S^{2}aaS Cloud Architecture [5]

1. The system must be general enough to be implemented on a large subset of opportunistic and participatory sensors. Examples of these systems include mobile phones that use Android or iOS operating systems.

2. The system must be flexible and be re-configurable remotely. The ability to provide remote updates allows for more efficient algorithms to be implemented. This also allow for firmware adjustments to be broadcast to the S^2aaS cloud users to preserve interoperability in the event changes occur to the open source interface standards or telecommunications policy.

3. The required power to perform the sensing energy must be minimized as much as possible, and the sensing activity must not detract or prevent normal mobile user operations.

4. The system must have an incentive mechanism to encourage mobile phone and other IoT users to participate in the sensing activities.

In addition to these high level capabilities, an S^{2}aaS architecture should support the following system functions: 1) support a web interface, 2) generate sensing task, 3) track mobile phones and users, 4) support recruitment of mobile users, 5) schedule sensing activities, 6) manage sensors of the mobile users, and 7) process and store the collected data. These functions are directly traceable to the S^{2}aaS capabilities above as is typically done during architecture development for a new system or system of systems [5]. These high level capabilities are reflected in the MCS architecture developed as a part of this research and can be observed in the capability hierarchy found in chapter IV and Appendix A.

It is clear that utilizing mobile phone users as an input to a GPS MCS architecture is possible with today's technology. The Khan *et al* article identifies the state of the art in what is possible for mobile phone sensing, and the Sheng *et al* article highlights some key functionality and system attributes that a MCS should support. Additionally the Sheng *et al* article highlights ongoing research regarding energy efficient mobile sensing for large numbers of users, and possible incentive structures for recruitment of mobile users. This article does not focus on the development of lower level technologies or algorithms for data processing once a sensor task is released into the S^{2}aaS cloud.

2.6 Model Based Systems Engineering

One of the key benefits of a SOA is that it allows a large number of systems to inter-operate through defined interfaces utilizing an Enterprise Service Bus (ESB) and Message Oriented Middleware (MOM). The use of an ESB and MOM is esspecially viable within a federated set of services [7]. But SOA are not limited to exchanging information between known, or federated participants, and capturing and understanding all the interactions of the independent systems presents complex challenges for system developers. In fact as the number of systems increase, and the complexity of the service interactions increases the interfaces between systems should be strengthened and the coordination streamlined to facilitate optimized service capability [6]. MBSE can be used to understand these complex interactions. MBSE is used to maintain system integrity during design and for exploring performance prediction based on simulated changes to architectural elements [14, pg 222]. The Hu *et al* article provides a meta-model that combines the key components large System of Systems (SoS) should include within the context of an enterprise architecture framework, such as the DoDAF. Using this meta-model to capture architectural concerns Hu *et al* then explore a three tiered hierarchical decomposition of the SOA. These layers are identified as Computation Independent Model (CIM), Platform Independent Model (PIM), and Platform Specific Model (PSM). These aggregated models are comprised of lower level architecture products that guide development of the SOA from a planning and analysis phase, through design, and finally through simulation. Figure 6 depicts the model driven approach utilizing this three layer approach [6] At the lowest level of

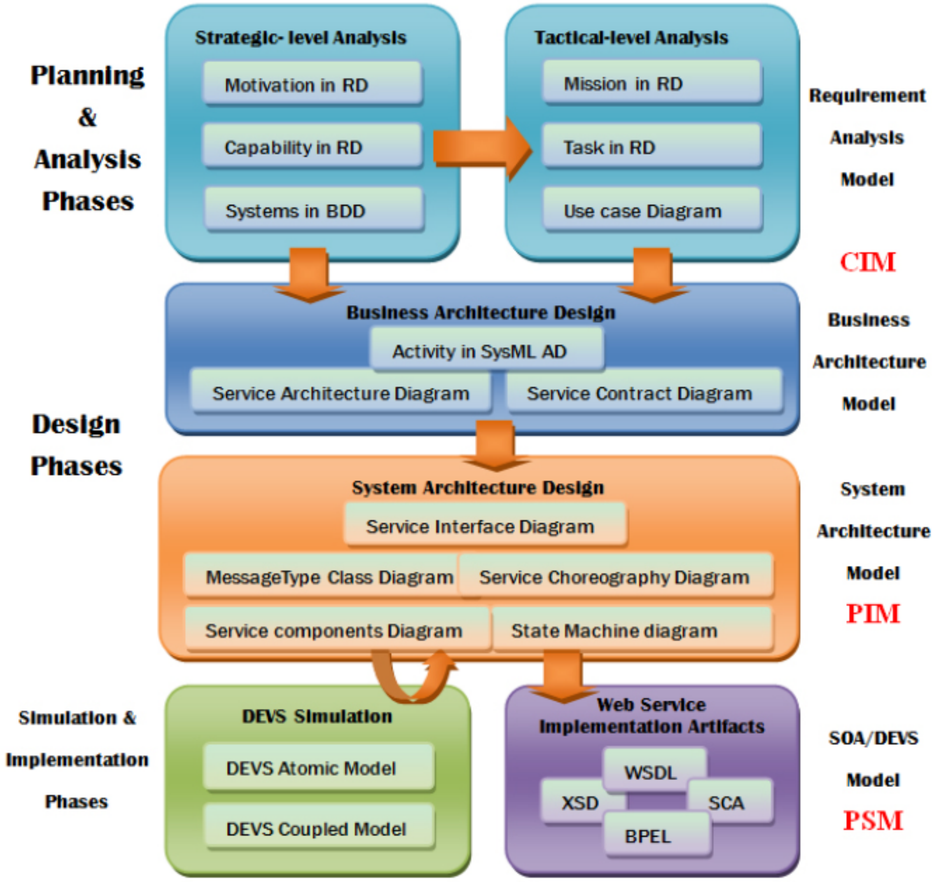

Planning & Analysis Phases
Strategic- level Analysis
Motivation in RD
Capability in RD
Systems in BDD
Tactical-level Analysis
Mission in RD
Task in RD
Use case Diagram
Requirement Analysis Model
CIM
Business Architecture Design
Activity in SysML AD
Service Architecture Diagram
Service Contract Diagram
Business Architecture Model
Design Phases
System Architecture Design
Service Interface Diagram
MessageType Class Diagram
Service Choreography Diagram
Service components Diagram
State Machine diagram
System Architecture Model
PIM
Simulation & Implementation Phases
DEVS Simulation
DEVS Atomic Model
DEVS Coupled Model
Web Service Implementation Artifacts
WSDL
XSD
SCA
BPEL
SOA/DEVS Model
PSM

Figure 6. Model driven architecture development [6]

the proposed SOA framework a Discrete Event Simulation (DES) is implemented that captures the interaction between services described in the CIM and PIM levels. The architecture service component diagrams contain all the necessary interface descriptions to establish links between components within the DES. The DES component states are monitored during the simulation, and trigger events are used to initiate transactions between the components. The DES is used predicatively to evaluate the effectiveness of the proposed SOA, and the interface management allows different system components to be used interchangeably to identify the best SoS configuration [6].

Similarly, Abusharekh, Gloss, and Levis utilize a DES for evaluating a SOA and include three qualifiers for the performance of the SOA: 1) loose coupling, 2) service linkages with supporting applications, and 3) SOA infrastructure specifications [7]. That is to say that a SOA should be comprised of reusable services that interact through an ESB and MOM built upon an established network environment.

Abusharekh *et al* provide a framework for evaluation of SOA in quantifiable terms. The authors explore the development of a DES developed from DoDAF architecture products decomposed from capabilities, to operational activities, to system functions. The Abusharekh architecture evaluation framework is described below in Figure 7.

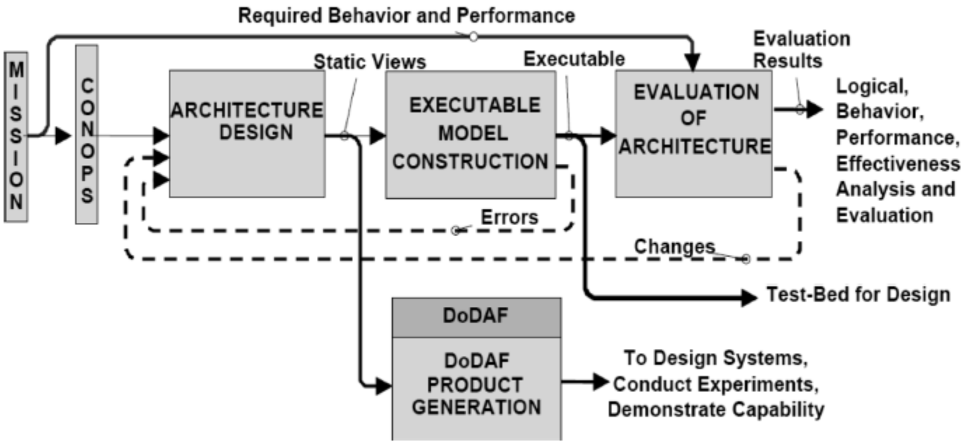

Required Behavior and Performance
MISSION
CONOPS
Static Views
Executable
Evaluation Results
ARCHITECTURE DESIGN
EXECUTABLE MODEL CONSTRUCTION
EVALUATION OF ARCHITECTURE
Logical, Behavior, Performance, Effectiveness Analysis and Evaluation
Errors
Changes
Test-Bed for Design
DoDAF
DoDAF PRODUCT GENERATION
To Design Systems, Conduct Experiments, Demonstrate Capability

Figure 7. Proposed Method for DoDAF Architectures [7]

This method relies on each component of the SOA to have an established operational relationship and an understood set of rules for exchanging message traffic and data. These exchange rules are required for different domains to be considered federated. Once these relationships are established, the evaluation method proposed provides an avenue for comparing arbitrary unions of multiple SOAs. The framework proposed by Abusharekh *et al* can be used to extend the architecture principles in the Hu *et al* article and provides additional specifics for the development of a SOA DES. To clarify the Hu article provides details on the desirable and necessary SOA traits, and the Abusharekh text establishes a methodology for evaluation of these identified traits. For example, the Abusharekh development framework highlights the need for a detailed communication resource profile for evaluating the SOA effectiveness. These details include understanding what devices, protocols, data types, and data volume the SOA will be expected to manage. To this effect the article establishes five service profiles that describes the performance characteristics of the SOA:

1. Network Profile - "... the Physical view of the architecture developed during the Architecture Design Phase [7]"
2. ESB profile - captures the process delays due to the ESB
3. Business Services Profile - processing delays of business services
4. Business Processes Profile - "The main source of this profile is the Functional view of the architecture [7]"
5. Scenario Profile - defines the anticipated inputs into the SOA, and defines how the request load might change during operations.

The Abusharekh *et al* model assumes that a set of static architecture products are available to that describe the business logic within the SOA. Essentially the executable model is used in place of the traditional dynamic DoDAF views to describe and provide analysis on the effectiveness of the proposed architecture. The article concludes with an example scenario that simulates a SOA and a select subset of its business processes. The example utilized the simulation of the architecture to compare two network configurations to execute the same requested business process. The Abusharekh model provides a clear example that modeling and simulation to provide insight into a SOA potential feasibility and performance is plausible and insightful. Particularly relevant to this research the article validates the use of DES as an acceptable method for modeling and analysis for SOA [7]. However, in contrast to the

Abusharekh article this research did not focus on attempting to develop all five profiles highlighted in the above method. Specific network data was not available, and the set of possible devices that could provide inputs to the GPS MCS architecture is anticipated to be a large heterogeneous mix. Chapter V includes additional recommendations to explore development of these other profiles, particularly the network profile.

Similar work to the Abusharekh *et al* research was conducted at AFIT four years earlier in 2006. The research focused on developing a method for architecture effectiveness evaluation. This process was termed the "Architecture Based Evaluation Process (ABEP). The eight step process outlined in the AFIT research highlights the development of a concept of operations and identification of MOEs that are "relevant to the decision/evaluation [33]." The research highlights a minimal set of DoDAF architecture products required to execute an architecture analysis through the use of DES. This research also includes an implementation of the ABEP method within the context of an operational relevant scenario which highlights the benefits of implementing the ABEP. The ABEP is similar to the architecture evaluation process outlined in Abusharekh *et al*, but includes more specifics on which DoDAF views would be required prior to implementing a MBSE evaluation. The ABEP process highlighted in the AFIT research was used to help derive which DoDAF products would be required prior to development of the MCS M&S activities [33]. However, since the AFIT research focuses on development of a system architecture and this research is concerned with development of a SOA a one to one implementation of the suggest DoDAF architecture products is not applicable. This research used the best practices identified in the ABEP and Abusharekh *et al* method to implement a MBSE evaluation approach to this research's MCS architecture.

2.7 GPS and the use of Geographic Information Systems

To this point all the literature has focused on development of a SOA and how to evaluate it. Throughout this research it became apparent that the identifying the potential location of GPS jammers and capturing the state of GPS users in an AoI would require some method of geo-spatial processing. The final portion of this chapter focuses research that focuses on the application of GIS technologies to process GPS data to understand spatial and temporal patterns.

There is no official definition to describe GIS technologies, but all of the researched terms include much of the same themes. Below are two definitions that provided the clearest understanding of what GIS is and how it is used:

1. "A geographic information system (GIS) is a computer-based information system that enables capture, modeling, manipulation, retrieval, analysis and presentation of geographically referenced data." [8]

2. "GIS allows us to view, understand, question, interpret, and visualize our world in ways that reveal relationships, patterns, and trends in the form of maps, globes, reports, and charts." [28]

The basic structure of GIS can be decomposed into groups of subsystems that directly correlate to the capabilities required by a MCS architecture: Data input, data storage and retrieval, data manipulation and analysis, and data output and display [8]. Figure 8 is a flow diagram that describe the interaction of the GIS sub-systems. Each of these sub-systems should be represented in any GIS analysis effort, and was considered during the course of the GPS MCS architecture development.

One predominate question with any MCS architecture is will there be adequate input sensors to provide valid and reliable output information. One of the most predominate crowdsensing GIS projects is known as OpenStreetMap (OSM). OSM is a

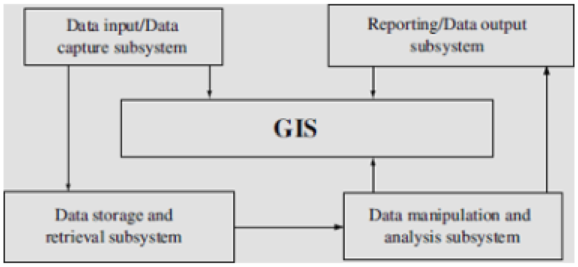

Data input/Data capture subsystem
Reporting/Data output subsystem
GIS
Data storage and retrieval subsystem
Data manipulation and analysis subsystem

Figure 8. GIS architecture and sub-system flow [8]

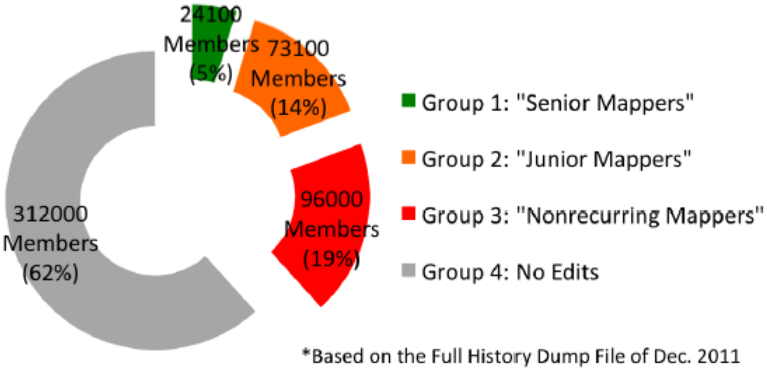


Figure 9. OSM contribution groups [9]

geodata platform where volunteers and professionals gather information and upload it to a central database via a web interface [9]. The Neis article focuses its analysis of the contribution behavior of registered members of the OSM program. The article paints a desperate picture for any architecture expecting contribution from mobile users. Of the 505,000 registered members of OSM approximately 312,000 (62%) members have never contributed to the mapping service. With no data on the activities of the non-participants the article then explores the tendencies of the remaining 193,000 contributors. The article identifies three groups based on number of contributions made by the OSM participant. A senior mapper who contributes over 1000 updates, junior mappers who contribute less than 1000 but more than 10, and nonrecurring mappers who contribute less than 10 updates to the service. Figure 9 depicts the numbers associated with each OSM group. The figure shows that only 5% of the members can be considered Senior mappers and only 14% junior mappers.

A startlingly low number of what can be considered active contributors. The article provides more insight into the geographic correlation of each OSM group and highlights that most of the OSM contributors are located in Europe. But the primary implications of this research indicate that even with a popular service it is unlikely that members continue to be active participants in the service beyond an initial contribution. This implies that the assumption regarding the number of participatory sensor nodes expected to support the GPS MCS architecture might not be a valid one.

However, the OSM consist of a system where participants must not only agree to the service, but also make a concerted effort to provide inputs into the OSM database. Other GIS research suggests that automation of reporting data can result in a viable amount of participants to provide usable outputs to consumers. It is then the responsibility of the sensing service to manage the resources provided by the consumers. A predominate example of this type of data generation and utilization is known as floating car data. Anyone who has ever utilized a GPS mapping service such as Google Maps, or a network enabled dashboard mounted GPS system like TomTom has utilized floating car data. Floating car data are derived from "vehicles equipped with positioning devices; most commonly these are GPS device, which record the movement of the cars and their location in space and time. [29]" Only recently has there been capacity and capability to gather, processes, and analyze large sets of floating car data. Liu and Ban utilized a set of over 85 million taxicab points collected in Wuhan, Hubei, China to explore the spatio-temporal clustering patters of vehicles [10]. This is especially applicable to this research due to the time frame of the collection. The study averaged over 14 million samples per day for six days. That equates to roughly a sample every 20-60s. Liu and Ban developed an algorithm to capture the movement patterns of the taxi data to determine where vehicle's average speeds were clustering, i.e. where were they forced to slow down and where could they speed up relative to the anticipated traffic conditions. Figure 10 below shows some of the results produced by the study. The spatio-temporal algorithm identified areas of long wait periods during rush hour in Wuhan. These type of results focus primarily on the clustering of data bounded by road features, but the principle of using surveyed data to identify hot and cold spots demonstrates the potential to apply GIS processing to GPS QoS. The algorithm developed for by Lui and Ban is specific to identifying velocity patterns, but provided statistical insight in to the patters of the vehicles in the AoI by sampling over time. Similar statistical findings would be critical for understanding results from a system that utilized inputs from

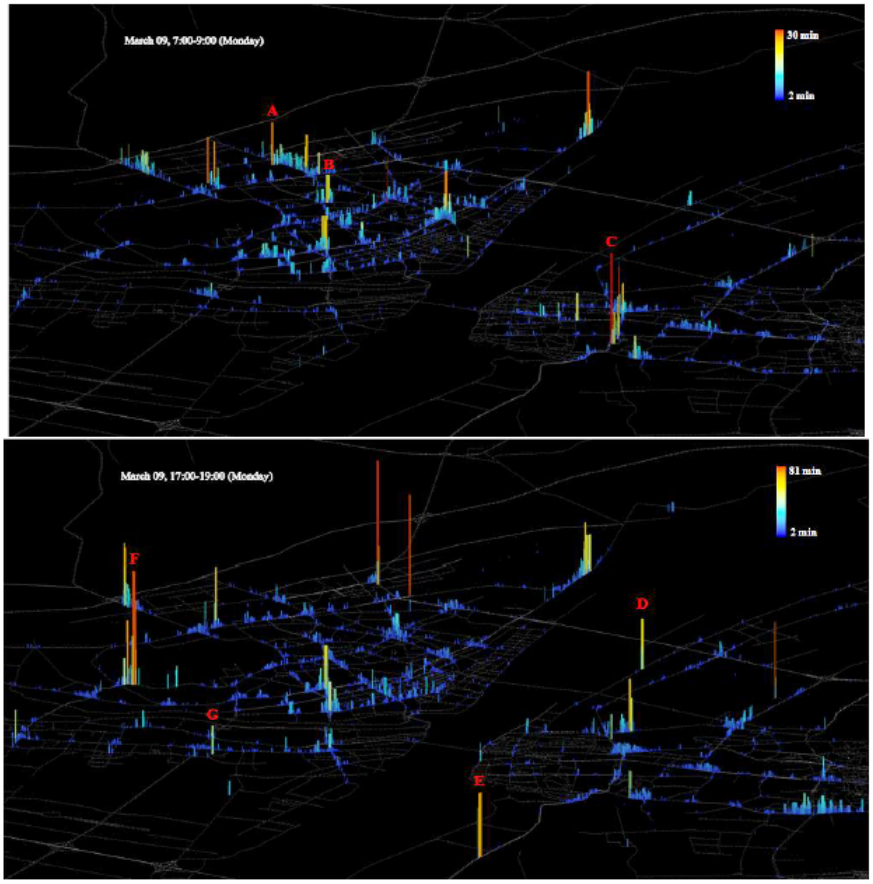


Figure 10. Spatio-temporal weight periods according to their lifetime during rush hour [10]

both road restricted and free roaming sensors.

One of the challenges encountered during spatio-temporal analysis is determining a sample rate of the mobile sensors. Ranacher *et al* provide a recommended sampling schema that attempts to minimize the "effects of error on movement parameters while avoiding the collection of redundant information. [29]" The Ranacher article contends that any floating car data study should consider effects due to measurement and interpolation error. The article highlights the use of a random walk rediscretization approach to compare the actual receiver movement patters to the theoretical random walk model. Ultimately the Ranacher article concludes that the floating car data measurement error is spatially auto-coorelated with the error measurement of the GPS data. The article highlights however that since the error is similar for consecutive sample points the errors tend to cancel out and a 1 Hz sampling rate provides a good approximation of the floating car's movement [29]. To avoid interpolation error for pathing information the article recommends the use of a 1/3 - 1/5 Hz sampling rate. This is a rate that could be supported by a vehicle mounted GPS sensor, but the periodicity of sampling on a battery and processing limited mobile device might not be capable of supporting such rates. Additional investigation would be required to understand the power consumption requirements on mobile device to support this sampling rate, and application of the statistical method capture in Ranacher's article would need to be applied to dismounted users to establish appropriate sampling rates based on the different movement parameters.

III. Methodology

3.1 Overview

The methodology section describes the processes used to develop a GPS MCS architecture. First, an overview of the architecture development approach is provided. This section includes process followed to develop the core DoDAF architecture products and includes a brief description of how the Service Oriented Architecture Modeling Language (SoaML) was used to supplement and enhance certain DoDAF products. Included within this section is a list of the DoDAF and SoaML products developed to describe the GPS MCS architecture. A description of each product and its intended purpose is also included to provide a clear definition of the scope of each product. All of the identified architecture products are included as data items in chapter IV This chapter also includes a description of the M&S activities conducted to analyze a set of MCS architecture variants. The M&S description includes the development process for the architecture simulation, the set of governing assumptions used during the M&S process, and details which parameters were varied across each of the architecture variants for sensitivity analysis. This chapter concludes with a breakdown of the data analysis techniques employed to extract and evaluate each of the MCS architecture variant's MOEs.

3.2 Architecture Development Process

As indicated above the framework selected for developing the MCS architecture was the DoDAF framework. As indicated in Figure 11 below the first step in creating a DoDAF architecture is to identify the stakeholder requirements, objectives, and overall purpose of the architecture. In accordance with this guidance the architecture development approach for this research began with identifying a problem

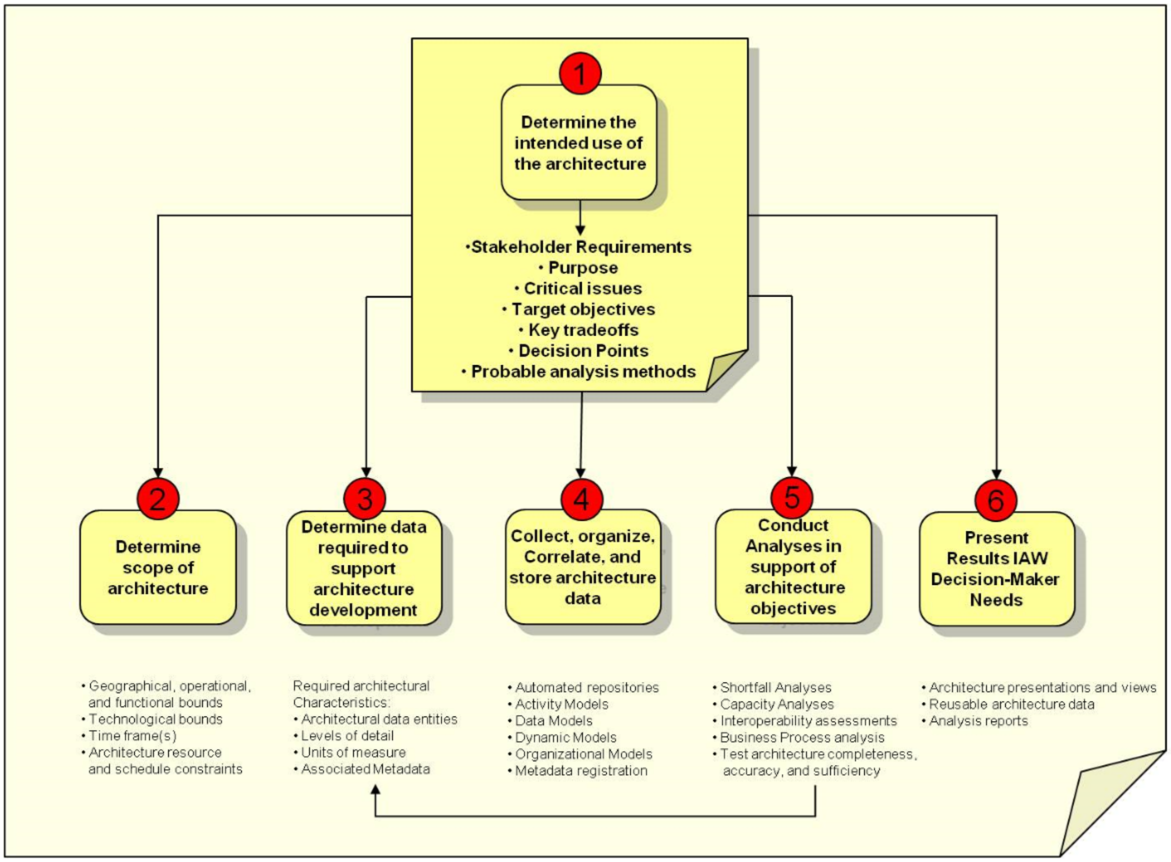

1
Determine the intended use of the architecture
•Stakeholder Requirements
• Purpose
• Critical issues
• Target objectives
• Key tradeoffs
• Decision Points
• Probable analysis methods
2
Determine scope of architecture
3
Determine data required to support architecture development
4
Collect, organize, Correlate, and store architecture data
5
Conduct Analyses in support of architecture objectives
6
Present Results IAW Decision-Maker Needs
• Geographical, operational, and functional bounds
• Technological bounds
• Time frame(s)
• Architecture resource and schedule constraints
Required architectural Characteristics:
• Architectural data entities
• Levels of detail
• Units of measure
• Associated Metadata
• Automated repositories
• Activity Models
• Data Models
• Dynamic Models
• Organizational Models
• Metadata registration
• Shortfall Analyses
• Capacity Analyses
• Interoperability assessments
• Business Process analysis
• Test architecture completeness, accuracy, and sufficiency
• Architecture presentations and views
• Reusable architecture data
• Analysis reports

Figure 11. DoDAF Architecture 6-Step Process

statement. The architecture problem statement is not identical to the research objective statement identified in chapter 1. The research objective statement provides a focus for the overall research effort, but an architecture problem statement focuses on the identification of an operational capability gap presented by the research sponsor. The research objective, in conjunction with discussions with the research sponsor, was used to establish the initial scope of the MCS architecture which addressed the second development step in Figure 11. Figure 12 depicts the DoDAF architecture development process that

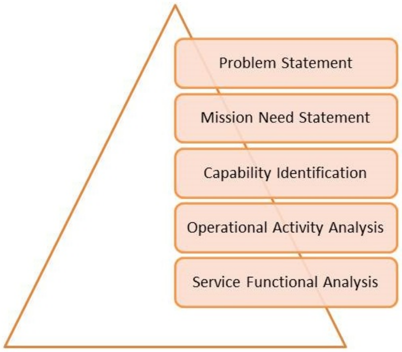

Problem Statement
Mission Need Statement
Capability Identification
Operational Activity Analysis
Service Functional Analysis

Figure 12. General DoDAF Architecture Development Process

the remainder of this section explores.

Once the problem statement was clearly defined the next architecture product known as the mission need statement was developed. The mission need statement was used to define the target objectives of the architecture and also helped further refine the scope of the MCS architecture. The problem statement focuses on the operational gap and the mission need statement identifies what is required to address the identified gap. It was imperative that these two high level architecture products be clearly understood by all the research stakeholders since they are critical to defining the scope and system boundary for the MCS architecture.

Next an overall concept of operations was envisioned for how the mission need statement might be fulfilled. This concept of operations was captured in the DoDAF view the High-Level Operational Concept Graphic, or OV-1. The OV-1 was developed to provide a high level view of the MCS participants and their interactions. Also, the OV-1 was developed to provide an initial scope of MCS architecture. This view was also used to provide a photographic description of an applicable scenario for the MCS architecture.

The next step in the architecture development is the decomposition of the mission need statement into a set of desired system capabilities. As is traditional in capability based analysis the capability set developed is intended to be system agnostic, and represents the required capabilities that any system attempting to fulfill the mission need statement should exhibit. Deriving the first layer of the system architecture in this manner is known as a capability based assessment and establishes direct tractability from the lower level capabilities to the higher level architecture products. Utilizing the DoDAF guidance for capability identification resulted in a hierarchical decomposition of the higher level abstract capabilities into lower level implementable capabilities. The hierarchical decomposition ensures that as the architecture's operational activities, system functions and services are developed from the capability set there is tractability back to the problem and mission need statements. DoDAF guidance recommends this approach to help ensure concordance across the MCS architectures products. The capability hierarchy was captured in the DoDAF product CV-2, which is known as the capability taxonomy.

Following the capability identification, the lower level capabilities are used to establish a high level set of operational activities that the architecture will perform. The operational activities are decomposed in similar fashion as the capabilities with the most abstract activities being at the top of the hierarchy, and more specific operational activities forming the leaf levels of the hierarchy. This hierarchical view of the operational activities was captured in an Operational Activity Decomposition Tree, or DoDAF OV-5a. The OV-5a is a system agnostic view, but the DoDAF framework supports the allocation of the operational activities to system participants or nodes. Following the guidance each operational activity was assigned to participant nodes within the architecture. This allocation was accomplished utilizing an Operational Resource Flow Description or DoDAF view OV-2. This DoDAF view was also used to capture the high level flow of resources between participant nodes. The OV-2 provided the initial set of resource need lines which were decomposed into more specific resource exchanges in the service architecture views.

In traditional systems architecting, the operational activities are used to derive the next layer of the system architecture, which is a set of the system functions. However, in developing the MCS architecture as a SOA, the system functions are replaced by a set of services. Effectivelly, services are system functions that are exposed to other architecture participants through defined interfaces, these interfaces are typically exposed through a web interface. The architecture services not only identify the open interfaces between the architecture participants, but also the required messages that flow between them and the message's structure. As a part of this research the development of an exact data protocol or message format was not within the scope of the architecture development. Future iterations of the MCS architecture will require this expanded scope, but during this initial research effort the desired interactions of the MCS participants was able to be modeled and simulated without the need to explicitly define a Web Service Description Language (WSDL) or EXtensible Markup Language (XML) schema [4]. Instead each of the service's messages and data objects were represented within the DoDAF view by a notional representation of what the participants might exchange to request or process a service. The DoDAF service views focus on defining a list of services and their required resources, where these service are hosted, and the flow of resources from one service to the other [30]. It is not uncommon for a single operational activity to rely on a number of services to facilitate their execution [7].

An issue does arise when attempting to utilize the native DoDAF nomenclature and service views for describing model entities when considering a SOA. The DoDAF does not include a meta-model that describes the types of service model entities or how these entities should interact. Consequently this can lead to entity utilization inconsistencies between views. Alternatively the service view may not adequately convey its intended purpose effectively since each entity stereotype can be used differently between architectures. To avoid the possible concordance issue the service views for the MCS architecture were developed utilizing the SoaML guidance. Each service architecture product utilized the SoaML diagram guidance and nomenclature to describe the model entities and their service interactions. The SoaML products were used to fulfill the intent of the DoDAF guidance listed in the CIO architecture product descriptions [30]. The service views focused on describing the behavioral flow within the MCS architecture. These views are ultimately what guided the development of the modeling and simulation effort, and allow for evaluation of the architecture variants. The development of the SoaML products helped insure that this methodology supports the tenants of a SOA to be loosely coupled and reusable [31]. The service views were the final architecture products developed to support this research. The next section provides additional details for each of the MCS architecture products and their use in guiding the MCS M&S activities.

3.3 Architecture Product Descriptions

The architecture framework utilized for the development of the MCS architecture is the DoDAF Version 2.02 following the latest guidance released by the DoD Chief Information Officer (CIO). The CIO is responsible for creating and enforcing DoD system architecture policy. The CIO guidance states that all DoD system architecture efforts must conform to DoDAF guidance ”to the maximum extent possible.” [30] Primarily each DoDAF product is required to be fit for purpose, or able to convey its intended message without a large amount of additional context. Therefore, the system architect is not limited to using only DoDAF resources.

Table 1 lists the architecture products that were identified in the previous section. The table includes the product’s intended scope and a brief description. The description of each DoDAF view is referenced from the DoD CIO description and each view supplemented by a SoaML diagram utilizes guidance from the SoaML specification. An identifier is included in column three of the table if the DoDAF product was used to directly define interactions within the M&S activities.

The use of SoaML was primarily selected for its well defined meta-model for development of SOA and its similar scope to the DoDAF views. The SoaML scope is defined in the specification as:

> “Defining service consumers and providers, what requisition and services

Table 1. List of included DoDAF and SoaML products

DoDAF View	Description	M&S
OV-1	Depicts a high level graphical representation of the operational concept	included
OV-2	A description of the Resource Flows exchanged between operational activities	–
OV-5a	Hierarchical decomposition of operational activities	–
CV-2	Hierarchical decomposition of capabilities	–
Service Architecture (SvcV-4)	The identification of service contracts, service roles, and their interconnections (modeled using SoaML guidance)	included
Service Contracts (SvcV-4)	The roles and messages required to execute perform and request services and the flow of data/messages between participants (modeled using SoaML guidance)	included
SvcV-10c	Describes critical sequences of events during service execution (modeled using SoaML guidance)	included

> they consume and provide, how they are connected and how the service functional capabilities are used by consumers and implemented by providers in a manner consistent with both the service specification protocols and fulfilled requirements" [32]

This SoaML scope clearly encompasses the DoDAF SvcV-4 and 10c purpose and scope.

In order to provide the required feasibility analysis, only a select set of operational activities and service operations needed to be modeled architecturally. As indicated above the application of MBSE facilitates the evaluation of the dynamic DoDAF and SoaML views. Primarily the M&S will execute the SvcV-4 and SvcV-10c flows to compare the set of architectures. The following section details how the service architecture views were utilized to guide the M&S activities and the development process of the MCS architecture model.

3.4 Modeling and Simulation Description

The M&S development is intended to provide a means for architecture feasibility analysis, and to enable sensitivity analysis of a set of MCS architecture variants. The overall goal of the M&S is to understand how well variations of the MCS architecture satisfy the identified MOEs. Often the MOPs of a system are used as inputs to the M&S sensitivity analysis or numerical analysis [33]. The complete set of MOEs of the MCS architecture is determined by the performance of multiple service's. Each service's performance is directly linked to the values of its MOPs. The MOPs used as inputs for the M&S effort were derived from a combination of architecture configuration parameters and standard values identified in the research literature for the appropriate model entity. Once a MOP was set for an architecture variant it remained consistent throughout the simulation analysis period. For example, the number of receivers providing inputs into the MCS architecture will remain at a constant number during a simulation, but may vary from variant to variant. The selection and assumptions detailing the MOPs value is included in the data and analysis section. The remainder of this section details the STK simulation development and how it models the MCS architecture, the methods used for developing entity behavior, and a description of the how the MCS MOP were extracted from the model.

The first step in the M&S process was to select an appropriate modeling technique for evaluating the MCS architecture. The MCS architecture is an event driven system that responds to stimulus from the simulated environment. To extract the MOEs from the simulation the model entity's internal states and architecture state variables needed to be evaluated at fixed time steps. To facilitate this type of analysis the use of a Discrete Event Simulation (DES) technique was selected. Other modeling techniques such as Agent Based Modeling (ABM) or real-time simulation were considered and subsequently dismissed as modeling options. ABM is typically used when "Individual behavior is nonlinear and can be characterized by thresholds, if-then rules, or nonlinear coupling..." and when "Individual behavior is complex" [34]. Given the unknown nature of the behavior of participants within the MCS architecture a notional set of rules could have been developed to govern the entity behavior, but this could result in non-representative behavior of the participants. Trend analysis of real world data could be used to develop behavior models for each MCS participant but it was beyond the scope of this research.

Real-time simulation was also not an option due to time constraints. The technical solution for monitoring GPS QoS from a phone through a web interface has been demonstrated via the MITRE TADA program [24]. However, utilizing the experimental TADA application on a scale large enough to effectively simulate the MCS architecture would have required manpower levels that were not available. Also the prohibitive nature of GPS jamming testing made this type of analysis unfeasible in the given time to conduct this research. Another prohibitive constraint was that the data processing capability of the aggregated TADA information would require a significant amount of development effort, and addition of bidirectional communication to the TADA application would have needed to be developed; both of which are beyond the scope of this research.

DES enables the use of stochastic modeling techniques and evaluation of the simulation's state based on a desired sampling rate. The use of discrete time steps also, stochastic model attributes, and predefined behavior of model participant's allowed for the consideration of a broad array of propagation based simulation tools. Having a broader set of modeling tools to select from was critical to achieve the goal of simulating the participants interaction within a representative GPS environment. Also the ability to perform replications once a DES is defined would eventually allow for statistically relevant analysis of the MCS architectures performance. However, the required number of replications was not executed as a part of this research due to computational and time constraints. The implications of this constraint are explored further in chapter IV

Once the appropriate M&S process was selected the appropriate tools for implementing the model and evaluating the architecture were determined. Several different tools were considered as potentially viable modeling enviornments. These programs consisted of the Aerospace Corp's Satellite Orbit Analysis Program (SOAP), the SMC advanced systems and development directorate's System Effectiveness Analysis Simulation (SEAS), and AGI's STK were considered as potential tools to model each of the MCS architecture variants. All of the programs are propagation based models and are able to model the GPS constellation and perform access calculations between model entities. However, SEAS is most often used to model two opposing forces in a more traditional combat modeling oriented analysis. It is also primarily used to support ABM simulations which were not required for this research [35]. SEAS is part of the Air Force Standard Analysis Toolkit, was developed in conjunction with the Air Force Space and Missile Command (AFSMC), and is approved, accessible, and validated for us in Air Force combat modeling simulation research [36]. Ultimately due to the requirement to utilize a simulation environments that not only supports geo-spatial analysis, but also Radio Frequency (RF) propagation characteristics, AGI's STK was selected.

STK is a physics based geometry engine that is capable of solving dynamic complex analysis problems over a given time period [37]. STK is a common tool utilized by the AFSMC and by AFIT for analyzing complex space problems [38] [39]. To simulate the expected message exchanges and GPS sensor capability the STK RF computation capability was used to compute the received GPS signal quality by all the GPS user model entities, capture the effects due to jamming, and compute the GPS user's signal to noise plus interference level, $C/(N_o + I_o)$. Each of the MCS simulation entities were developed from the built in STK object tools; such as ground vehicles, transmitters, receivers, etc... The GPS constellation was imported into the model from AGI's satellite database which is frequently updated with the GPS satellite orbital parameters and status information published by the Joint Space Operations Center (JSPoC). These entities and their behaviors were defined via the built in STK connect capability.

STK connect "is a library of string commands for STK, originally designed to operate over a TCP/IP socket." [11] Connect is a powerful tool that allows a user to feed in data to STK via any TCP/IP third party application. Below in Figure 13 is a depiction of the STK connect capability implementation.

The connect interface was used to automate the generation of the MCS architecture within the STK application. Similarly the commands to perform the required access and RF calculations were sent through the connect interface. The connect commands rely only on string inputs from the external program once the basic command structure for each model entity is constructed they are easily modified to rapidly generate the MCS architecture variations. The last commands sent through the connect interface for each architecture variant were the commands to generate the data analy-

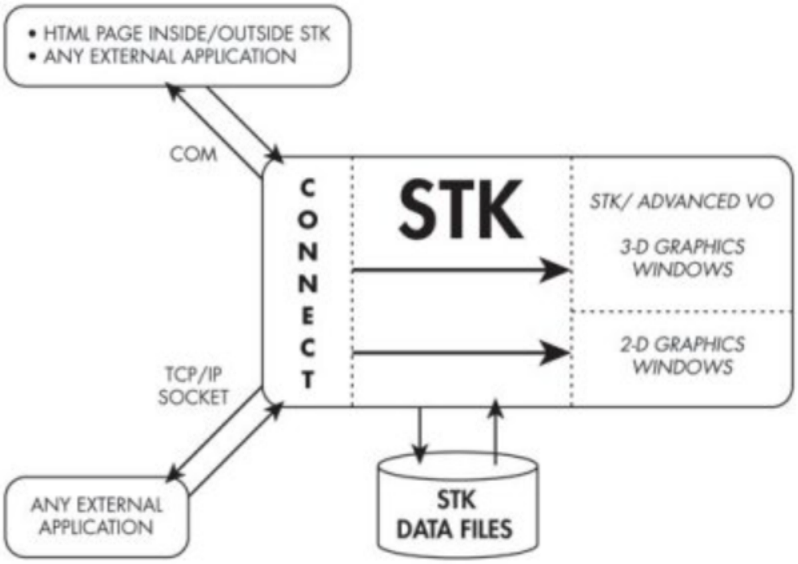


Figure 13. STK Connect [11]

sis reports and extract them to a basic text document. STK supports custom export configurations for the requested reports. The minimum number of reports generated for each architecture variant was over 2000 the files were standardized into a basic comma delimited format for ease of processing the data during data analysis.

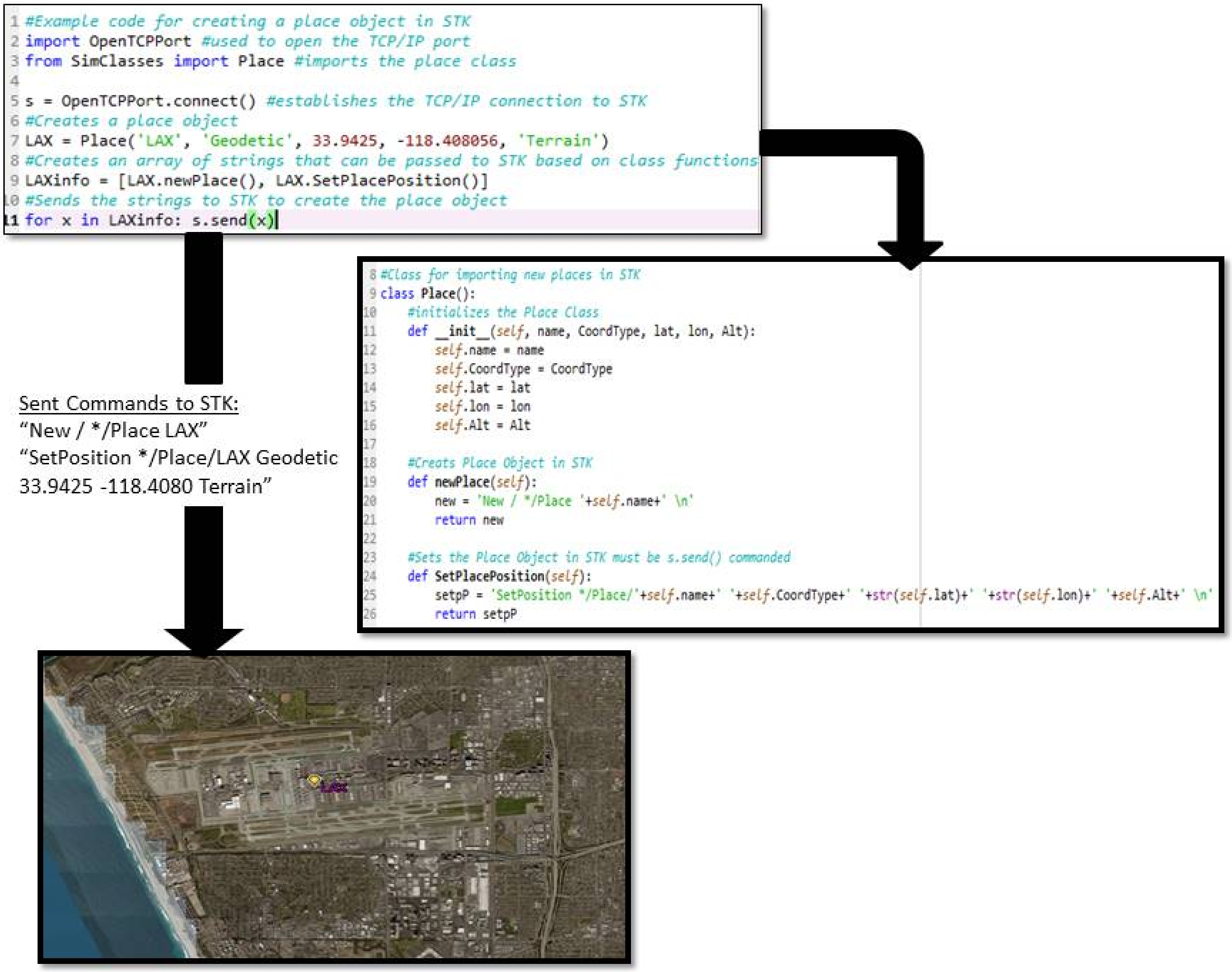
As indicated above in Figure 13 the connect interface is driven by a third party application. The programming language Python was chosen to develop the string commands that were used to instantiate each MCS architecture variant, for parsing the exported reports from STK, and for formatting the data prior to uploading it to the geo-spatial analysis software. Python is an open source high-level data structured programming language that supports object-oriented programming methods and can be extended through installation of third party developed Python modules [40]. Python was chosen due to its ease of availability, the level of familiarity with the programming language by the researcher, and its proven track record for being capable of driving complex analysis in STK [39] [38]. Together STK and Python provided all the tools required to model the MCS architecture variants and simulate the basic message exchanges between the architecture participants.

3.5 MCS Model Development

The following sections describes the execution of the MCS architecture variant simulation and identifies the relevant data generated by each simulation. The method for processing of the architecture data are discussed later in this chapter. This section also describes the development process of the MCS architecture model and the assumptions that framed the MCS architecture variants.

In order to evaluate the GPS MCS architecture a representative scenario was developed based on the GPS disruption events that occurred near the Newark, NJ airport in August of 2013 [21]. A baseline scenario was modeled that would exercise key services of the MCS architecture, and then the baseline scenario was perturbed to evaluate the effectiveness of different architecture configurations. This section includes the specifics of how the modeling tools described above were used to drive the development of each scenario, the assumptions that apply to the model, and how the simulation's data were extracted for analysis.

As indicated above, Python was used to send commands to STK via the connect interface. To provide flexibility when automating scenario generation, tracking individual entity attributes during generation, and facilitate addition of non-native STK data an Object Oriented Software Development (OOSD) approach was used during the Python code development. This OOSD approach entailed the creation of unique classes for each STK object. Each class included a set of functions that were called to generate the required string line commands that are sent to STK via the TCP/IP connection. The classes and functions were developed on an as needed basis to produce the desired behavior and are not inclusive of every STK command. Additionally, classes and functions were only created to manipulate the standard STK objects (satellites, transmitters, receivers, etc...), unique or new STK root objects were not defined as a part of this research. Figure 14 is an example of how the Python classes were used to provide commands to STK. The complete set of Python code is available in electronic format through the AFIT systems engineering department.



```
#Example code for creating a place object in STK
import OpenTCPPort #used to open the TCP/IP port
from SimClasses import Place #imports the place class

s = OpenTCPPort.connect() #establishes the TCP/IP connection to STK
#Creates a place object
LAX = Place('LAX', 'Geodetic', 33.9425, -118.408056, 'Terrain')
#Creates an array of strings that can be passed to STK based on class functions
LAXinfo = [LAX.newPlace(), LAX.SetPlacePosition()]
#Sends the strings to STK to create the place object
for x in LAXinfo: s.send(x)
```

```
#Class for importing new places in STK
class Place():
    #initializes the Place Class
    def __init__(self, name, CoordType, lat, lon, Alt):
        self.name = name
        self.CoordType = CoordType
        self.lat = lat
        self.lon = lon
        self.Alt = Alt

    #Creats Place Object in STK
    def newPlace(self):
        new = 'New / */Place '+self.name+' \n'
        return new

    #Sets the Place Object in STK must be s.send() commanded
    def SetPlacePosition(self):
        setpP = 'SetPosition */Place/'+self.name+' '+self.CoordType+' '+str(self.lat)+' '+str(self.lon)+' '+self.Alt+' \n'
        return setpP
```

Sent Commands to STK:
"New / */Place LAX"
"SetPosition */Place/LAX Geodetic 33.9425 -118.4080 Terrain"

Figure 14. MCS Simulation Development Flow

Being able to manipulate simple key values within a Python script and automate the simulation generation and analysis was critical to exploring the MOP trade space of the MCS architecture. However, to quote Dr. J. O. Miller of the AFIT Operational

Sciences Department, ”You can’t model everything. And even if you could, models are always wrong, but some are useful”. In order to address Dr. Miller’s first comment a set of assumptions were developed to bound the scope of the simulation. As to the later portion of his quote the developed model was at the very least useful to this particular research effort.

The key assumptions correspond to all the MCS architecture simulations. There were several factors that contributed to establishment of these assumptions. The GPS receiver’s parameters were derived from the Kaplan and Hegarty text [16]. Any additional parameters that were not set based on the text were set to the default STK settings. The parameters from the jammer were set based on the report by Chronos Technology Limited which lists some examples of GPS jammer parameters [2]. For the GPS constellation the STK default GPS transmitter was used with the provided GPS block II L1 antenna. The assumption set for each model entity is listed below all of these assumptions and values were input into the simulation via a string command passed through the connect interface from Python to STK:

GPS Satellite

1. Each GPS satellite’s ephemeris data consist of the latest information from the JSPOC
2. Each GPS satellite conforms to the satellite performance specification identified in the GPS SPS

GPS Transmitter

1. The default GPS transmitter provided in STK adequately represents the performance identified in the GPS SPS
2. The transmitters are a homogeneous set

GPS User

1. All users have access to a communication network for relaying GPS data
2. All users are participatory members of the GPS MCS service
3. All users provide data when requested
4. When mobile, the receivers move at an average pace of 1.39 meters per second and a standard deviation of .15 meters per second [41]
5. All users are considered a mobile user

GPS Receiver

1. All receivers are homogeneous
2. Receivers can only receive the GPS L1 signal
3. Receivers report correct signal to noise ratio
4. Receivers do not possess any anti-jam GPS capability
5. Receivers require a minimum $C/N_o = 41.9dB - Hz$ [1]
6. Receivers are capable of operating up to $C/(N_o + I_o) = 28dB - Hz$ [1]

GPS Jammer User

1. The jammer user will travel the same path for all architecture variations
2. There is only ever one jammmer user

GPS Jammer Transmitter

1. The jammer is only capable of jamming the L1 GPS frequency band
2. the jammer is only capable of RF jamming and not able to perform spoofing

[1] Example values derived from [16, p. 262-265]

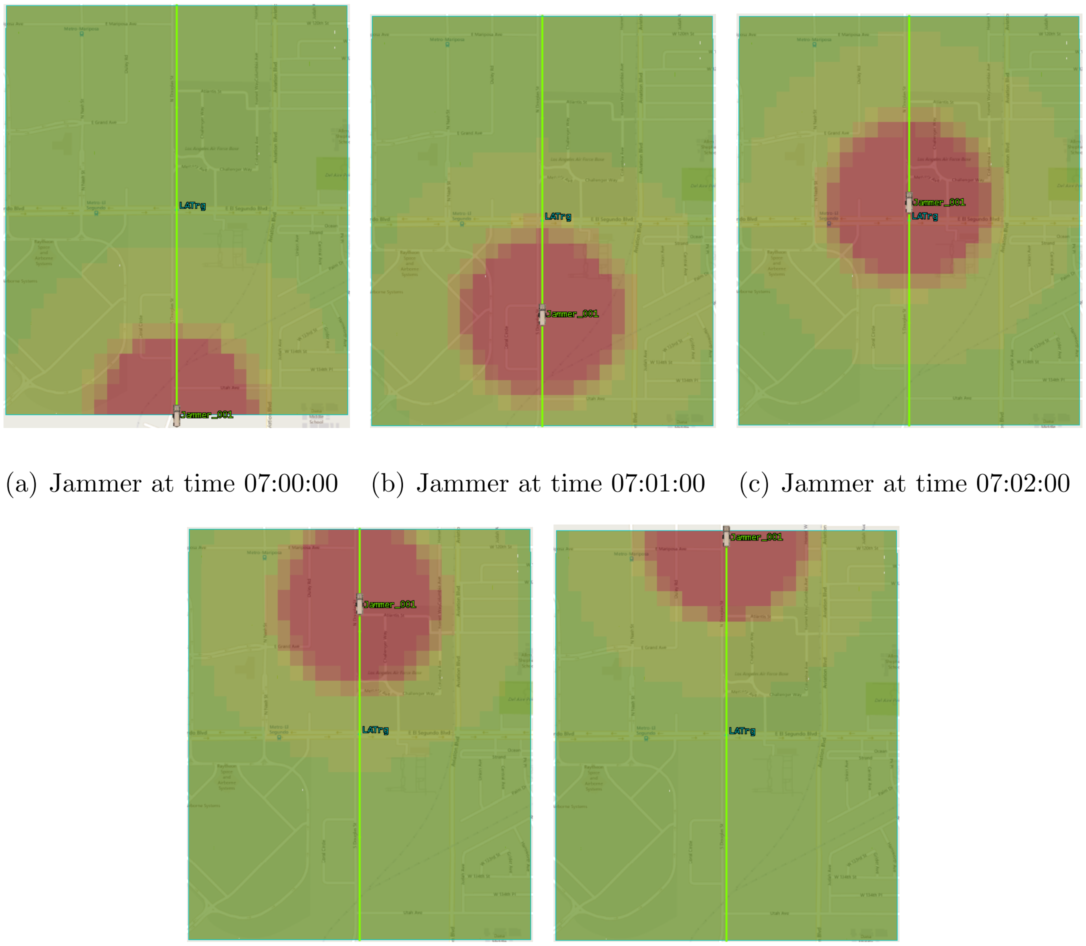
3. During the period of interest the jammer is always on
4. The power provided by the jammer during each simulation run does not fluctuate
5. The radiation pattern of the jammer remains fixed as hemispherical
6. The jammer speed is set to a value of eight meters per second, and does not fluctuate (in reality could be slower due to Los Angeles traffic)

Physical Environment

1. The impacts due to weather or atmospheric degradation of the GPS signal are not included in any simulation
2. Access and RF interference due to building features and terrain is not included as a part of this analysis

The MCS architecture scenario created using the method and assumptions described above consisted of a set of GPS receivers operating in and around Los Angeles Air Force Base in Los Angeles, California. For each architecture variant, a set of homogeneous receivers are placed throughout the AoI which is defined by a one square mile square with the center of the square placed at $(33.9164, -118.383)$ degrees latitude and longitude respectively. During the operational period a single GPS jammer attached to a moving vehicle traverses the AoI from south to north along a fixed route. Figure 15 depicts the jammer as it traverses the AoI. The area of influence of the jammer is calculated by STK and is represented in red.

To perform the architecture variant analysis, the architecture configuration and/or behaviors of the MCS model entities were varied across the options in table 2. As described above this was accomplished via manipulation of the input values within the Python simulation manager which adjusts the string commands sent to STK. Table 2 includes the set parameters that were varied to perform the sensitivity analysis. The



(a) Jammer at time 07:00:00 (b) Jammer at time 07:01:00 (c) Jammer at time 07:02:00

(d) Jammer at time 07:03:00 (e) Jammer at time 07:03:40

Figure 15. Baseline scenario and jammer area of influence

variant set resulted in sixteen unique combinations for the MCS architecture. For all variants, including the static user variants, it is still assumed that the receivers are attached to a user who could be mobile. The potential of the users to be mobile applies a constraint to the analysis that a receiver's position cannot be known when jammed. This constraint prevents the architecture from simply finding the center of a cluster of receivers who have a degraded GPS signal. Of note this constraint was what ultimately lead to the inclusion of the ArcGIS statistical analysis tool set. Also as indicated in the assumptions list, interfearance due to buildings are not included in this analysis. Consequently, for the mobile architecture variants the receiver's movement

Table 2. Architecture Variation Attributes

Number of Rx	500	1000
Sensing Rate	Slow (1 m)	Fast (15 s)
Rx Movement	Static	Random
Rx Distribution	Uniform	Normal

was not restricted due to intersecting a building. Additionally this eliminates any obstructions between all the transmitters and receivers, so losses due to signal attenuation through structures is not included in the RF calculation. The static receiver sets are generated similar to the initial instantiation of the "Random" movement pattern variants. Both variants utilize two independent random number draws from the assigned distribution to independently generate the latitude and longitude values for each user way-point set or static position. Each distribution is centered on the AoI; 33.9164^o *latitude* and -118.383^o *longitude*. The normal distribution utilizes a standard deviation of 0.00364 degrees, and the uniform distribution uses the edges of the AoI to bound the random number draws. Exact values of the average number of users in a one square mile area that utilize the GPS service on their mobile device at different periods of the day was unavailable for this analysis, so the average number of available receivers was estimated. This was done conservatively using data gathered by the U.S. census bureau from 2010. The average Los Angeles city population per square mile for the AoI is over 7000 [42]. Specifically the AoI chosen is near the El Segundo, Manhattan Beach, and Torrence subdivisions which have a population density of $3,048$, $8,923$, and $7,102$ per square mile respectively. As indicated in chapter II an MCS architecture can rely on participatory or opportunistic sensing activities [5]. As indicated above each architecture variant assumes all users are active participants.

Once all the scenarios were developed and simulated the final portion of the MCS analysis consisted of extracting the MOE for each of the architectures. As indicated in Chapter I the overall goal of this work is to monitor the GPS QoS via MCS. In order to control the scope of this initial analysis a single MOE was chosen to evaluate the architecture variants. The MOE is as follows:

MOE 1: RF Jammer Identification - The ability to identify an RF interference source within the specified AoI. The MOE will be evaluated by comparing the estimated jammer's latitude and longitude with the jammer's simulated "truth" location.

In order to extract MOE 1, several data transformations and processing tools were required. STK was used to simulate the GPS RF sensor performance utilizing the STK communication system object to define the relationships between the GPS jammer, receivers, and transmitters. Once the RF relationships were defined the communication information was calculated and exported via the "ReportCreate" command. This command was used not only to obtain the communication system information for each receiver, but also the latitude and longitude for all the GPS users. These reports were output in a text format using a one second time step for the full five minute scenario. Python was then used to aggregate the data for each receiver into a single “sensing data report.” Initially an attempt was made to process the receiver data utilizing a Python image processing library known as OpenCV to extract MOE 1. However, image analysis did not allow for direct manipulation and analysis of the raw latitude and longitude data. OpenCV does include built in capability to do shape and object recognition, but due to the random distribution and disjointed nature of the sensors image detection proved challenging at best to identify the jammer's location. Ultimately, it was determined that another tool would be required to analyze the sensor data to extract the desired MOE.

A commercially and readily available tool for geo-spatial processing was identified to complete the MCS architecture evaluation. The GIS tool developed by ESRI known as ArcGIS was chosen due to its reliability, pedigree, and availability. ArcGIS specializes in processing geographic information and includes a suite of built in spatial analysis and statistics tools. ArcGIS includes built in capability to manipulate GIS data through their software as a service option. This means that an organization can utilize the full suite of ArcGIS tools through a web interface and publish analysis results directly to the internet for viewing [43] [44]. ArcGIS's ability to ingest GIS standardized data sets and publish them to an open web interface is the exact behavior desired from a SOA. More detail is included in chapter V on possible expansion of the use of this tool's web publishing capability.

In order to process the STK simulated data the GPS sensors data needed to be formatted into GIS standardized format, imported into ArcGIS, filtered based on desired receiver performance and time instance, and finally have the spatial analysis and statistics tools applied. Lastly the estimated jammer location produced by the ArcGIS analysis process was exported for a final error calculation which was performed in Python. In order to import the data into ArcGIS the thousands of reports per architecture variant had to be parsed and initialized as a recognizable GIS data format. Another benefit to utilizing ArcGIS is it's built in Python library known as Arcpy. Arcpy provides access to all of the ArcGIS commands that control the import, export, manipulation, and analysis of any of the GIS data. Figure 16 describes the flow and analysis of the receiver data between Python and ArcGIS.

The first three steps of the GIS analysis flow are self descriptive. The final portion of this chapter is dedicated to explaining the remaining steps in the flow diagram.

In order for the GIS spatial analysis tool kit to be considered reliable the ESRI guidance recommends that the data be placed into a projected coordinate system. A key tenent of working with GIS data is to understand and control the spatial

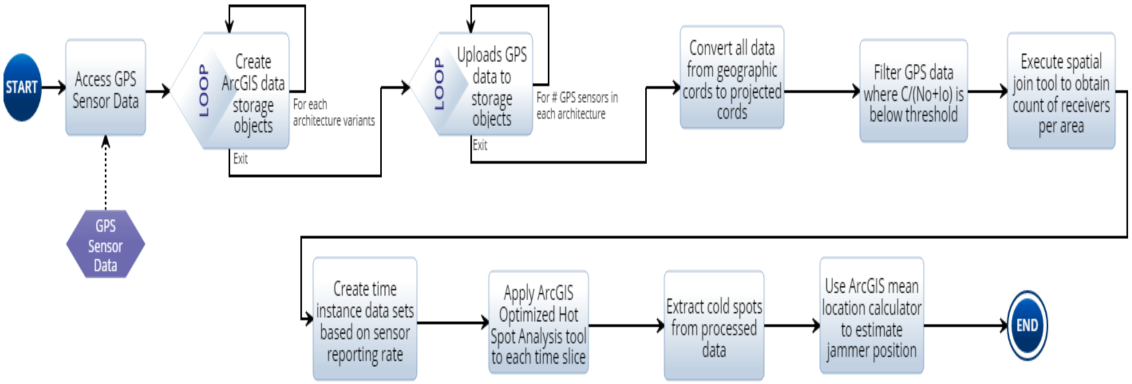

START
Access GPS Sensor Data
GPS Sensor Data
LOOP
Create ArcGIS data storage objects
For each architecture variants
Exit
LOOP
Uploads GPS data to storage objects
For # GPS sensors in each architecture
Exit
Convert all data from geographic cords to projected cords
Filter GPS data where C/(No+Io) is below threshold
Execute spatial join tool to obtain count of receivers per area
Create time instance data sets based on sensor reporting rate
Apply ArcGIS Optimized Hot Spot Analysis tool to each time slice
Extract cold spots from processed data
Use ArcGIS mean location calculator to estimate jammer position
END

Figure 16. GIS Analysis Flow Diagram

reference coordinate system of the during data analysis. A map projection is simply ”the method by which the curved surface of the earth, or a part of it, is represented on a flat surface on a certain scale [8]. The STK simulation captures and reports data using a three dimensional spherical model of the earth, so the output data must be projected for processing. STK includes several models for representing the earth’s shape. Given the short scenario time period orbital perturbation were not considered and a simple spherical earth model was selected. STK’s default World Geodetic System 1984 (WGS 1984) was selected for defining the position of receivers on the globe. This is a common format for GIS data and ArcGIS contains a suite of built in conversions to project WGS 1984 data. As identified above, the scenario is set in the Los Angeles area. An appropriate projection for the Los Angeles area is the NAD 1983 State Plane California V FIPS 0405. This projection is not the only projection that can align the GPS sensor data for accurate processing, but it is the spatial reference for data produced by the Los Angeles county GIS Steering Committee. This allows for all the generated data to be compatible with the official GIS governing body for the AoI. The Los Angeles GIS Steering Committee has made a concerted effort to make GIS source data available to the public [45]. Including data sets for all buildings in Los Angeles county, road ways, traffic lights, bike trails, etc... This publicly available data in conjunction with the ArcGIS Online Los Angeles base map were used to develop a data base layer to frame and provide context for the GPS sensor data. Ensuring interoperability with the steering committee data format lays the ground work for future research regarding this topic to easily include more source data from the GIS governing body in the AoI.

Once the data was projected appropriately it was filtered to represent an actual response from a request for GPS QoS. STK provides all the latitude and longitude of the sensors regardless of their signal to noise ratio, so a Structured Query Language (SQL) filter was applied to the data within ArcGIS to display and analyze only those points where the signal to noise ratio is above the defined threshold of 28 dB-Hz. ArcGIS does have a suite of temporal analysis capabilities. However, the desired hot spot analysis tool cannot be run as a function of time on a single data set even if each data point is time tagged. To facilitate the desired analysis Python was once again called upon to filter through all the architecture variants based on the desired sampling rate to create new filtered data set. Each of these time slices were saved to the ArcGIS feature layer format which is one of the supported formats for the optimized hot spot analysis tool. The data was then reconstructed after the optimized hot spot analysis was completed. Next the data was filtered to only show those areas where the number of users in a given area of the AoI statistically lower than the average number of users. In the latest release of ArcGIS, version 10.3.1, there is added capability for an emerging hot spot analysis, and the potential uses of this tool are described in chapter V, but it was not made available during the course of this research. The built in ArcGIS optimized hot spot analysis tool makes use of a set of pre-processing optimization routines to provide a clearer indication of receiver clustering or absence and then executes the calculation of the Getis-Ord

Gi statistic. The Getis-Ord Gi statistic measures the association of entities that are spatially distributed on a "local" scale. The following quote from the Getis and Ord paper best describes the G-statistic:

> "[The] statistical measures the degree of association that results from the concentration of weighted points (or area represented by a weighted point) and all other weighted points included within a radius of distance *d* from the original weighted point [46]"

Once calculated if the G-statistic is determined to be significant then for the local area then the null hypothesis that the points in the area are randomly distributed is rejected. ArcGIS utilizes the G-statistics Z-score to display where large or small concentrations of points or values are clustered. A high Z-score corresponds to areas with a number of points high above the calculated mean within a distance d of the point of interest, and low z-scores represent a below average number of points [46]. By using the optimized hot spot analysis tool in ArcGIS the critical distance d is automatically determined through the use of an incremental Moran's I measurement. The Moran's I series calculation return a distance that indicates a peak intensity of spatial clustering. ArcGIS performs the hot spot analysis using this provided distance, and utilizes the G-statistic z-score to determine if the spatial pattern reflects a theoretical random random. The value of the z-score also applies a confidence interval to the predicted spatial analysis via the standard normal standard deviation values. Figure 17 below shows the ArcGIS standard normal distribution for determining the Gi-Bin of each analyzed area. Each significance level corresponds to a confidence interval with the first standard deviation resulting in a 90% confidence interval, then 95% and 99% for the next two standard deviations [47]. In order to apply the optimized hot spot analysis to a raw count of entities in a given area the number of receivers able to provide a valid GPS signal were aggregated in to sub-areas. As indicated in the flow diagram, the optimized hot spot analysis tool was then applied to each of

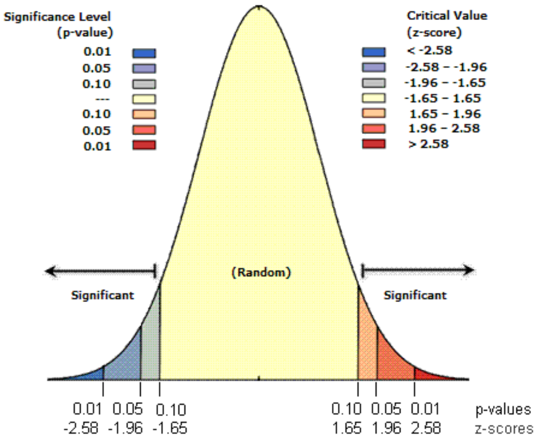


Figure 17. Standard normal distribution for Gi-Bin determination

the data sets. In order to identify the estimated jammer position the analysis focused on where GPS receiver responses were below the expected average by two standard deviations or more. These cold spot areas were extracted for each sample time. The last step of the analysis process utilized the ArcGIS spatial mean center tool. This tool calculates the geographic center of a data set and generates a point feature layer marking the mean location value. Since the input data to the mean center calculation is a polygon feature class the mean center tool utilizes the center of the polygon feature area to perform the calculation. This introduces some aggregation error into the jammer location estimate. The generated mean center value was calculated for each time slice where the optimized hot spot tool produced a usable data set. This center location was the value used as the estimate for the jammer's position at each time step.

3.6 Method Review

The research method described in this chapter utilized a combination of system architecture frameworks and multiple modeling and simulation resources. The method used to generate and analyze the MCS architecture follows similar methods described in chapter II [48][49][6]. The methodology centers around development of key architecture products that are used to insure concordance from the capabilities down to the lower services. Then with these service and resource flow views defined a model of a candidate scenario is developed for simulation and analysis. Then the simulation activities were executed focusing on exploring a small subset of the total architecture functionality. The simulation activities are used to understand the interactions of the MCS architecture and what factors or MOPs might impact the desired MOE. The next chapter details the data that was generated by following the methodology described in this chapter.

IV. Data and Results

The data and analysis chapter is focused on presenting the architecture products that were developed as a result of the method described above. The lower level SoaML architecture products describe the GPS MCS architecture's activities and relationships to describe the interaction of the model entities used to capture the MOEs identified in chapter III. This chapter also includes the results of the M&S efforts, and concludes with a feasibility analysis of the candidate architectures.

4.1 Architecture Products

Prior to any DoDAF architecture views being developed an official problem statement was developed. As is typical of any DoDAF development effort which is guided by the architecture development pyramid in Figure 12. This problem statement is similar to the research objective identified in chapter I, but focuses on the operational problem as defined by the SMC/GPS directorate.

GPS Problem Statement - The GPS SPS QoS is not monitored, and is susceptible to attack and disruption which could jeopardize United States Critical Infrastructure.

With the problem statement clearly identified the mission need statement can be derived. Similar to the problem statement the mission need statement will borrow from sections of the research objective identified in chapter I. Here the focus is placed on identifying the top level desired capability that the underlying architecture ultimately achieves. The mission need statement focuses on the need to monitor the GPS SPS signal QoS on a global scale. The official mission need statement is identified below.

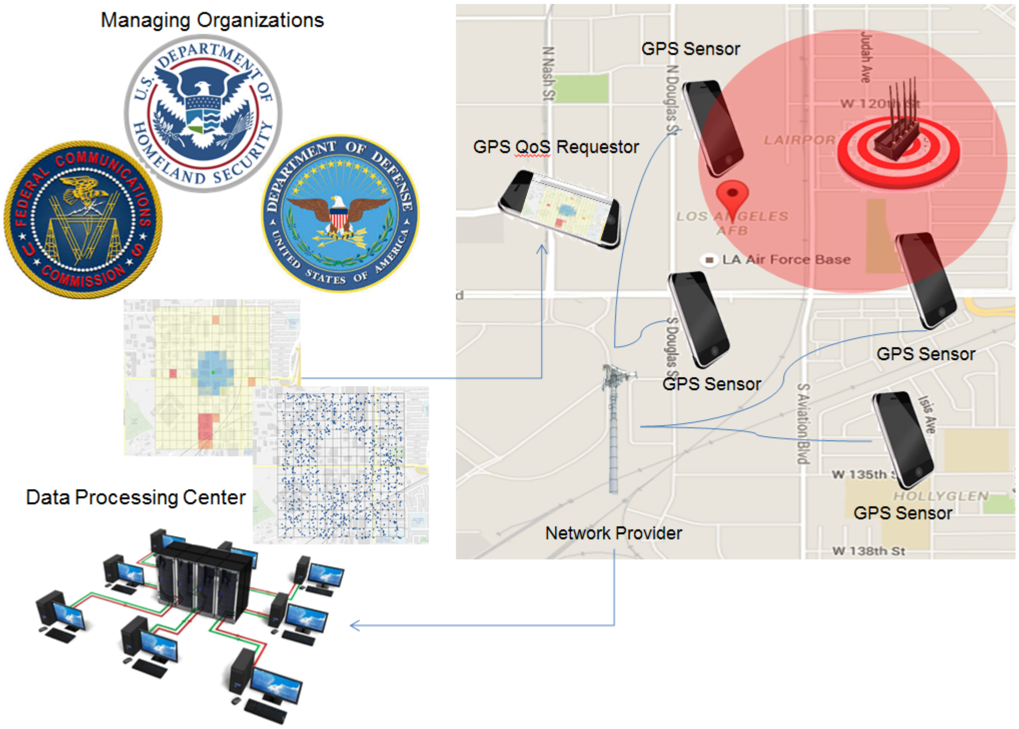


Figure 18. MCS Architecture OV-1

Mission Need Statement - The GPS SPS QoS needs to be monitored in near-real time on a global scale. Additionally the QoS data should be capable of providing insight into potential disruption events to the GPS signal in an AoI.

As indicated in the methodology the next DoDAF product presented is the OV-1, Figure 18. The OV-1 depicts the MCS architecture's intended mission. The captured scenario closely parallels the simulation conducted as a part of this research. The OV-1 depicts a set of GPS sensor providing inputs to a Data Processing Center through a Network Provider. The three governmental departments identified are the notional government organizations that are anticipated to be involved with the initialization and operation of the MCS GPS QoS architecture.

The next architecture product developed was the capability taxonomy, the DoDAF view CV-2. The capabilities identified in the CV-2 are derived from the mission need statement and define the MCS architecture capability set. Figure 19 highlights the derived high level capability set for the GPS MCS SOA. The complete CV-2 is included in appendix A. The CV-2 is decomposed from the top capability to monitor

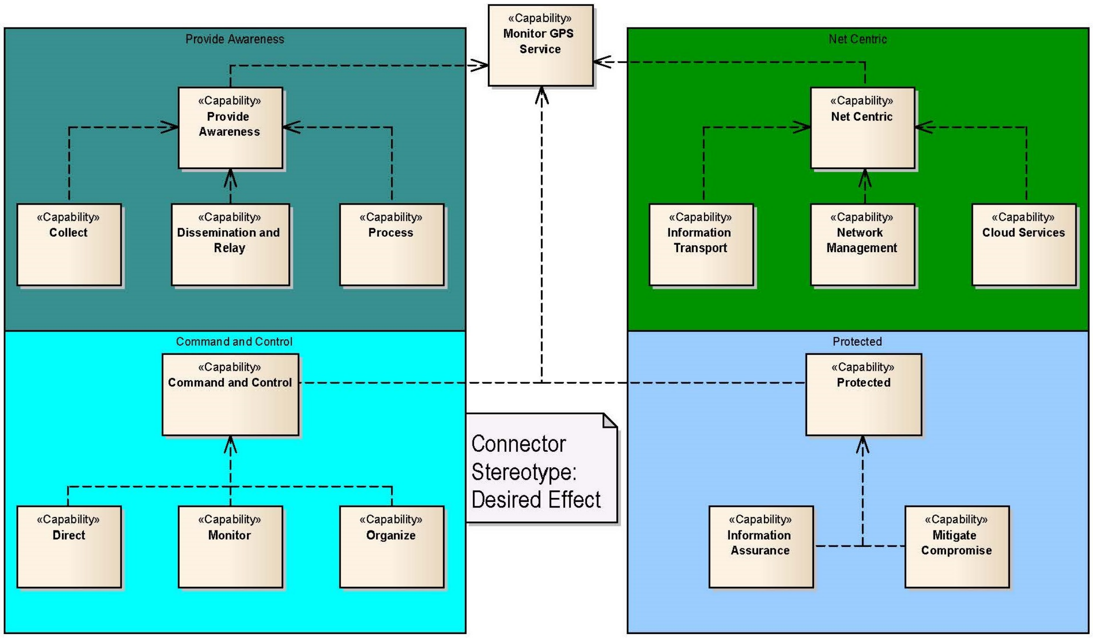

«Capability» Monitor GPS Service
Provide Awareness
«Capability» Provide Awareness
«Capability» Collect
«Capability» Dissemination and Relay
«Capability» Process
Net Centric
«Capability» Net Centric
«Capability» Information Transport
«Capability» Network Management
«Capability» Cloud Services
Command and Control
«Capability» Command and Control
«Capability» Direct
«Capability» Monitor
«Capability» Organize
Connector Stereotype: Desired Effect
Protected
«Capability» Protected
«Capability» Information Assurance
«Capability» Mitigate Compromise

Figure 19. MCS SOA Capability Taxonomy (CV-2)

the GPS service into 4 distinct swimlanes, which are color coded in the CV-2 diagram. These swimlanes help to frame the other hierarchical products that describe the MCS architecture. The capability set included as a part of the MCS architecture are derived from a combination of the Joint Capability Area (JCA)s and the literature described in chapter II. The JCAs are described by the DoD joint chiefs of staff which provides a common set of capability definitions for the DoD community [50]. Many of these capabilities are cross cutting between the JCA and the literature reviewed as a part of this research. However, the ”Cloud Services” capability is a unique capability not found in the JCA list. It is included in the top level capability set due to the need for a S^2aaS architecture to support distributed processing capabilities.

The next derived product for the MCS architecture is the hierarchical description of the operational activities, or the DoDAF view OV-5a. The use of the swimlanes provides clear traceability and helps ensure concordance between the derived activities and capabilities of the MCS architecture. Figure 20 and Figure 21 below depict the full set of operational activities. The OV-5a capture the set of activities the MCS architecture will support in order to provide GPS QoS information. The OV-5a lower level activities are assigned to the MCS participant nodes, and provide a clear definition of the activities each service participant will have to accomplish to fulfill the service contracts developed in the SoaML views.

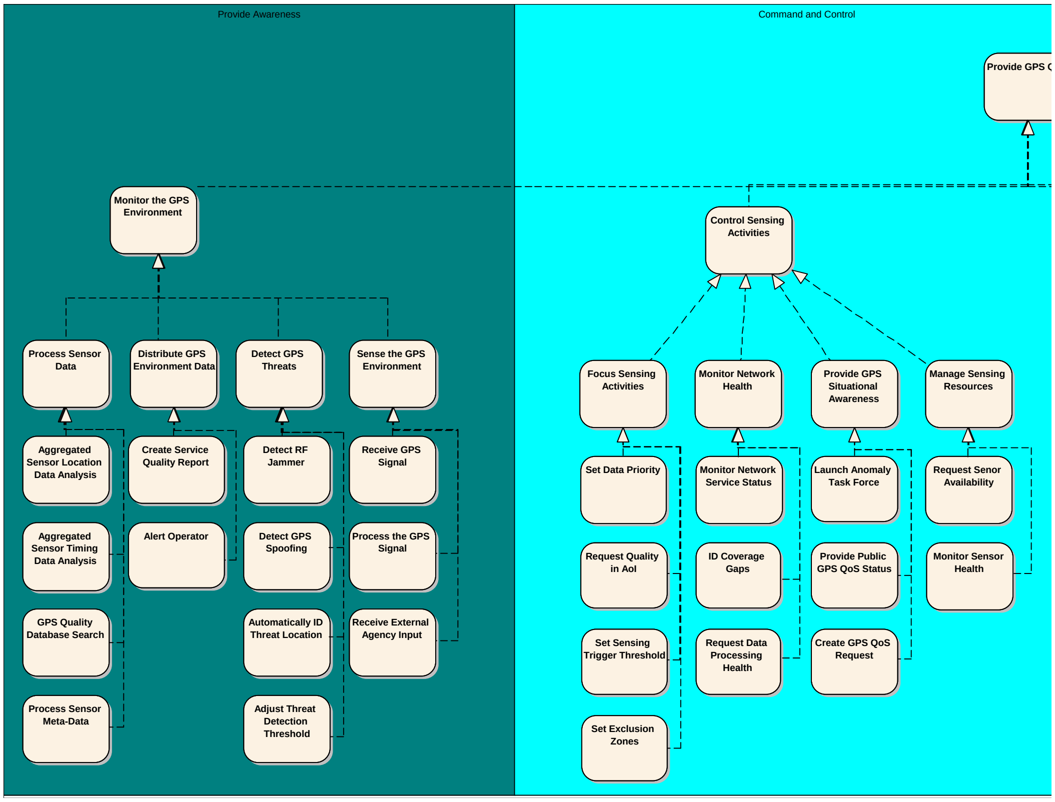

Provide Awareness
Command and Control
Provide GPS Q
Monitor the GPS Environment
Control Sensing Activities
Process Sensor Data
Distribute GPS Environment Data
Detect GPS Threats
Sense the GPS Environment
Focus Sensing Activities
Monitor Network Health
Provide GPS Situational Awareness
Manage Sensing Resources
Aggregated Sensor Location Data Analysis
Create Service Quality Report
Detect RF Jammer
Receive GPS Signal
Set Data Priority
Monitor Network Service Status
Launch Anomaly Task Force
Request Senor Availability
Aggregated Sensor Timing Data Analysis
Alert Operator
Detect GPS Spoofing
Process the GPS Signal
Request Quality in AoI
ID Coverage Gaps
Provide Public GPS QoS Status
Monitor Sensor Health
GPS Quality Database Search
Automatically ID Threat Location
Receive External Agency Input
Set Sensing Trigger Threshold
Request Data Processing Health
Create GPS QoS Request
Process Sensor Meta-Data
Adjust Threat Detection Threshold
Set Exclusion Zones

Figure 20. MCS SOA Operational Activity Diagram (OV-5a)

The next set of architecture products describe the service interactions of the MCS architecture. The scope of these products focuses only on what is necessary to describe the MCS M&S effort. This is done to explore in more detail the primary operational

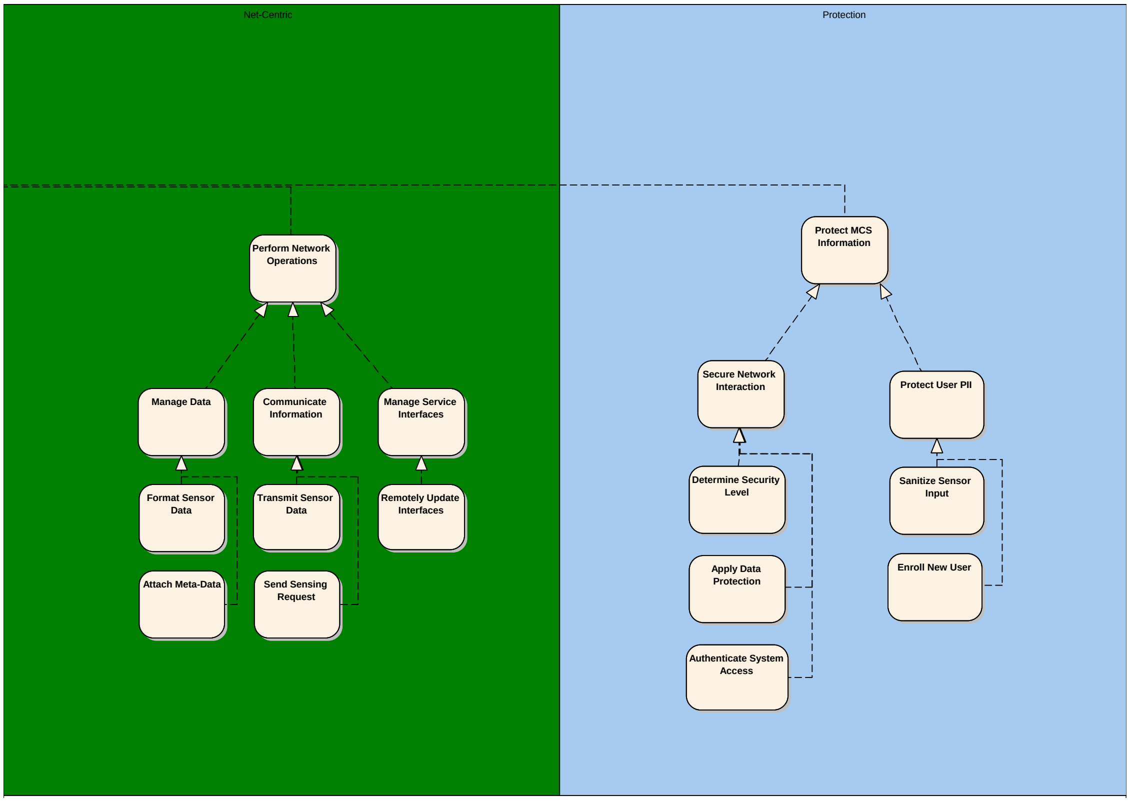
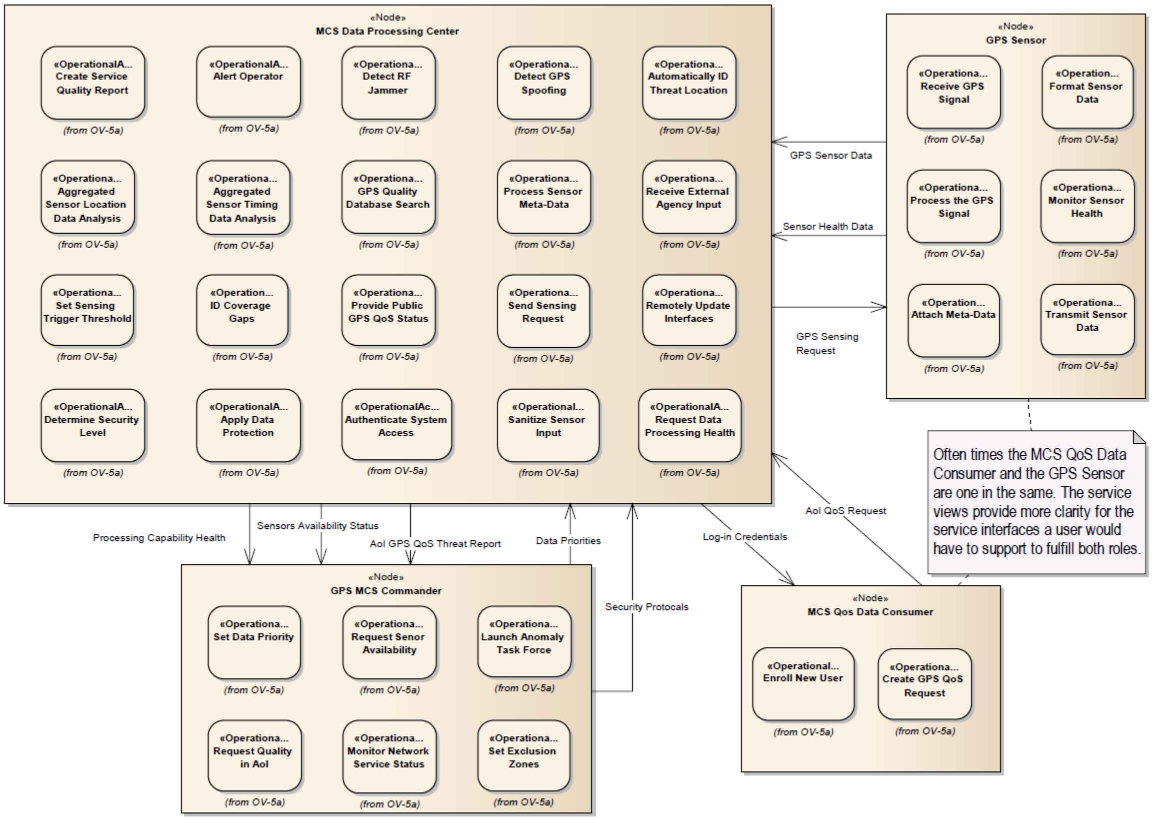


Figure 21. MCS SOA Operational Activity Diagram (OV-5a)

activities and services of the MCS architecture that govern its overall effectiveness.

The lower level service architecture products are developed utilizing the SoaML guidance as indicated in chapter III. SoaML, similar to the DoDAF process described above, is a capability based modeling language. SoaML defines capabilities as "Capabilities represent an abstraction of the ability to affect change [32]." This SoaML definition aligns with the DoDAF tenants to not predetermine a system to fulfill an mission need before developing the set of abstract capabilities. Additionally the SoaML specification states, "Each Capability may have owned behaviors that are methods of its provided Operations." Operational activities derived in the OV-5a are used to describe the operations required to fulfill the requirements of the SoaML ServiceContracts. The assignment of operational activities to different participants that fulfill the MCS ServiceContracts is in Figure 23

Before the operational activities could be assigned to a ServiceContract the operational activities were allocated to participants through the development of the OV-2. These nodes form the basis for the participants of the SoaML architecture products. The OV-2, presented below in Figure 22, shows at a high level the MCS architecture's participants, high level resource interactions, and participant's assigned operational activities. The interactions of the participant nodes in the OV-2 focus on "what" and not the precise "how" each participant will execute the operational activities assigned to it [32]. The lower levels of the MCS architecture implementation for each participant can be further refined once the relationships and influential factors between services is more clearly defined.


«Node»
MCS Data Processing Center
«OperationalA... Create Service Quality Report (from OV-5a)
«OperationalA... Alert Operator (from OV-5a)
«Operationa... Detect RF Jammer (from OV-5a)
«Operationa... Detect GPS Spoofing (from OV-5a)
«Operationa... Automatically ID Threat Location (from OV-5a)
«Operationa... Aggregated Sensor Location Data Analysis (from OV-5a)
«Operationa... Aggregated Sensor Timing Data Analysis (from OV-5a)
«Operationa... GPS Quality Database Search (from OV-5a)
«Operationa... Process Sensor Meta-Data (from OV-5a)
«Operationa... Receive External Agency Input (from OV-5a)
«Operationa... Set Sensing Trigger Threshold (from OV-5a)
«Operation... ID Coverage Gaps (from OV-5a)
«Operationa... Provide Public GPS QoS Status (from OV-5a)
«Operationa... Send Sensing Request (from OV-5a)
«Operationa... Remotely Update Interfaces (from OV-5a)
«OperationalA... Determine Security Level (from OV-5a)
«OperationalA... Apply Data Protection (from OV-5a)
«OperationalAc... Authenticate System Access (from OV-5a)
«Operational... Sanitize Sensor Input (from OV-5a)
«OperationalA... Request Data Processing Health (from OV-5a)
«Node»
GPS Sensor
«Operationa... Receive GPS Signal (from OV-5a)
«Operation... Format Sensor Data (from OV-5a)
«Operationa... Process the GPS Signal (from OV-5a)
«Operationa... Monitor Sensor Health (from OV-5a)
«Operation... Attach Meta-Data (from OV-5a)
«Operationa... Transmit Sensor Data (from OV-5a)
GPS Sensor Data
Sensor Health Data
GPS Sensing Request
Processing Capability Health
Sensors Availability Status
AoI GPS QoS Threat Report
Data Priorities
Security Protocals
Log-in Credentials
AoI QoS Request
«Node»
GPS MCS Commander
«Operationa... Set Data Priority (from OV-5a)
«Operationa... Request Senor Availability (from OV-5a)
«Operationa... Launch Anomaly Task Force (from OV-5a)
«Operationa... Request Quality in AoI (from OV-5a)
«Operationa... Monitor Network Service Status (from OV-5a)
«Operationa... Set Exclusion Zones (from OV-5a)
«Node»
MCS Qos Data Consumer
«Operational... Enroll New User (from OV-5a)
«Operationa... Create GPS QoS Request (from OV-5a)

Often times the MCS QoS Data Consumer and the GPS Sensor are one in the same. The service views provide more clarity for the service interfaces a user would have to support to fulfill both roles.

Figure 22. MCS SOA Operational Resource Flow Description (OV-2)

The SoaML service views are derived from the participant nodes and assigned operational activities defined in the OV-2. Unlike DoDAF, the SoaML specification does not delineate separate views or layers. Instead, SoaML defines the possible interactions between participants, and then assigns these interfaces and roles in an aggregated participant view. For the purposes of the MCS architecture the SoaML ServiceArchitecture product was developed to provide a high level view of the participant interactions. The MCS ServiceArchitecture is defined in Figure 23. As indicated above, the service views are only used to describe the services captured in the MCS simulation. Additional service interactions would be required to capture all the possible roles and exchanges of the architecture participants. The service architecture captures the full set of participants, their roles within each service contract or interface, and highlights the set of service contracts that are fulfilled within the M&S

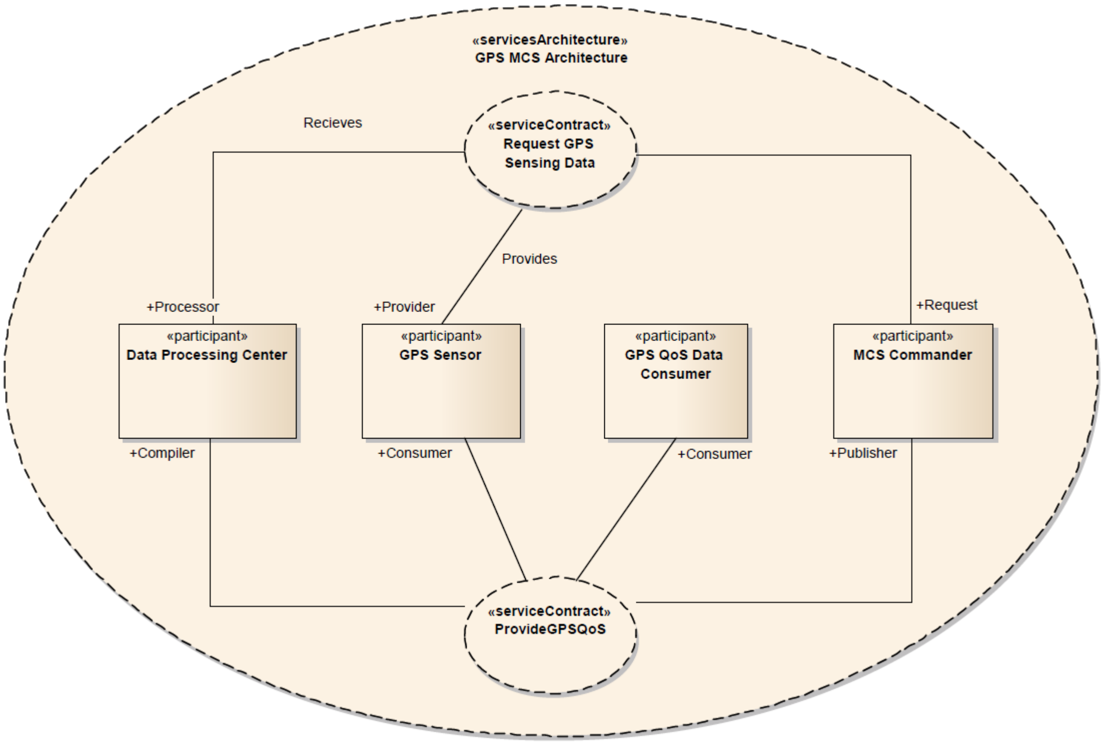


Figure 23. GPS MCS Service Architecture

activities. The ServiceArchitecture product is used to provide context for how services within an enterprise work and highlight dependencies that exist between participants [32].

Once the high level relationships are understood the more formal process for exchanging information or goods can be identified via the development of the service contracts. The contracts for “Request GPS Sensing Data” and “Provide GPS QoS” focus on defining what interfaces need to be exposed and what data needs to flow between participants to enable the MCS capability. It is important to notice that the roles identified in the service contract are agnostic of the participants. Clear definition of the interfaces allows for new participants to enter into the SOA and assume these defined roles provided they accommodate the interface specifications identified within the ServiceContracts. For the baseline GPS MCS architecture developed as a part of this research these interfaces are applied to the identified participant nodes. Figure 24 below shows the service contract for ”Request GPS Sensing Data” which includes the operations that support the defined interface, and depicts the data types that are required as inputs during a service exchanged. Most of the data types are basic data structures, but in some instances a notional custom data object is identified. Custom data objects are required as some of the data exchanges could require exchange of unique data types such as database or a formatted message containing a mixture of basic data types for easy integration into the GIS processing application. Figure 25 depicts the service contract for the publishing and request of the GPS MCS data.

The next MCS SoaML products are used to describe the timing and flow of the service identified within the service contract views. At its core SoaML is an extension of the Unified Modeling Language (UML) which facilitates the use of traditional UML sequence diagrams to depict the flow of the ServiceContracts. Figure 26 and 27 capture the order of message exchanges between roles of the request GPS sensing

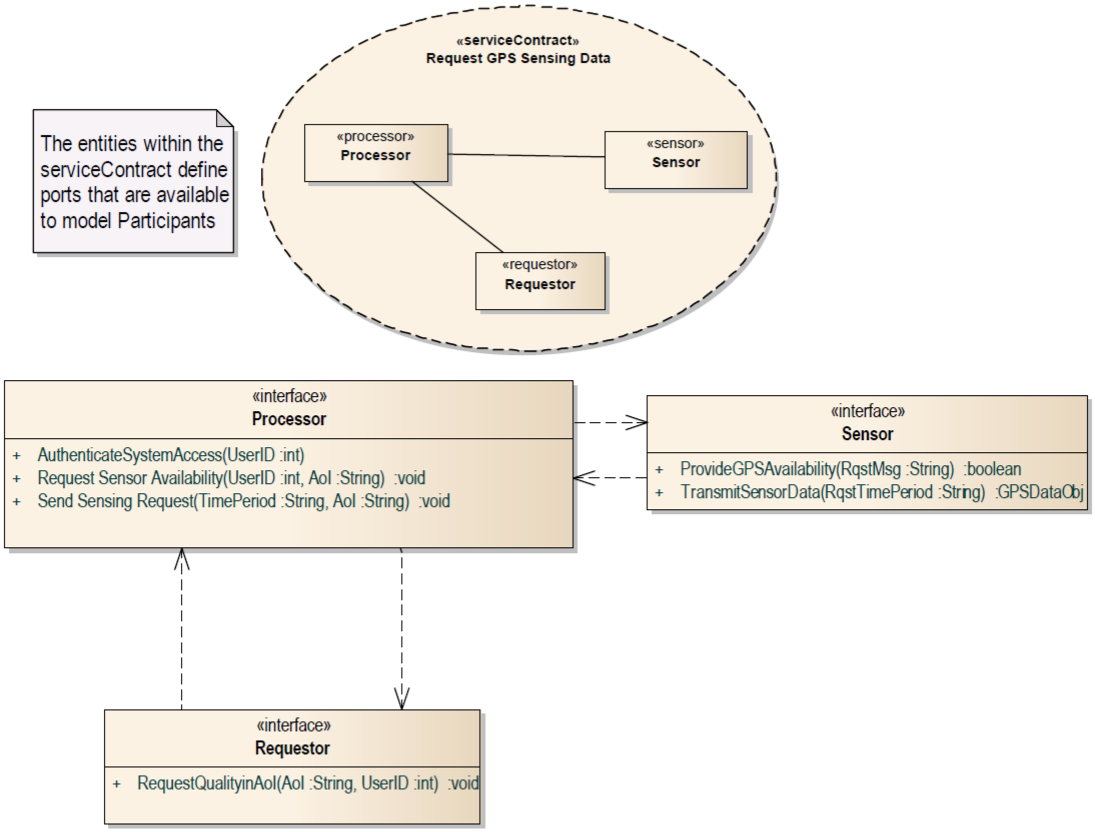

«serviceContract»
Request GPS Sensing Data
The entities within the serviceContract define ports that are available to model Participants
«processor»
Processor
«sensor»
Sensor
«requestor»
Requestor
«interface»
Processor
+ AuthenticateSystemAccess(UserID :int)
+ Request Sensor Availability(UserID :int, AoI :String) :void
+ Send Sensing Request(TimePeriod :String, AoI :String) :void
«interface»
Sensor
+ ProvideGPSAvailability(RqstMsg :String) :boolean
+ TransmitSensorData(RqstTimePeriod :String) :GPSDataObj
«interface»
Requestor
+ RequestQualityinAoI(AoI :String, UserID :int) :void

Figure 24. Service Contract: Request GPS Sensing Data

data and processing sensing data respectively. Not included within the sequence diagram are the "ownedBehaviors" of the participants that are activated once the exchange messages are received. The sequence diagrams are only intended to capture the exchanges of information and messages. These sequence diagrams also capture the flow of the MCS architecture's M&S activities. Key "ownedBehaviors" that are called during the course of fulfilling the service requests are included in SoaML participant views.

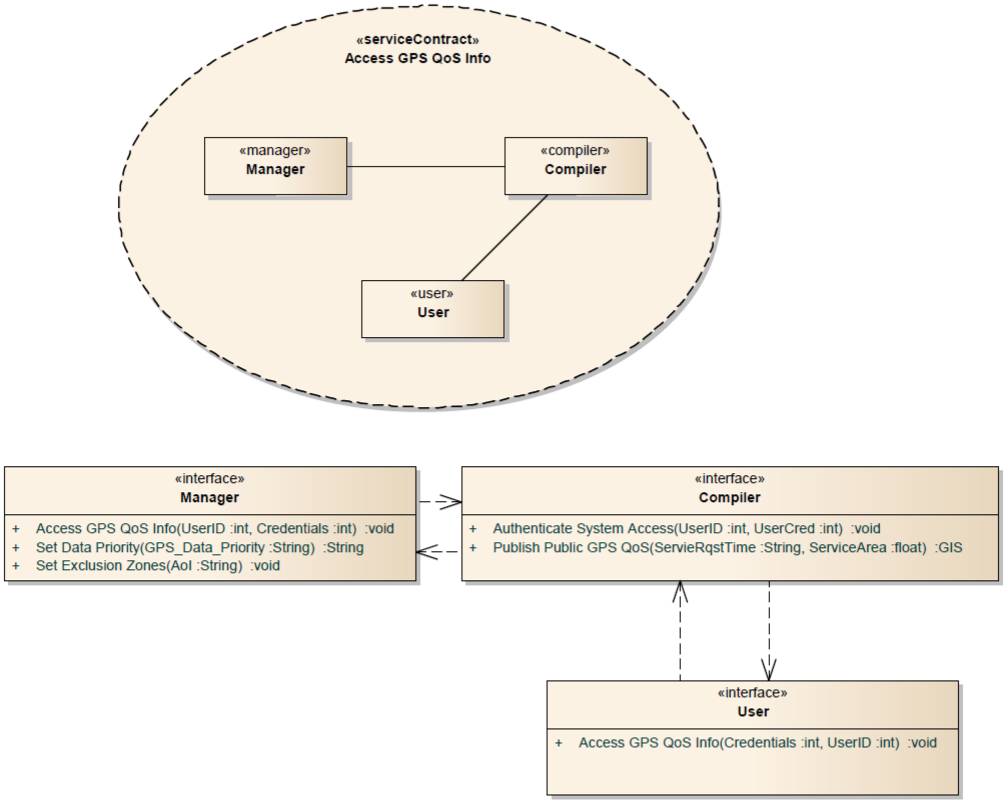

«serviceContract»
Access GPS QoS Info
«manager»
Manager
«compiler»
Compiler
«user»
User
«interface»
Manager
+ Access GPS QoS Info(UserID :int, Credentials :int) :void
+ Set Data Priority(GPS_Data_Priority :String) :String
+ Set Exclusion Zones(AoI :String) :void
«interface»
Compiler
+ Authenticate System Access(UserID :int, UserCred :int) :void
+ Publish Public GPS QoS(ServieRqstTime :String, ServiceArea :float) :GIS
«interface»
User
+ Access GPS QoS Info(Credentials :int, UserID :int) :void

Figure 25. Service Contract: Provide GPS QoS

Participants within SoaML may be considered abstract or concrete [32]. If additional research is explored to continue the development of the MCS architecture representative, hardware and software solutions could be used to provide a more concrete instantiation of the architecture. However, for this research the participants are considered abstract to focus on understanding the relationships and influences of the notional architecture. The participant set is derived from the OV-2 nodes and each are assigned service ports that correspond to the service contract roles depicted above. While SoaML does not encourage the development of how each participant fulfills a service request, as this detracts from the SOA principle of flexibility and low coupling, it does allow for the definition of ”ownedBehaviors” of architecture participants. These owned behaviors are the methods accessed when a service request is

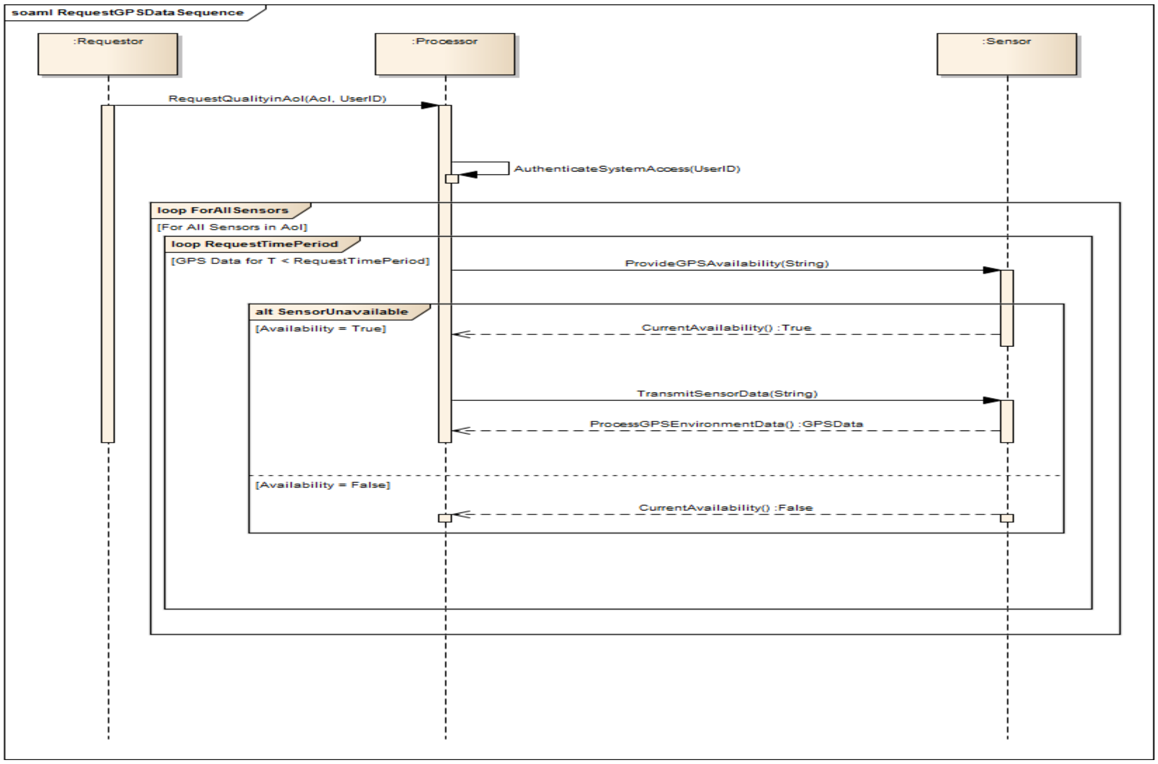

soaml RequestGPSDataSequence
:Requestor
:Processor
:Sensor
RequestQualityinAoI(AoI, UserID)
AuthenticateSystemAccess(UserID)
loop ForAllSensors
[For All Sensors in AoI]
loop RequestTimePeriod
[GPS Data for T < RequestTimePeriod]
ProvideGPSAvailability(String)
alt SensorUnavailable
[Availability = True]
CurrentAvailability() :True
TransmitSensorData(String)
ProcessGPSEnvironmentData() :GPSData
[Availability = False]
CurrentAvailability() :False

Figure 26. Service Contract Sequence Diagram: Request GPS Sensing Data

received that may not be exposed directly as an external service port. Figure 28 and 29 are the participant diagrams that define the roles and behaviors of each participant.

These views link the MCS participants to the roles defined in the service contract views, and specifies the requirements that each participant would have to fulfill to fully implement a provided service. Again these services and behaviors only focus on what was required to model and simulate the service contracts identified above. Additional services and behaviors would be required to completely describe the interactions of the full MCS architecture. The most influential participant during the M&S activities was the data processing center. The data processing center's services and ownedBehaviors are explored in more detail to define the necessary actions the M&S activities must support. Of note there is no jamming participant. This is due to the fact that a jammer is not an architecture participant, but instead influences

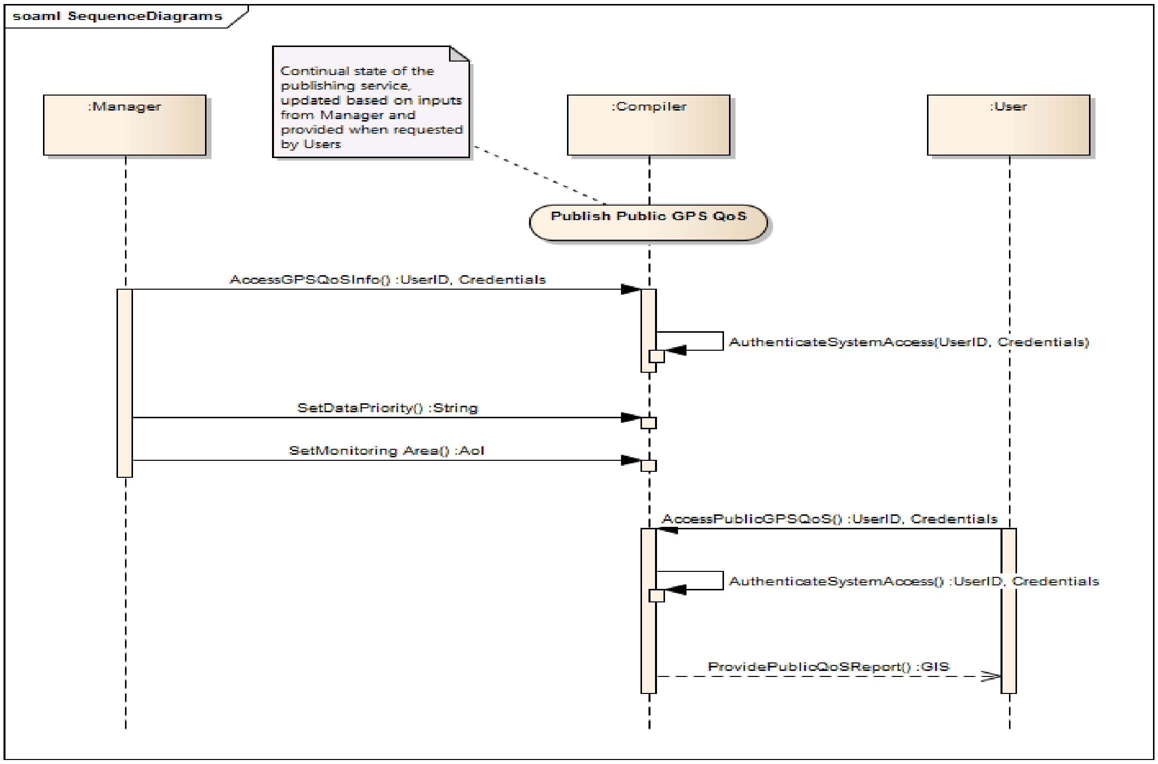

soaml SequenceDiagrams
Continual state of the publishing service, updated based on inputs from Manager and provided when requested by Users
:Manager
:Compiler
:User
Publish Public GPS QoS
AccessGPSQoSInfo() :UserID, Credentials
AuthenticateSystemAccess(UserID, Credentials)
SetDataPriority() :String
SetMonitoring Area() :AoI
AccessPublicGPSQoS() :UserID, Credentials
AuthenticateSystemAccess() :UserID, Credentials
ProvidePublicQoSReport() :GIS

Figure 27. Service Contract Sequence Diagram: Access GPS QoS Information

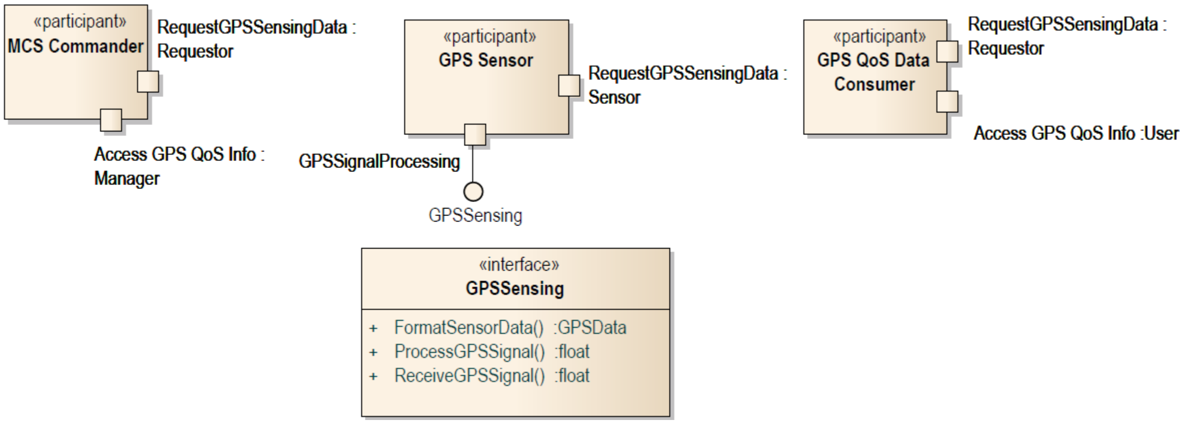

«participant»
MCS Commander
RequestGPSSensingData : Requestor
Access GPS QoS Info : Manager
«participant»
GPS Sensor
RequestGPSSensingData : Sensor
GPSSignalProcessing
GPSSensing
«participant»
GPS QoS Data Consumer
RequestGPSSensingData : Requestor
Access GPS QoS Info :User
«interface»
GPSSensing
+ FormatSensorData() :GPSData
+ ProcessGPSSignal() :float
+ ReceiveGPSSignal() :float

Figure 28. Participant View for the MCS Commander, Sensor, and Service User

the environment the service operates within.

With the architecture defined to an appropriate level the MCS architecture can now be explored through the use of MBSE. The next section in this chapter focuses on the data that was produced by the M&S activities described in chapter III.

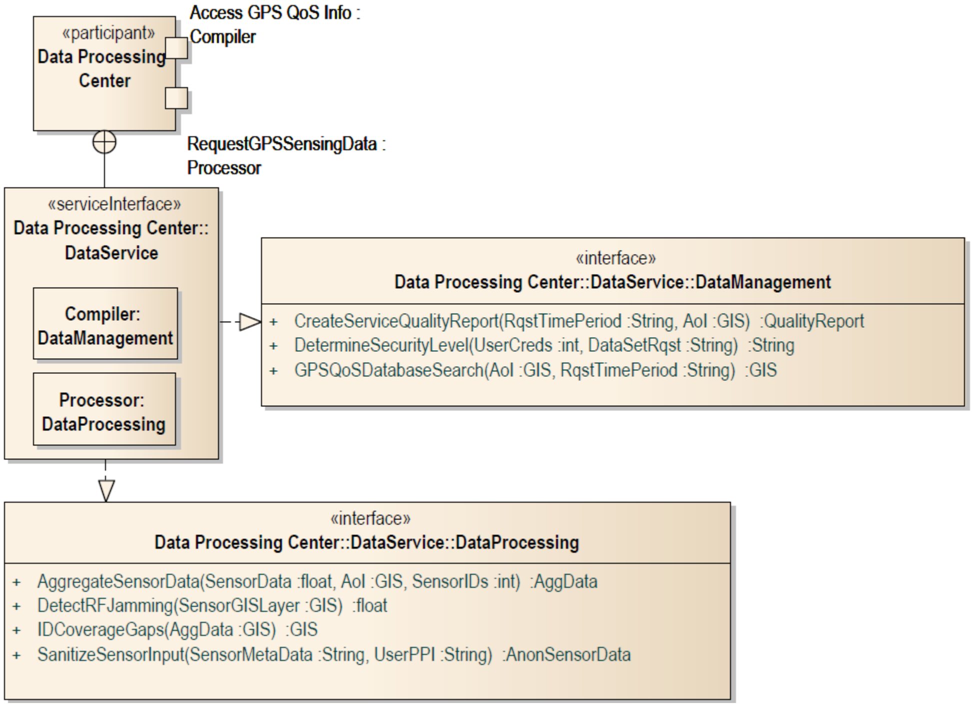

«participant»
Data Processing Center
Access GPS QoS Info : Compiler
RequestGPSSensingData : Processor
«serviceInterface»
Data Processing Center:: DataService
Compiler: DataManagement
Processor: DataProcessing
«interface»
Data Processing Center::DataService::DataManagement
+ CreateServiceQualityReport(RqstTimePeriod :String, AoI :GIS) :QualityReport
+ DetermineSecurityLevel(UserCreds :int, DataSetRqst :String) :String
+ GPSQoSDatabaseSearch(AoI :GIS, RqstTimePeriod :String) :GIS
«interface»
Data Processing Center::DataService::DataProcessing
+ AggregateSensorData(SensorData :float, AoI :GIS, SensorIDs :int) :AggData
+ DetectRFJamming(SensorGISLayer :GIS) :float
+ IDCoverageGaps(AggData :GIS) :GIS
+ SanitizeSensorInput(SensorMetaData :String, UserPPI :String) :AnonSensorData

Figure 29. Participant View for the Data Processing Center

4.2 Modeling and Simulation Results

The M&S results section focuses on the products developed within the ESRI ArcGIS toolkit. STK was used to provide the input parameters into ArcGIS through the python interface. The remainder of the document will focus on the products produced via the ArcGIS analysis, but it is worth noting again however that the development of the MCS architecture variants within STK was not trivial, and played a critical role throughout this research. The evaluation of the MCS architectures revolved around the analysis of the ArcGIS outputs from each of the architecture variants that were generated via varying the parameters identified in Table 2. This section includes the outputs from a subset of the total number of variants described in Table 2. The variants included in this section are those that provide unique insights into the MCS architecture interactions. The variants included in Appendix C either produced similar results to those mentioned below, or provided no significant

Table 3. GPS Jammer Parameters

Attribute	Value
Frequency	L1 (1.57542 GHz)
Transmitter Power	0.05 W
Antenna Pattern	Hemispherical
Jammer Velocity	18 mph

or reliable analysis.

Before describing the architecture variant analysis a brief description of the jammer information is required. The jammer location was extracted from STK as a function of the architecture variant's sampling time and is included in the Table 11. This table provides the information used as the truth data for the distance error calculation between the jammer's position and the estimated position. Table 3 below list the parameters that were identified in chapter III to define the jammer's performance across each architecture variant.

Each architecture variant analysis section includes a short description of each of the output data sets presented. The data artifacts presented are different for each variant, but every variant presented includes the ArcGIS optimized hot spot analysis output. Common to all the variants the optimized hot spot analysis is applied to the user set after the receivers were binned into a 16x16 tiled square grid. This grid aggregates the number of receivers into the sub-areas and the join count of receivers is the input attribute to the optimized hot spot analysis tool. The optimized hot spot analysis shows both statistically significant hot and cold spots, and includes a plotted point to mark the spatial mean of the cold spots. Lastly, the distance error for all the variants is aggregated into table 11 in Appendix C. Additionally the initial distribution of the receivers with no filtering is depicted in Appendix B The following analysis section focuses on a subset of the variants that are the most relevant to the

MCS feasibility analysis. Each variant includes a performance table that indicates the architecture's minimum, maximum, and average error when estimating the jammer's position. The first variant presented below is the 1000 uniformly distributed static receiver variant. This variant is presented first as it provides the best baseline estimate for the jammer's position, and it has the fewest analysis artifacts that resulted in anomalous behavior.

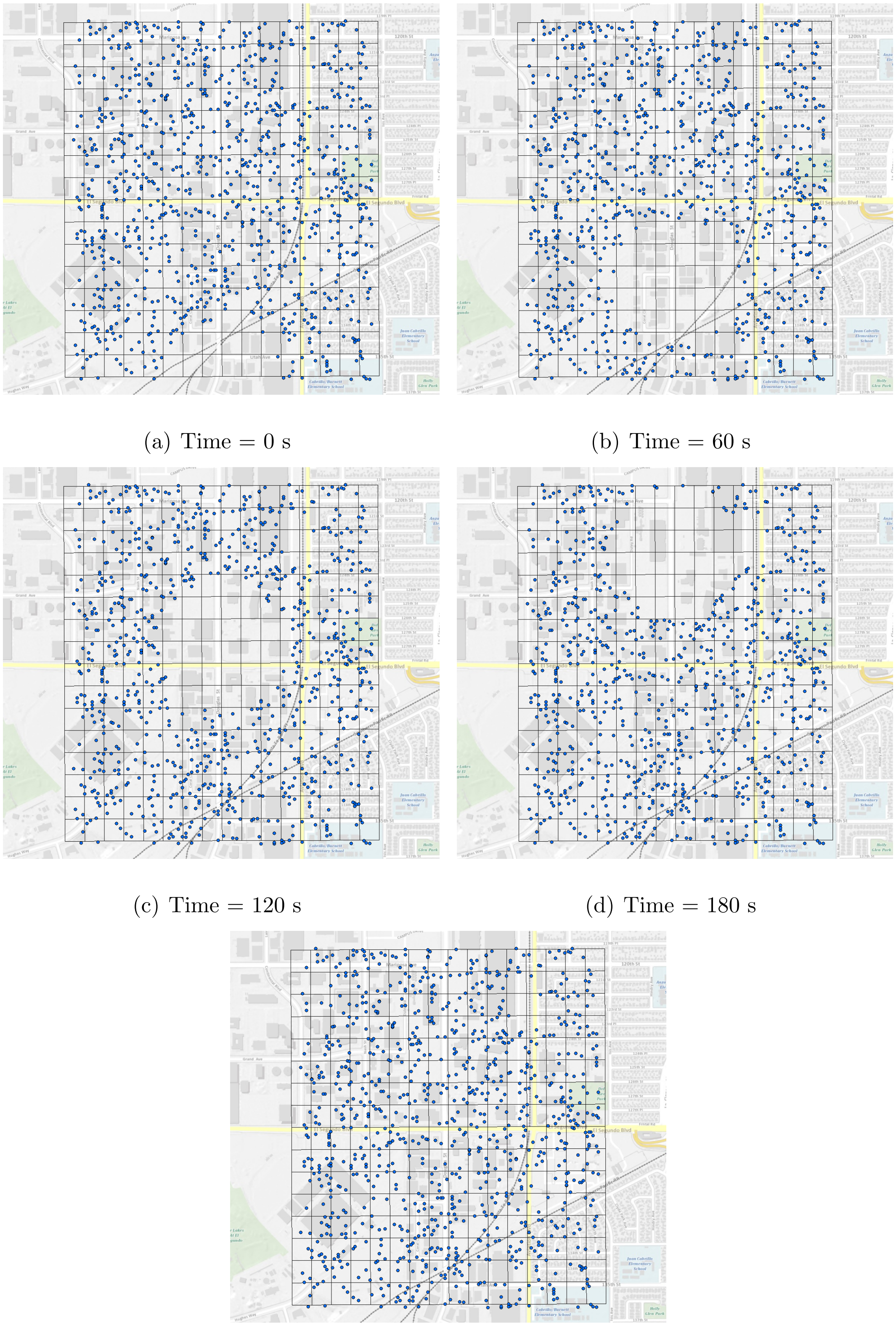
GPS Sensors: Uniform Distribution, 1000 Users, Static, Time Between Sensing 15s.

Table 4. 1000 uniformly distributed static receivers performance (m)

Minimum	Maximum	Average
13.72	121.95	46.23

As indicated above this variant provided a good baseline for the MCS architecture. The reported raw receiver locations for the on-the-minute samples are included in this section to provide context as to the inputs to the ArcGIS analysis process. The Figure set 30 is the raw receiver reported locations, and the Figure set 31 is the on-the-quarter minute sampled hot spot analysis output. The ArcGIS optimized hot spot tool is applied to the unfiltered receiver set to confirm that the receivers are in fact distributed in a uniform manner. The tool results for the unfiltered data set indicate no hot or cold spots, thus statistically validating the uniformity of the distribution.

A clear progression of a GPS disruption is clear from Figure set 30. The final figure in the set shows that the source of the disruption has left the area somewhere between the third and fourth sample time, and at the end of the simulation the distribution returns to the expected uniform distribution. Below in Figure set 31 is the optimized hot spot analysis results for sampling and processing the variant on-the-quarter minute. The estimated jammer location is denoted with an orange dot.



(a) Time = 0 s

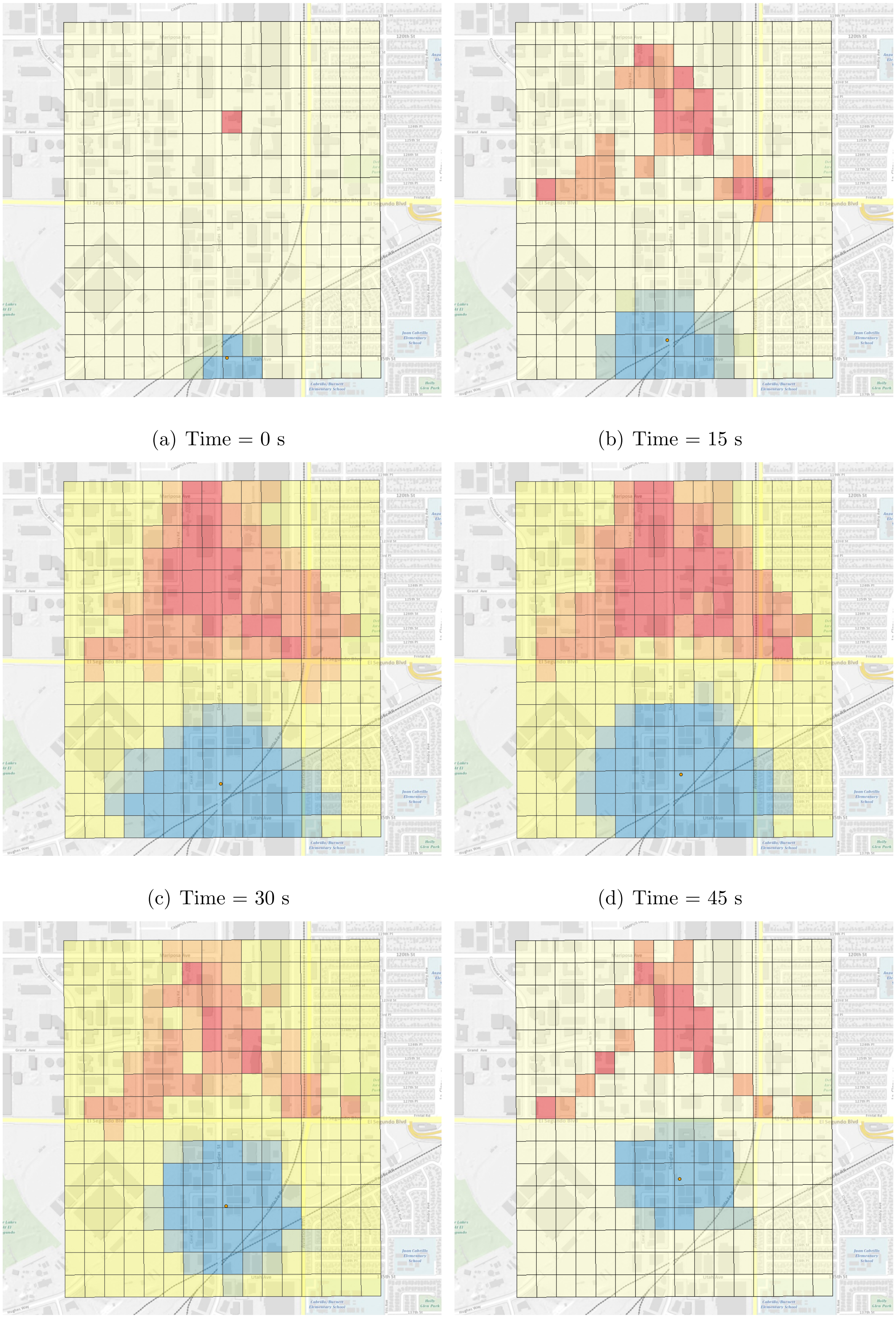
(b) Time = 60 s

(c) Time = 120 s

(d) Time = 180 s

(e) Time = 240 s

Figure 30. 1000 static uniformly distributed receivers: filtered out $(C/(No + Io) < 28dB$



(a) Time = 0 s

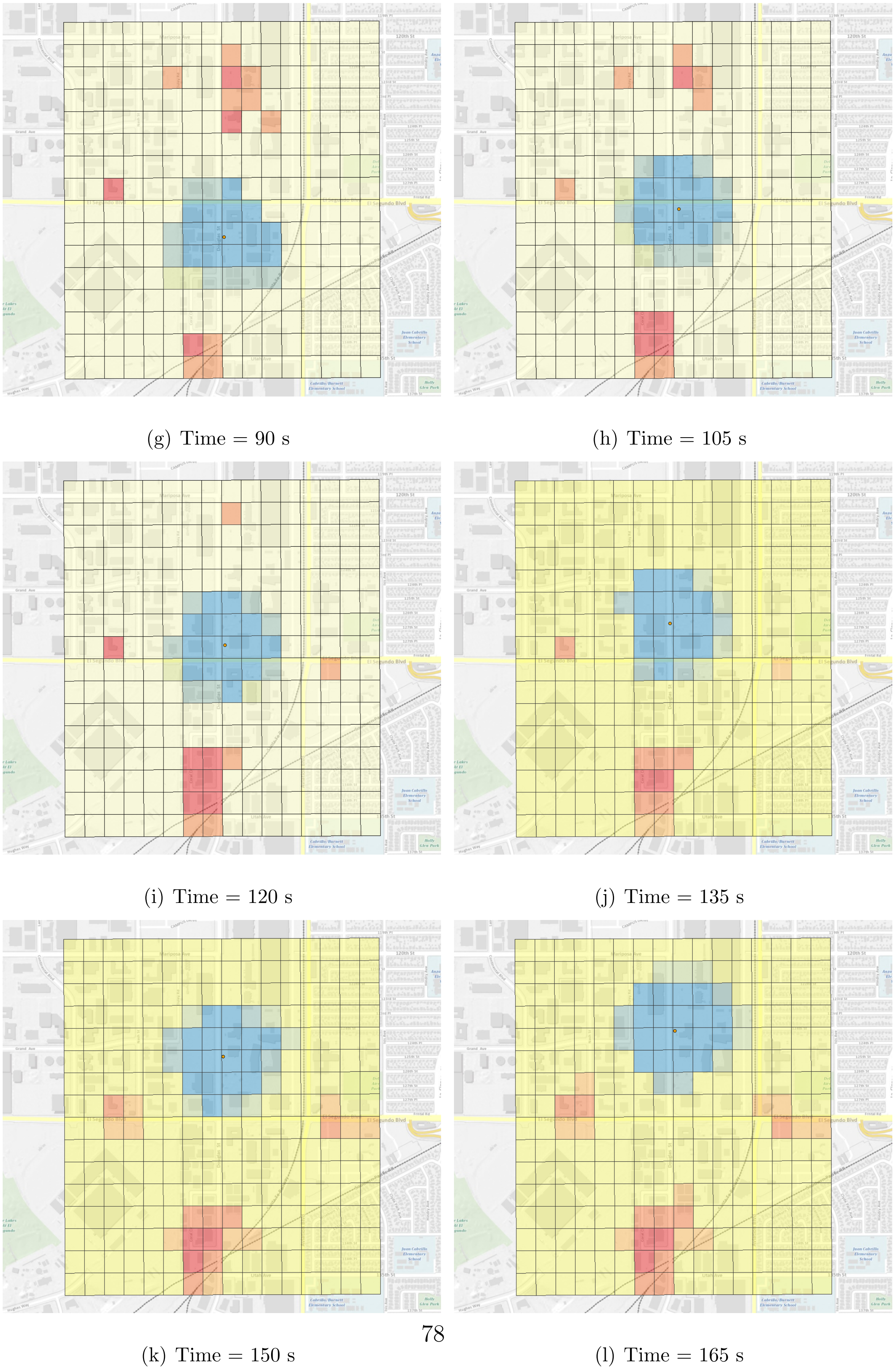
(b) Time = 15 s

(c) Time = 30 s

(d) Time = 45 s

(e) Time = 60 s

(f) Time = 75 s



(g) Time = 90 s

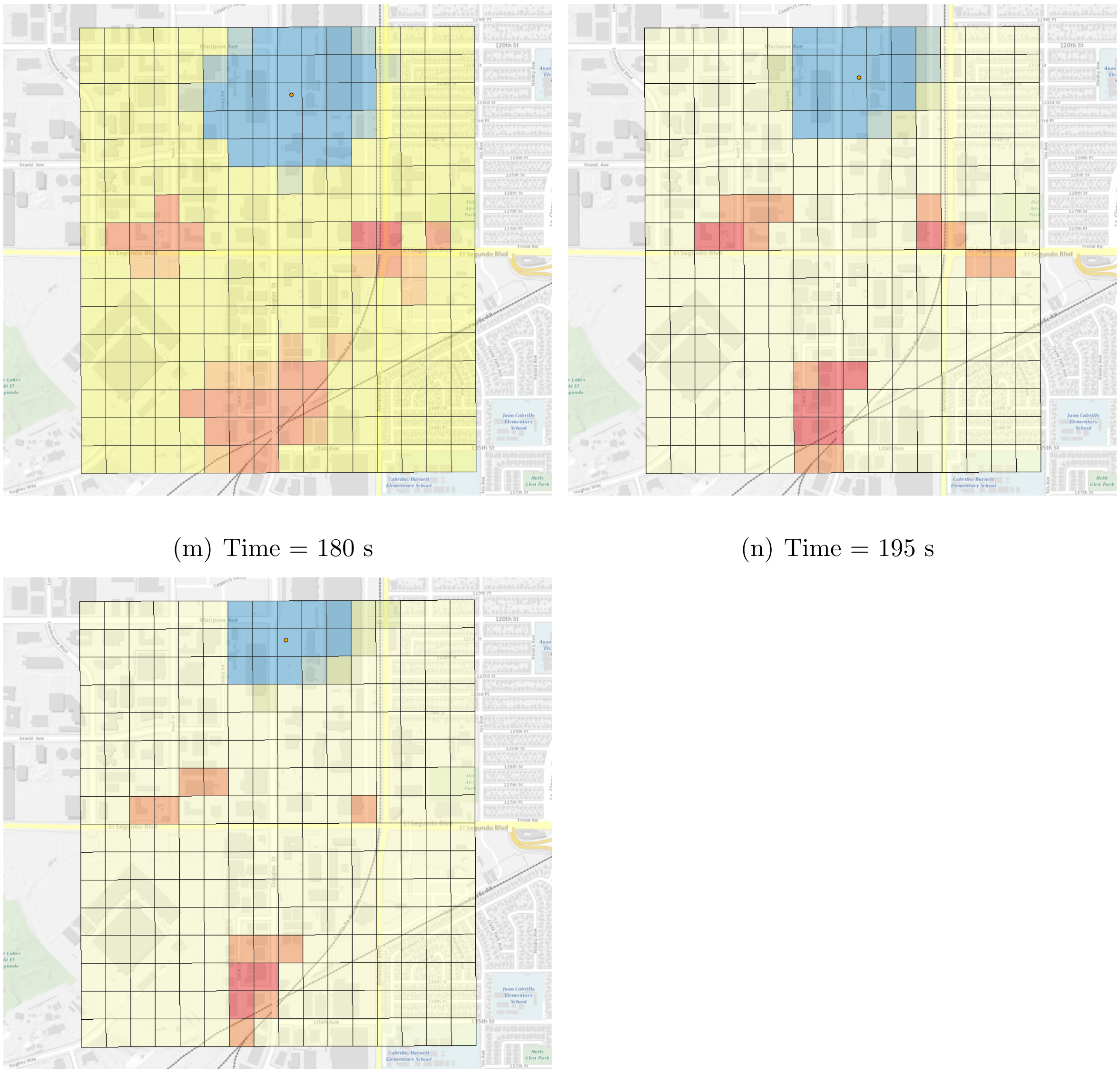
(h) Time = 105 s

(i) Time = 120 s

(j) Time = 135 s

(k) Time = 150 s

(l) Time = 165 s


El Segundo Blvd
120th St
Juan Cabrillo Elementary School
Cabrillo/Burnett Elementary School
Holly Glen Park
Utah Ave

(m) Time = 180 s

(n) Time = 195 s

(o) Time = 210 s

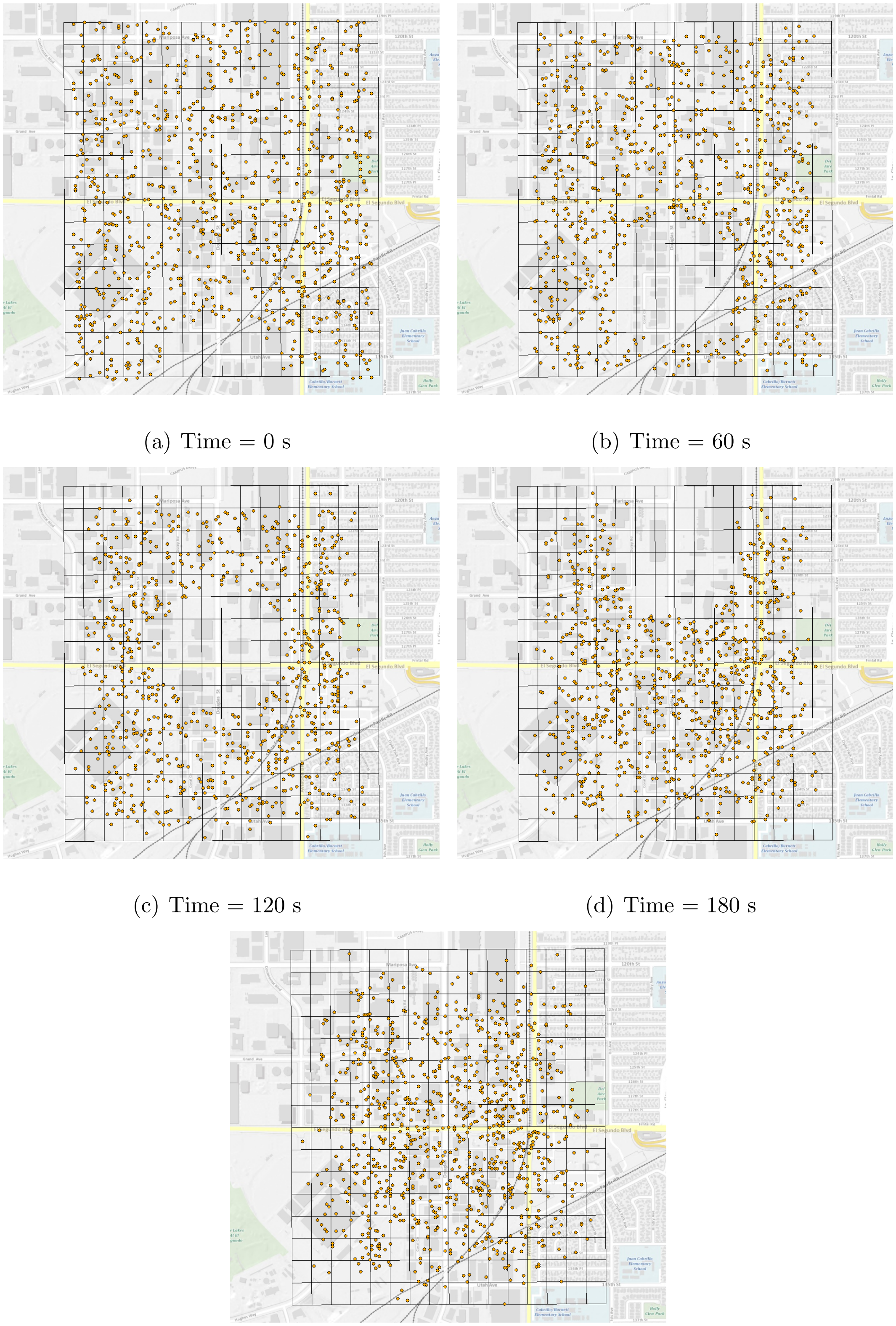
Figure 31. Optimized hot spot analysis for 1000 static uniformly distributed receivers

GPS Sensors: Uniform Distribution, 1000 Users, Mobile, Time Between Sensing 15s.

Table 5. 1000 uniformly distributed mobile receivers performance (m)

Minimum	Maximum	Average
5.86	320.94	83.31

This section focuses on the variant of the receivers starting in a uniform distribution, but the receivers are mobile during the scenario time period. Similar to the above variant this section includes three figure sets. Figure set 32 is the receivers sampled at each time step with filtering of the receivers who's $C/(No + Io)$ falls below the 28 dB/Hz threshold. Last is the set of figures depicting the optimized hot spot analysis sampled on-the-quarter minute. Analysis of this receiver set revealed as the receivers attempted to move between way points they began to distribute themselves into more of a normally distributed fashion. The optimized hot spot analysis figures end at 210 seconds. Analysis of the final 90 seconds revealed that the receivers maintained the normal distribution. So the 90 seconds after the jammer exits the AoI several false positives are reported for the estimated location of the jammer. The reconfiguration of the receiver distribution was not an expected behavior for this scenario. In this scenario the receivers are moving slowly, an average walking pace, and it was expected that they would move in a way that would preserve the uniform distribution as they traversed between the uniformly generated way points. A large take away from this data set is that there may be times during a day where the user set is distributed in an expected fashion, but during times of high number of transiting users a different set of assumptions may need to be applied during the analysis. The optimized hot spot figures encompass the on the minute sampling optimized hot spot analysis which is why there is not a dedicated subsection for the slower sampling rate variant.



(a) Time = 0 s

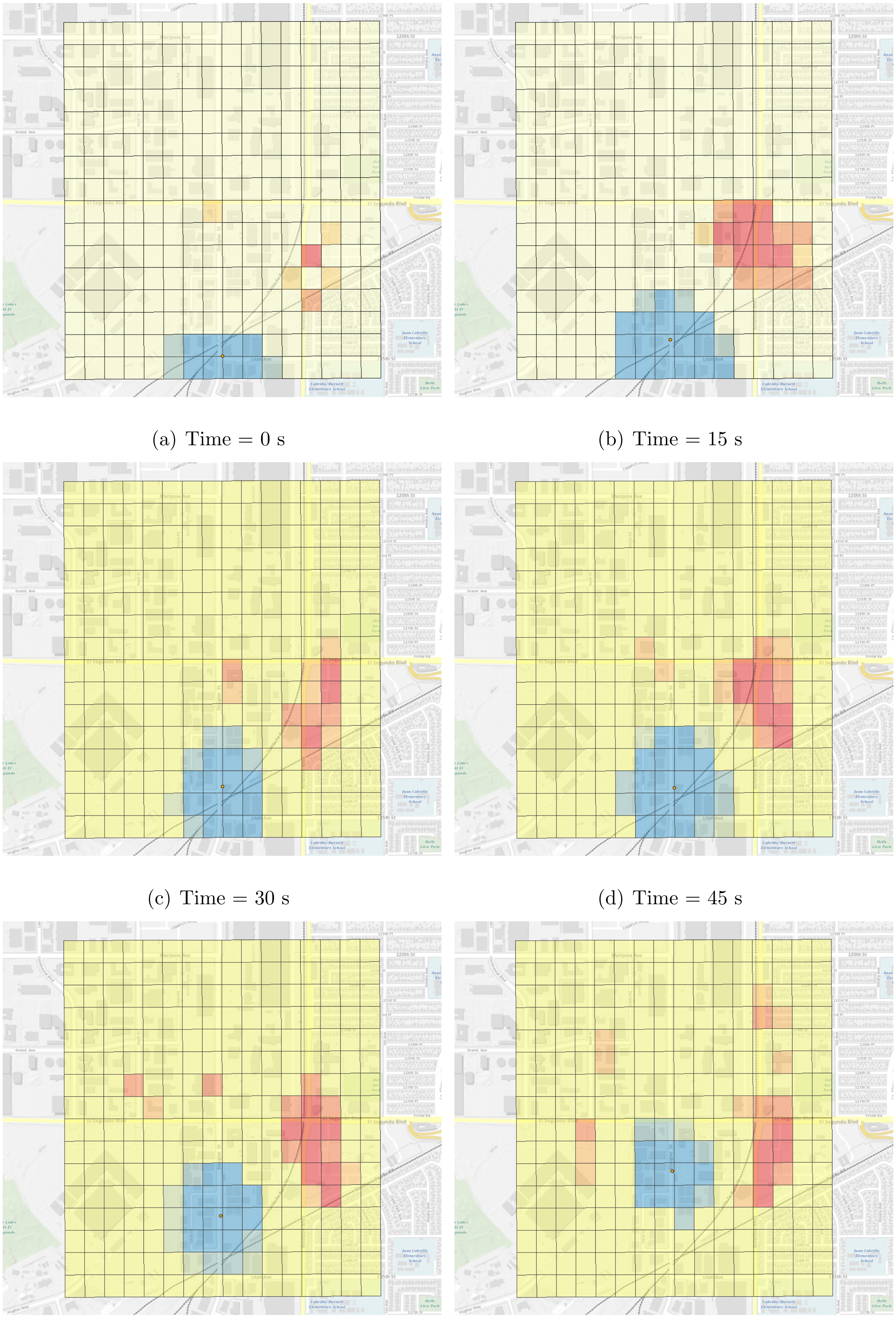
(b) Time = 60 s

(c) Time = 120 s

(d) Time = 180 s

(e) Time = 240 s

Figure 32. 1000 mobile uniformly distributed receivers: filtered out $(C/(No+Io) < 28dB$



(a) Time = 0 s

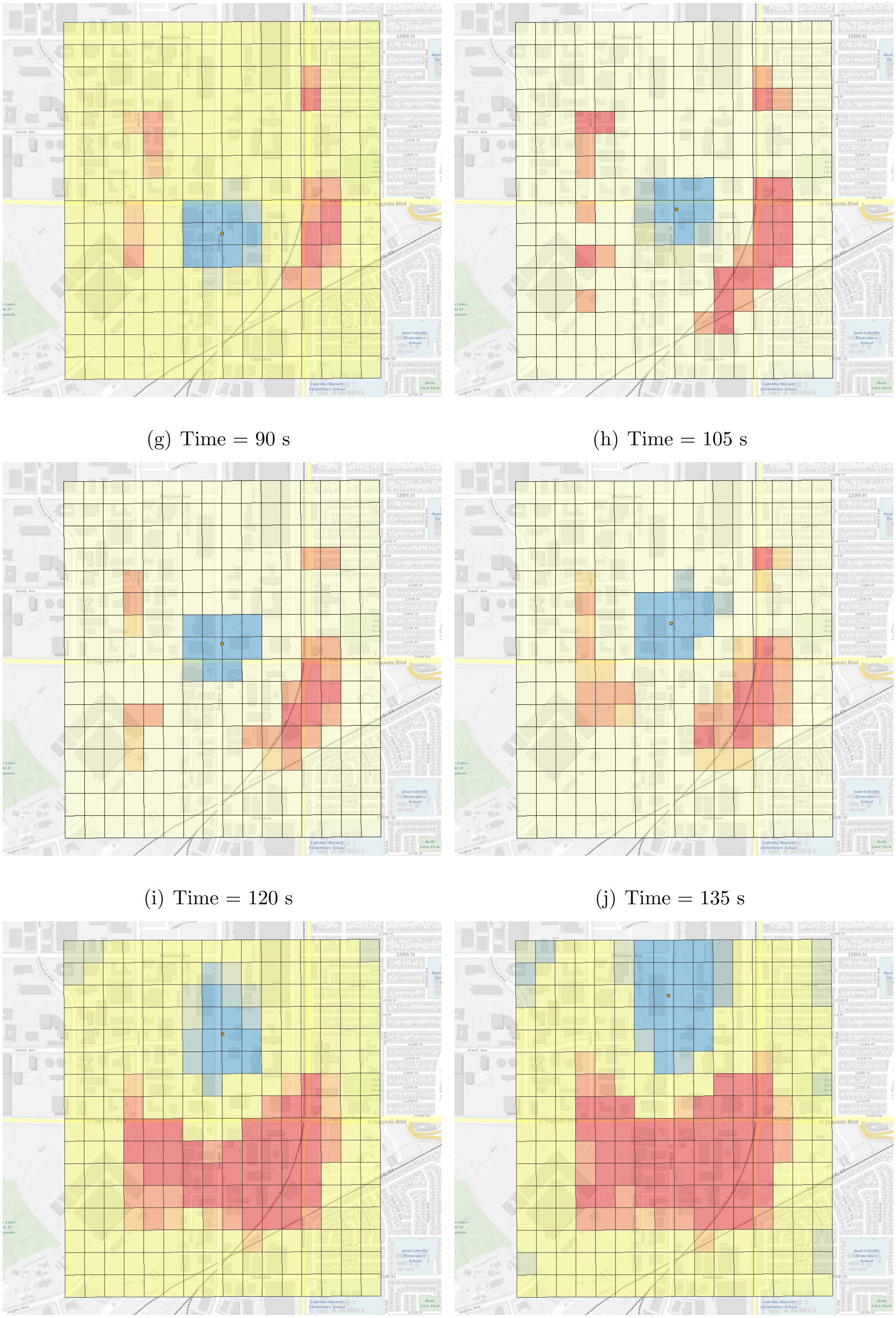
(b) Time = 15 s

(c) Time = 30 s

(d) Time = 45 s

(e) Time = 60 s

(f) Time = 75 s



(g) Time = 90 s

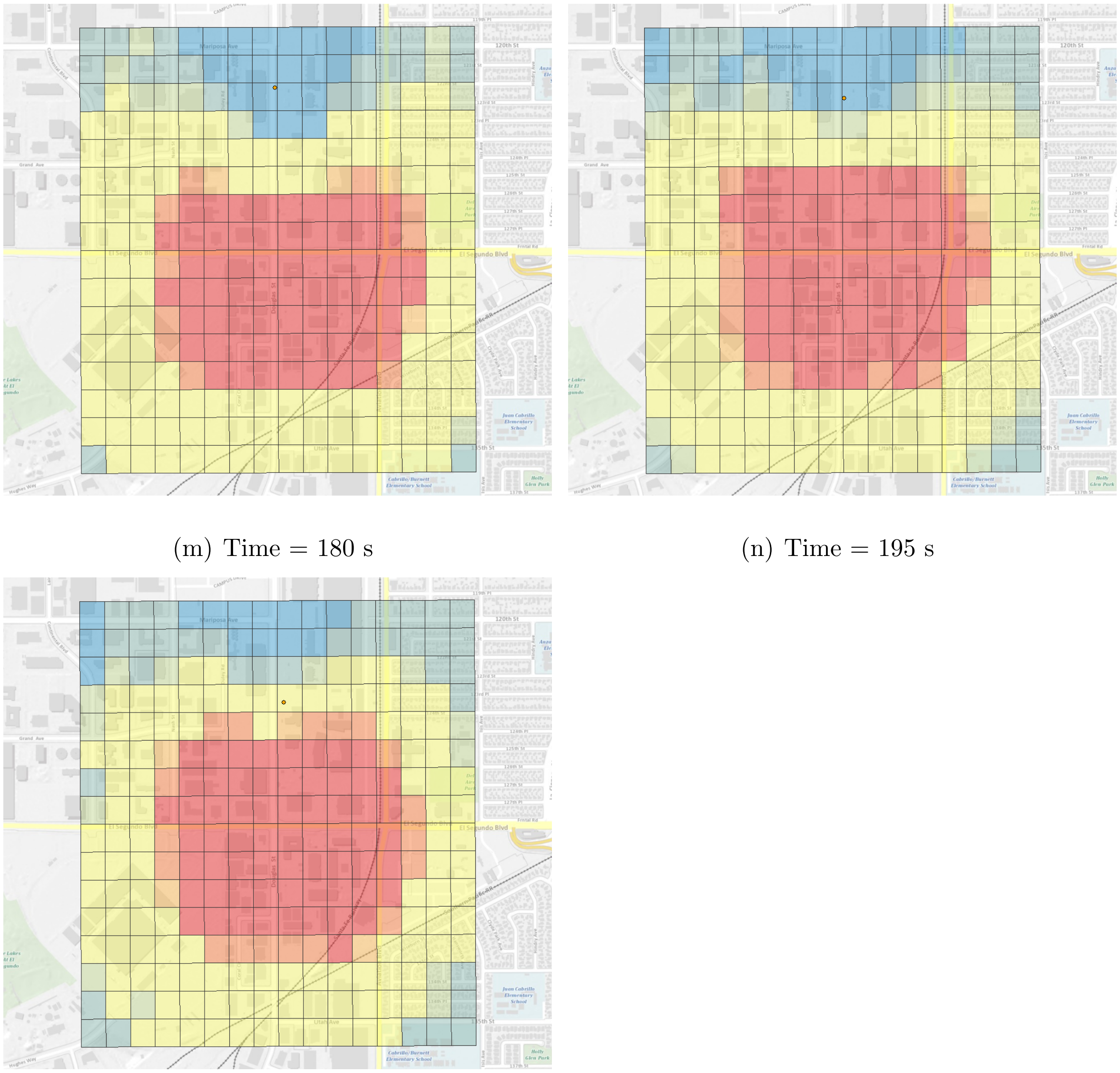
(h) Time = 105 s

(i) Time = 120 s

(j) Time = 135 s

(k) Time = 150 s

(l) Time = 165 s



(m) Time = 180 s

(n) Time = 195 s

(o) Time = 210 s

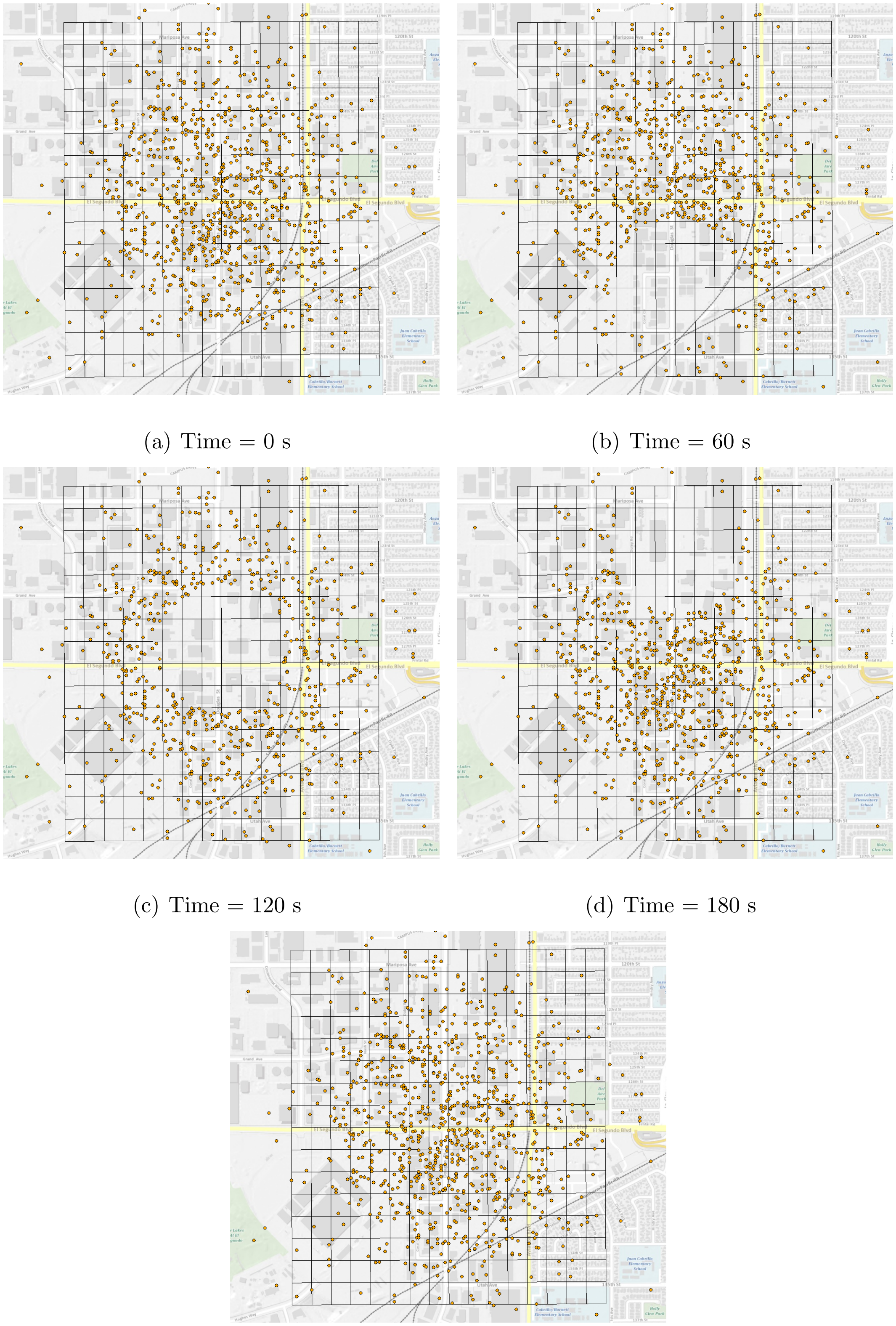
Figure 33. Optimized hot spot analysis for 1000 uniformly distributed mobile receivers

GPS Sensors: Normal Distribution, 1000 Users, Static, Time Between Sensing 15s.

Table 6. 1000 normally distributed static receivers performance (m)

Minimum	Maximum	Average
37.91	776.57	286.13

This section considers a similar data presentation flow for a set of normally distributed users. The on-the-minute filtered raw receiver data is included to provide context for what a data processing center might receive as inputs to the geo-statistics processing engine. Lastly the optimized hot spot analysis tool is applied based on-the-quarter minute sampling rate. The estimated jammer location is marked by a teal point feature. This data set encompasses the on-the-minute samples which is why there is not a dedicated subsection for the on-the-minute sampling rate. This data set results in a larger errors when estimating the jammer's location. When a jammer is not present, the normal distribution of the receivers generates a false positive of the jammer's location due to the lack of receivers in the outer subareas of the AoI. When the jammer is interacting with the users in the central area of the distribution the estimate becomes more balanced, but there are still influences on the estimated position by the cold spots at the fringes of the area.



(a) Time = 0 s

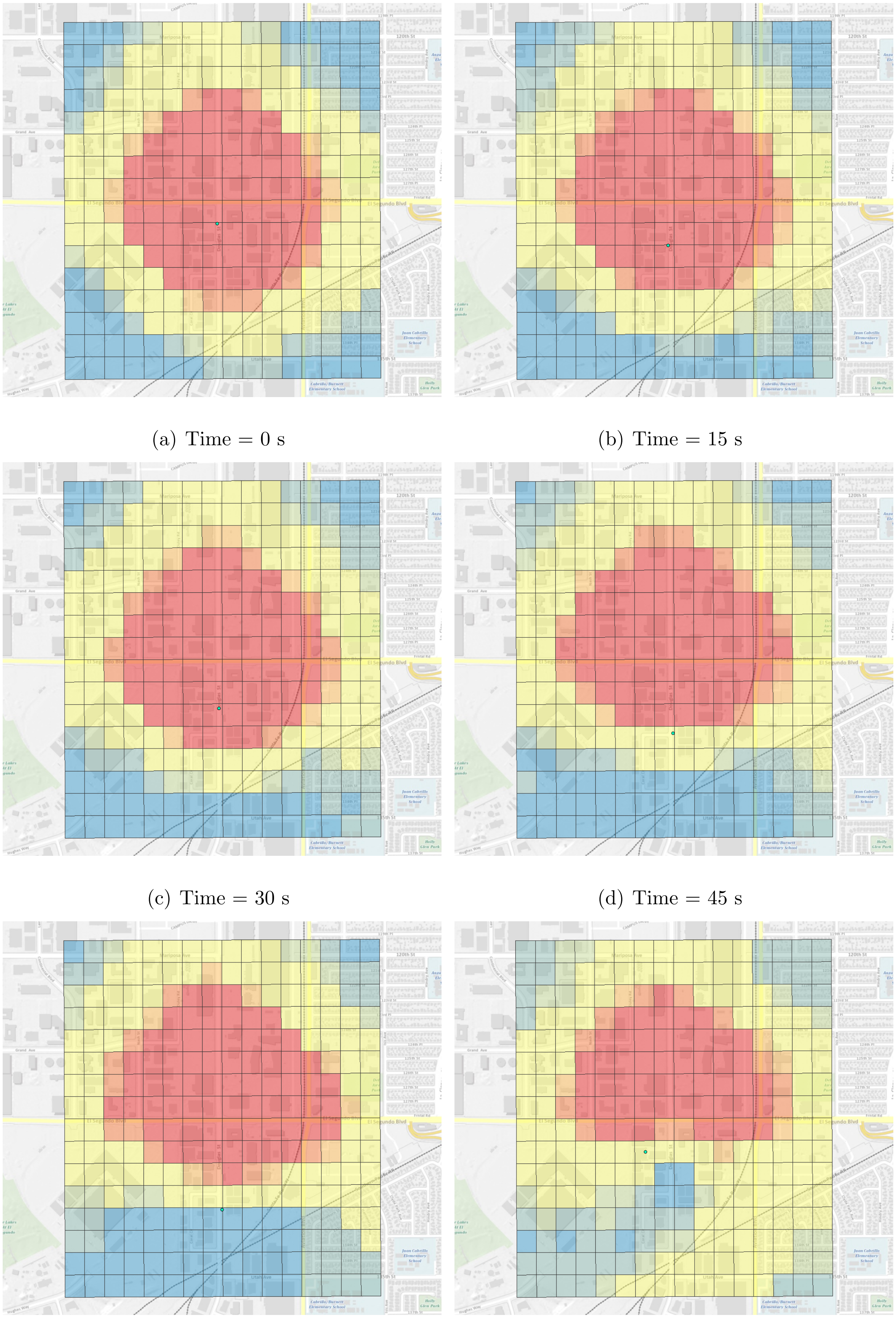
(b) Time = 60 s

(c) Time = 120 s

(d) Time = 180 s

(e) Time = 240 s

Figure 34. 1000 static normally distributed receivers: filtered out $(C/(No + Io) < 28dB$



(a) Time = 0 s

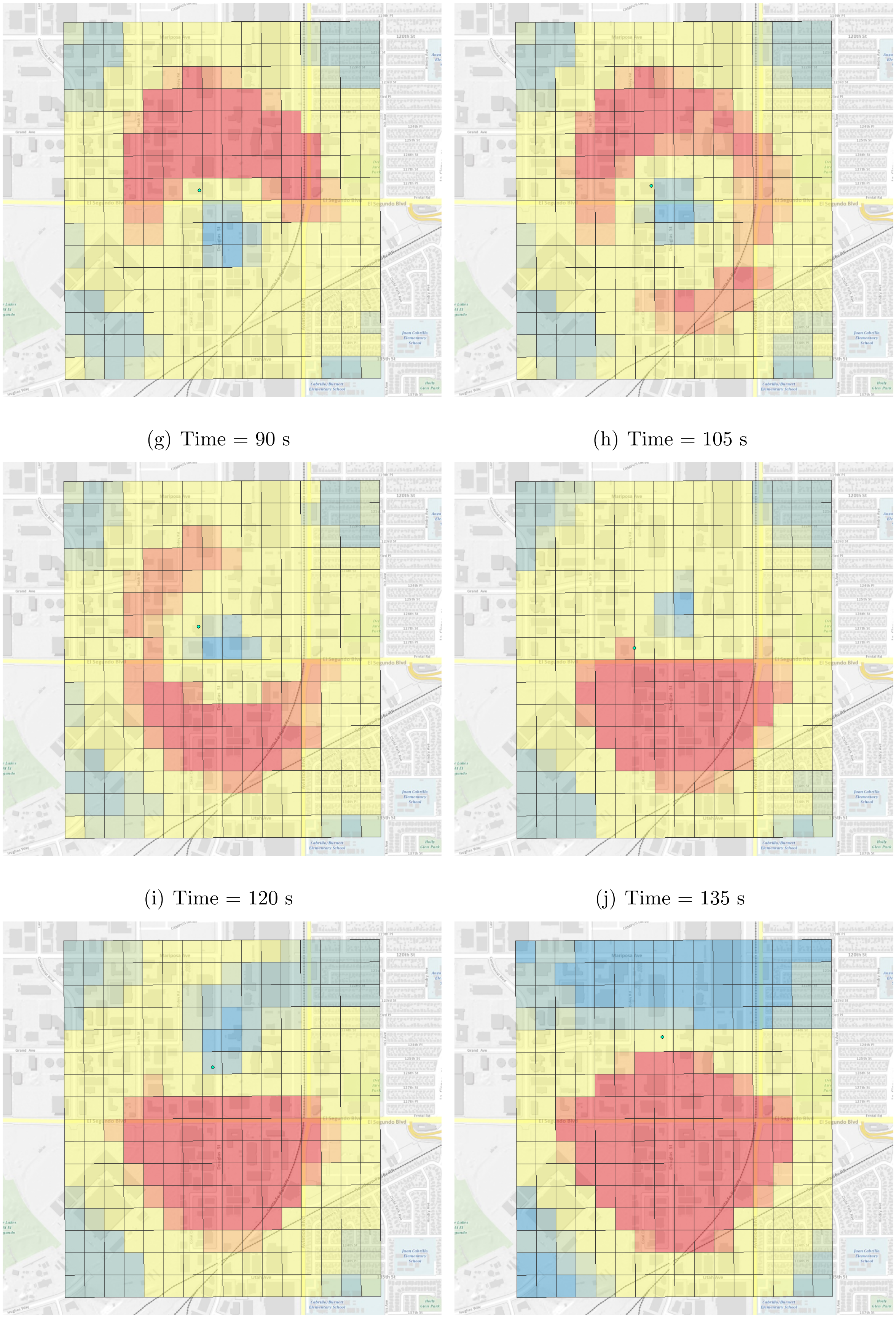
(b) Time = 15 s

(c) Time = 30 s

(d) Time = 45 s

(e) Time = 60 s

(f) Time = 75 s



(g) Time = 90 s

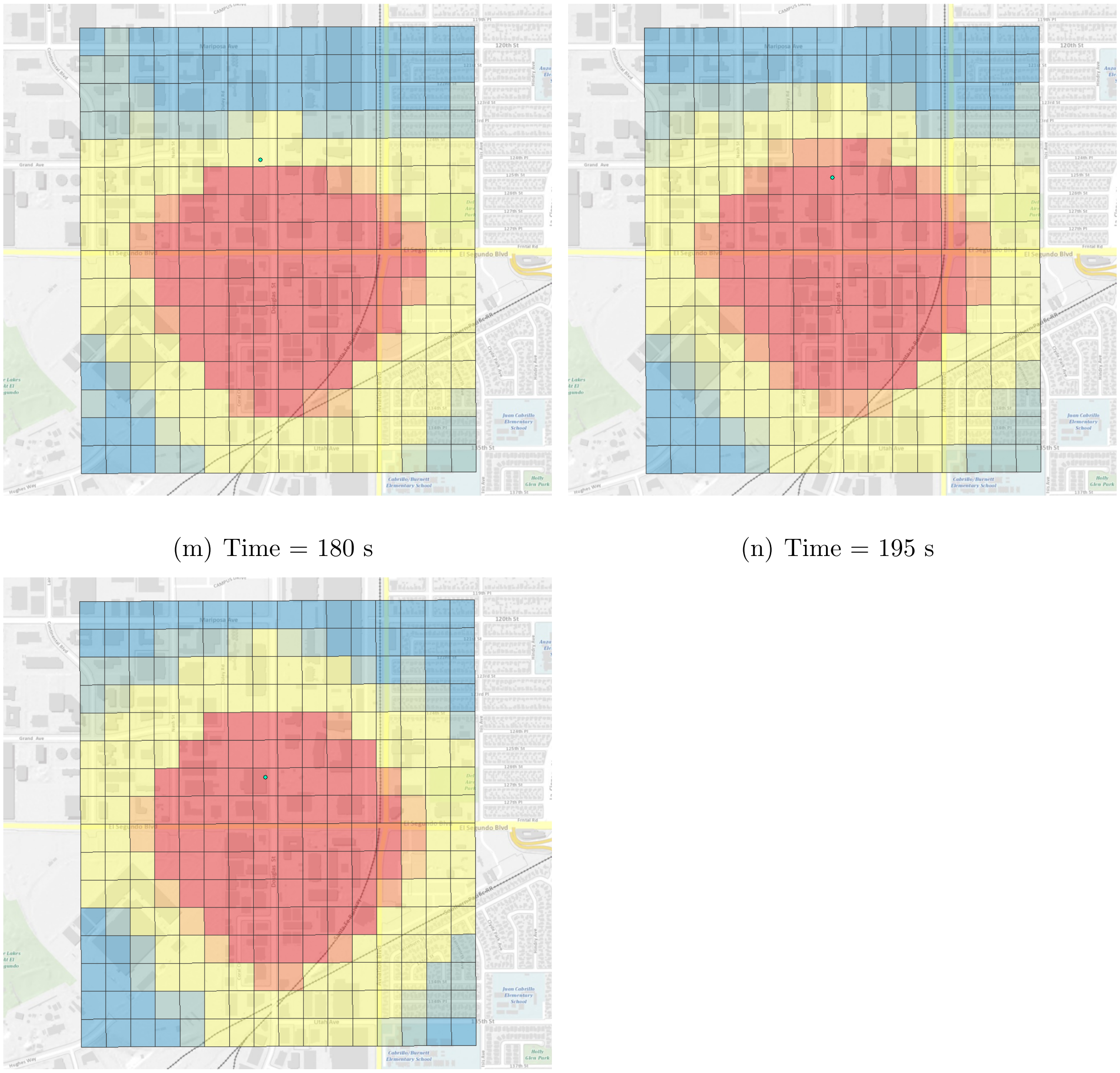
(h) Time = 105 s

(i) Time = 120 s

(j) Time = 135 s

(k) Time = 150 s

(l) Time = 165 s



(m) Time = 180 s

(n) Time = 195 s

(o) Time = 210 s

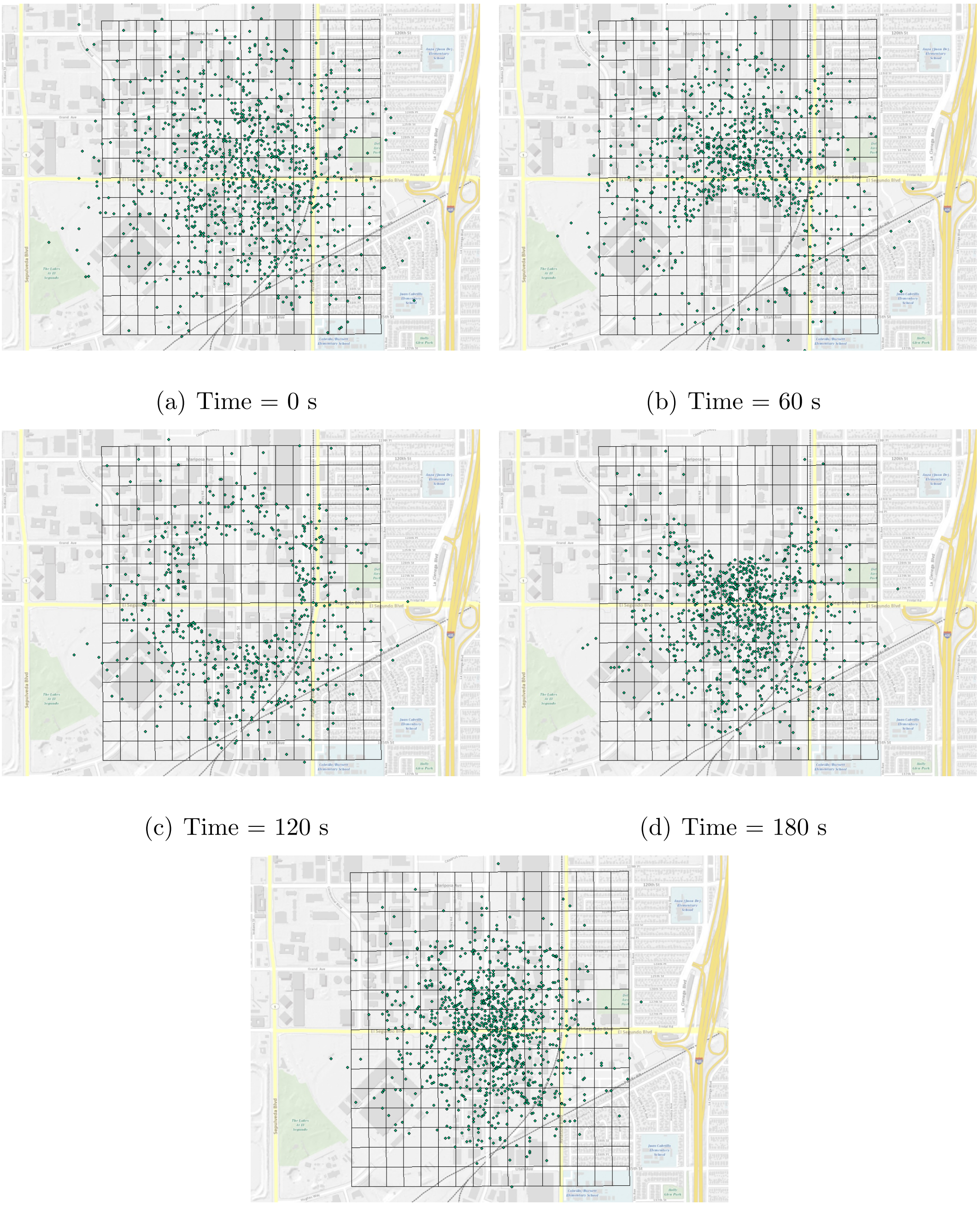
Figure 35. Optimized hot spot analysis for 1000 static normally distributed receivers

GPS Sensors: Normal Distribution, 1000 Users, Mobile, Time Between Sensing 60s.

Table 7. 1000 normally distributed mobile receivers performance (m)

Minimum	Maximum	Average
78.59	961.91	417.01

This section presents the data set for the 1000 mobile normally distributed user set. The filtered data is presented in accordance with the sampling rate in Figure 36, and finally the optimized hot spot analysis is presented in Figure 37. The estimated jammer location is marked by a green point feature. This data set is included simply to show that there were no major deviations from the static normal set explored above. However, unlike the uniform mobile variant as the receivers move during the scenario the distribution of the receivers is maintained. Based on the normally distributed random number draw it is more likely that the receiver's way points are set near the mean of the AoI, and the receiver speed is set to low for the initial cluster of receivers to reconfigure in any way prior to the jammer exiting the AoI. The variant provides the same false positive reports for the jammer location even after the jammer has left the AoI similar to the other normally distributed variants. Potential mitigation for this artifact are included in the future work recommendations of this document.



(a) Time = 0 s

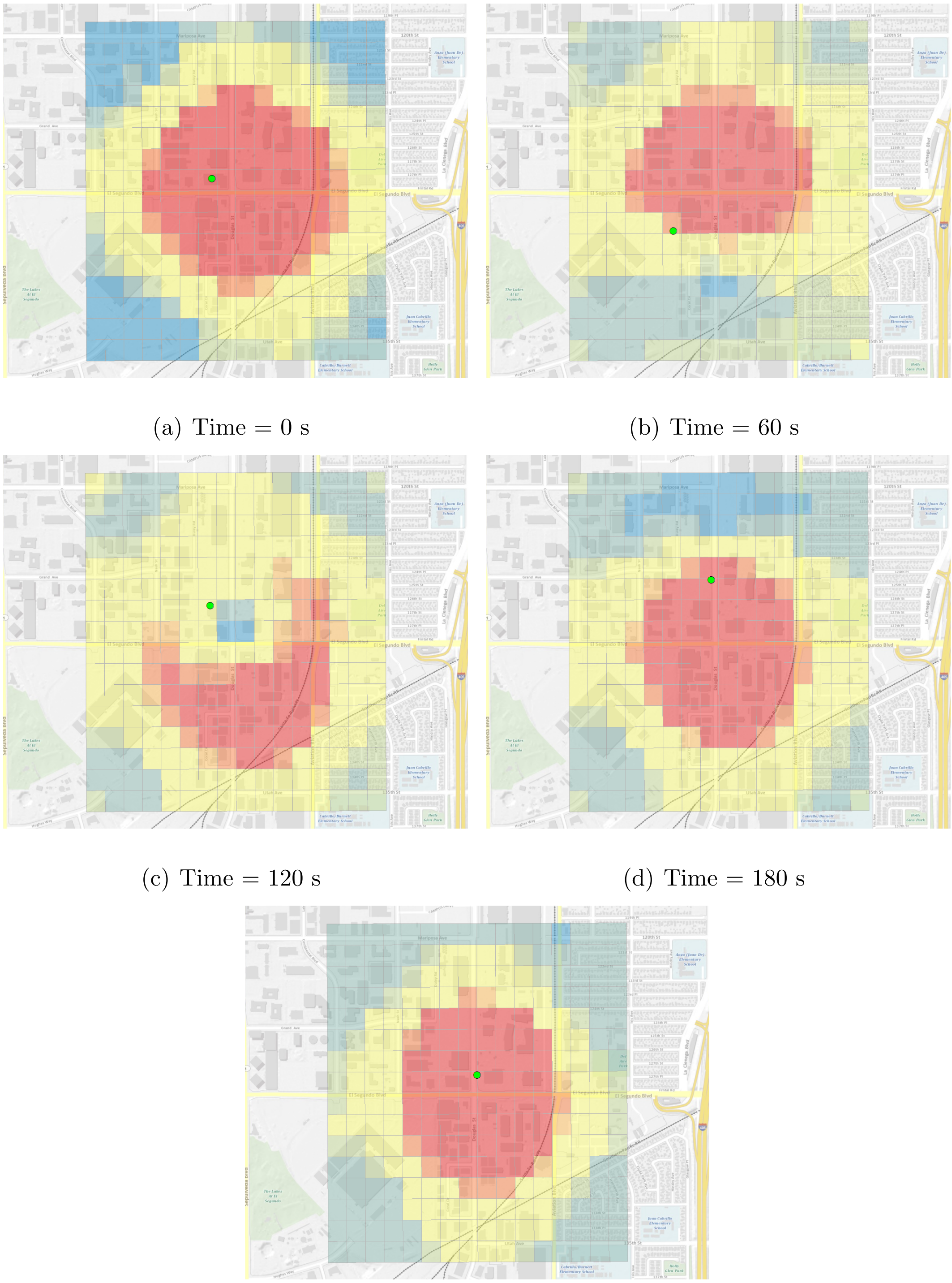
(b) Time = 60 s

(c) Time = 120 s

(d) Time = 180 s

(e) Time = 240 s

Figure 36. 1000 mobile normally distributed receivers: filtered out $(C/(No+Io) < 28dB$



(a) Time = 0 s

(b) Time = 60 s

(c) Time = 120 s

(d) Time = 180 s

(e) Time = 240 s

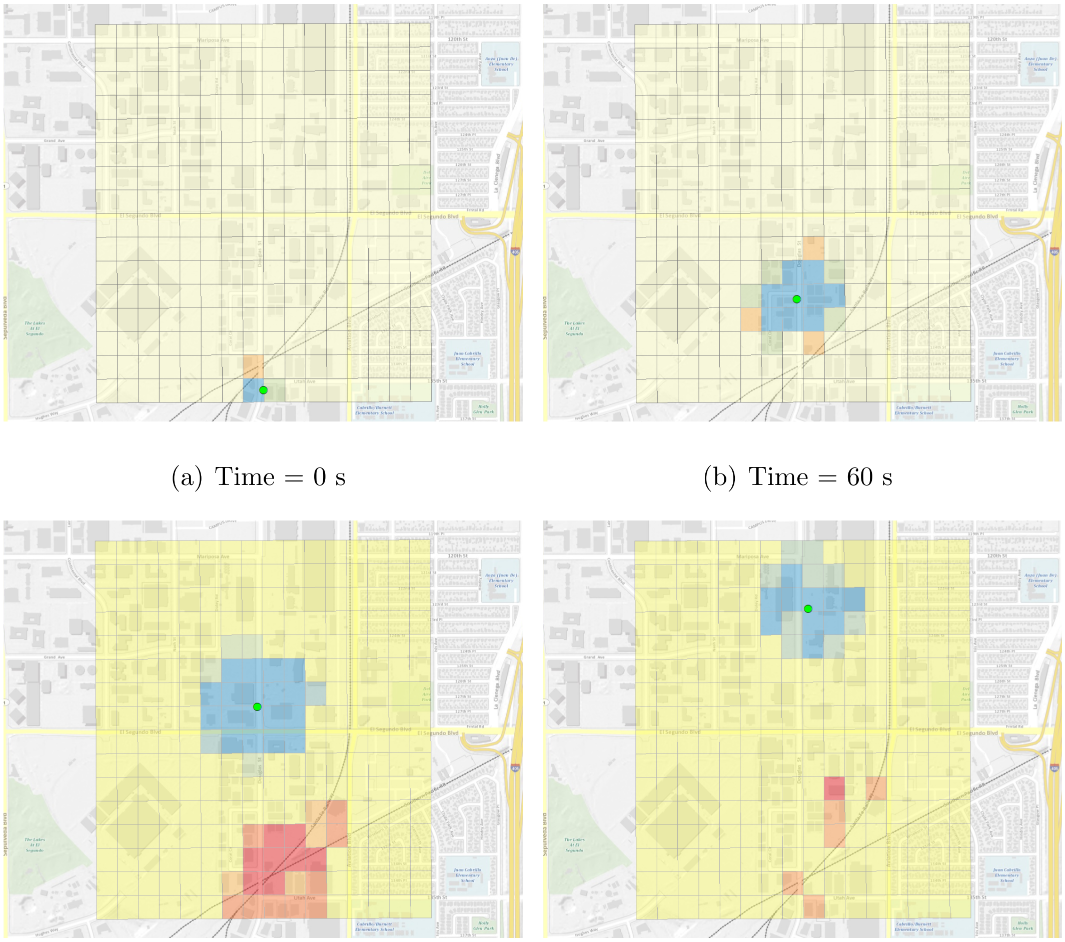
Figure 37. 1000 static normally distributed receivers: filtered out $(C/(No + Io) < 28dB$

GPS Sensors: Uniform Distribution, 500 Users, Static, Time Between Sensing 60s.

Table 8. 500 uniformly distributed static receivers performance (m)

Minimum	Maximum	Average
5.86	119.35	58.53

The next two variants considered a smaller number of available users during sensing operations. The first variant focuses on 500 uniformly distributed static receivers. The 500 receiver set performs similarly to the 1000 receiver set. The on-the-minute sampling time provides a clear picture to the effectiveness of the reduced number of users. Figure 38 includes the optimized hot spot analysis results. The filtered raw receiver location figures are omitted as they closely reflect the other static uniform samples above and exhibit the same behavior. The optimized hot spot analysis tool detected no significant hot or cold spots for the fourth and fifth minute of the analysis period. The absence of any significant clustering resulted in no false positive locations estimates for the jammer. A consumer of the sensing data could draw an intuitive conclusion that the jammer has either left the AoI or simply ceased operations. Additional runs would be required to develop averages and standard deviations for more statistically relevant analysis, but the initial analysis indicates that for the uniform distribution reducing the number of user down to 500 does not alter the uniform distribution variant's jammer location estimate. Any additional reduction of available receivers was not explored as a part of this research. Further exploration of number of receivers vs. location estimate is warranted. Additionally, correlation analysis of the effectiveness of the number of receivers vs. the power of the GPS jammer is also a topic for further research..



(a) Time = 0 s

(b) Time = 60 s

(c) Time = 120 s

(d) Time = 180 s

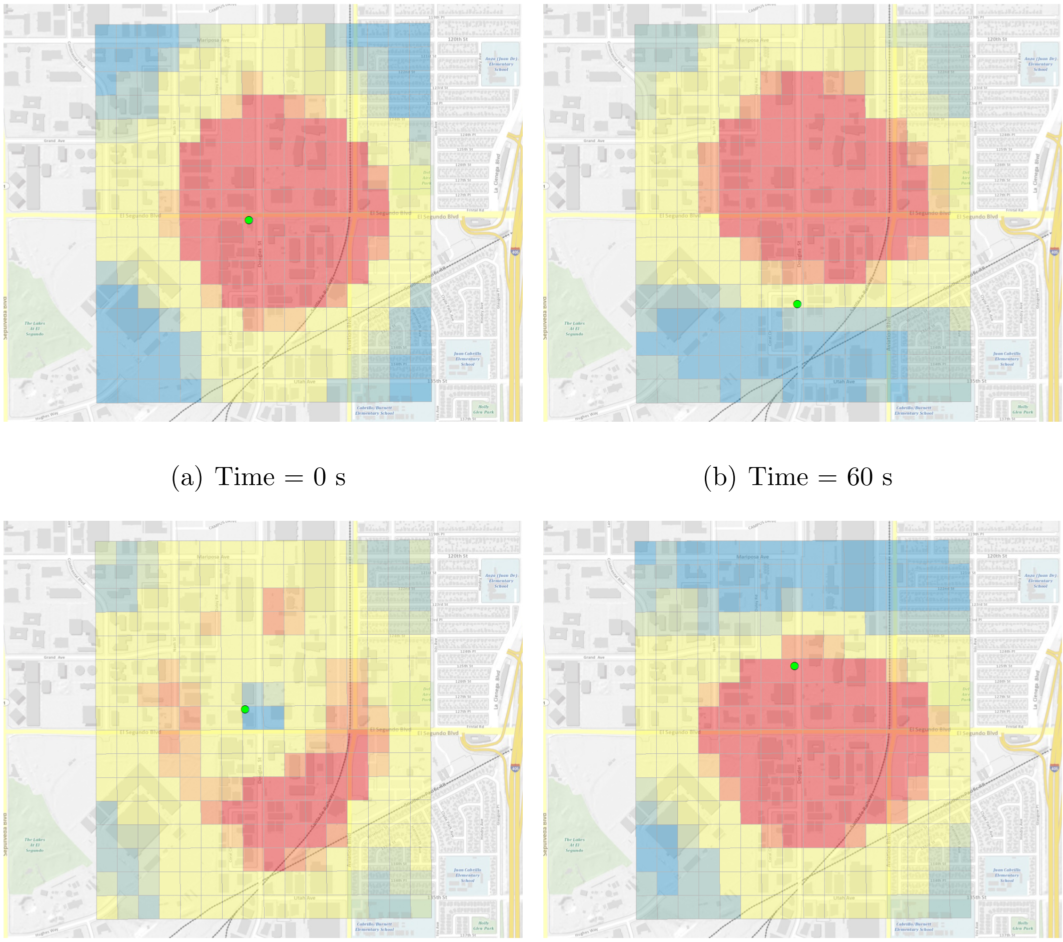
Figure 38. 500 static uniformly distributed receivers: filtered out $(C/(No + Io) < 28dB$

GPS Sensors: Normal Distribution, 500 Users, Static, Time Between Sensing 60s.

Table 9. 500 normally distributed static receivers performance (m)

Minimum	Maximum	Average
27.7	769.63	292.04

The final architecture variant included in this section is a reduced number of users that are normally distribution within the AoI. This data set is sampled every minute. Using the 1000 normally distributed receiver set above as a guideline it was anticipated that this configuration would experience similar issues with estimating the jammer's position. The Figure set 39 and error estimation data in Appendix C shows that the reduced number of user did not enhance or detract from the variants ability to identify the jammers position. The final sampled time is omitted as it was identical to the fourth sample. The variant did experience the same issue as the larger normally distributed user sets and the detected cold spots at the fringes of the AoI continue to influence the jammer location estimate.



(a) Time = 0 s

(b) Time = 60 s

(c) Time = 120 s

(d) Time = 180 s

Figure 39. 500 static normally distributed receivers: filtered out $(C/(No + Io) < 28dB$

4.3 Architecture Effectiveness Analysis

As a result of this initial analysis it is apparent that an inference can be drawn between the distribution of the receivers and the Data Processing Center's ability to calculate an estimated jammer location. The uniform distributions provide a much cleaner average estimate than the normally distributed variants. The data would also suggest that a faster sensing rate might not result in a better estimate of the jammer position, but does reveal changes within the user distribution that would otherwise go unnoticed. Without any adaptive processing techniques to account for this change the faster sampling rate does have a higher chance of produce more false positive estimates or high error estimates. The higher sampling rate does tend to provide a clearer estimate when the receivers are static and is clear that the high volume uniform distributed user set provides the best track of the jammer position. It is also clear that more understanding of the movement patterns of the users is required to understand emerging trends in the receiver distribution. Currently the latitude and longitude of mobile users during transit are calculated in STK and are generated by interpolating a line between two way points. The analysis suggest since these interpolated points are not drawn from the same distribution as the generated way points the distribution of the receivers is not fixed during movement. There are provisions within STK to assign way points based on a desired arrival time instead of utilizing velocity, but this method would not be representative of movement capabilities of users. Under the current way point determination scheme utilizing arrival time as the determining transit factor would cause users to virtually teleport between locations if the way points were set far apart. This issue could be mitigated by creating subareas within a larger AoI and evenly distributing the number of available receivers into each subarea. Then create way points uniformly within each subarea for the receiver sets. This would closer mimic user on traveling on foot over shorter distances, and could potentially preserve the intended distribution.

Another issue briefly discussed above is that issue caused due to the jammer's presence within the AoI only comprises a portion of the total analysis period. While this analysis did not focus on the effect of the permanency of the jammer on the MCS architecture it does impact the number of false positives for certain architecture variants. As the jammer exits the AoI any of the normally distributed data sets, and the final sample times of the uniform mobile variants create a false indicator for the presence of a jammer. This occurs due to the statistically significant lower number of users at the fringes of the analysis area and a large clustering of users in the center of the AoI. The optimized hot spot analysis finds the average location of the cold spots even before the jammer begins to effect a significant number of receivers and as the jammer egresses from the AoI an estimated point is placed in the center of the AoI where it is known that the jammer cannot be due to the high number of GPS responses. Potential ways to mitigate this effect are included in the advanced algorithm development recommendation in the final chapter of this document.

The analysis performed in this chapter revealed that it is possible for a MCS architecture to provide a report of the GPS QoS within and AoI. The analysis also indicates that utilizing a SOA with defined interfaces, cooperative participants, and enough sensor to provide input the MCS architecture can provide an estimate for the location of a GPS interference source. However, can be noted that there were not enough repetitions of each variant to conclude with any statistical relevance that varying the architecture configurations had an effect on the variants ability to detect and identify the GPS jammer. These conclusions provide initial insight into addressing the fourth research question identified in Chapter I. However, additional replication of each scenario would allow for the raw distance error MOE to be converted to an average distance error bounded by a confidence interval which would provide more conclusive evidence as to the effectiveness of the MCS architecture. This is one of many possible areas of potential follow research to this initial analysis. A summary table of the results presented above is included in table 10. Additional future research is highlighted in the final chapter of this document. These recommendations include enhancements and further refinement to the proposed SOA, adding more validity to the M&S efforts, and development of more robust geo-spatial processing algorithms to enhance the MCS architectures jammer identification capability.

Table 10. MCS Variant Effectiveness Summary Results

Variant	Minimum	Maximum	Average	Discussion
1000 Static Uniform	13.7	121.9	46.2	Best baseline jammer location estimate
1000 Mobile Uniform	5.85	320.9	83.3	Exhibited redistribution of users during simulation
1000 Static Normal	37.9	776.6	286.1	Large error due to non-jammer related cold spots
1000 Mobile Normal	78.59	961.9	417	Worst performance due to tighter clustering of receivers as scenario progressed
500 Static Uniform	5.9	119.3	58.53	Minimal impact due to reduction in users
500 Static Normal	27.7	769.6	292	Minimal impact due to reduction in users

V. Conclusions and Future Work

This chapter concludes the study of the potential use of a MCS architecture to monitor the GPS QoS within an AoI. This chapter provides a summation of the activities performed during this research, an overview of the final conclusions drawn from the data produced by the methodology, and a discussion of how this research method compares to other similar topics. The final portion of this chapter provides a set of future research topics that could be used to gain more insight into the use of a MCS architecture for GPS QoS monitoring.

5.1 Conclusion

This research focused on addressing a capability gap identified by the GPS program office for monitoring the QoS of the GPS SIS in near real time. As indicated in Chapter I there is currently no monitoring capability available to protect the United States CI that relies on the GPS signal for everyday operations. This research utilized system engineering techniques of MBSE and model based architecture evaluation to assess the potential use of modern day mobile electronic devices and devices considered a part of the IOTs to address this capability gap.

The literature review conducted as a part of this research provides a good frame of reference for the current state of research in areas of GPS jamming, SOA, and GIS technology. The intersection of these topics was the focus of this research and the best practices extracted during the literature review were influential during the development of the MCS architecture. The Hu *et al*, Kotsen *et al*, and Sheng et al articles addressed the second research question proposed in Chapter I regarding the key factors to consider when developing a MCS architecture [6][4][5]. This section also provided examples of the MITRE TADA program and capabilities of Chronos

Technology to address the first research question regarding what methods exisit for detection of GPS SPS jamming [24][25].

The integration of the SoaML and DoDAF framework was a unique approach to solving system architecture problems that require a SOA within the context of a DoD architecture solution. SoaML was used to supplement the more ambiguous guidance within the DoDAF framework for capturing the service interactions of the MCS architecture. Defining the service interactions was a key factor into exploring the possibility of using web based open source standards for exchange of service requests and responses. For the purposes of this research development, a complete set of architecture products was not required. The key products developed described the high level configuration of the MCS architecture and detailed the interactions of architecture participants in the M&S activities. This architecture approach was utilized do address a portion of the third research question identified in Chapter I. This processes addresses the development activities that are required prior to the lower level M&S activities and identifies applicable techniques for architecting a MCS architecture.

Another topic explored by this research were the concepts of MBSE and model based architecture analysis through the use of the two architecture frameworks and a suite of available industry tool kits. Utilizing MBSE to evaluate the potential of the MCS architecture allowed for quick evaluation of multiple architecture variants, and provided a framework and baseline for any future work that would seek to add more fidelity to the architecture and architecture model. The application of MBSE and the techniques identified in the Abusharekh *et al* and Hu *et al* text identified DES as an appropriate modeling technique for SOAs which addressed a large portion of this research's third research question.

This research relied heavily upon the use of a set of M&S tools to demonstrate the feasibility of monitoring the GPS QoS through the use of a SOA. These tools provided mechanisms to simulate idealized RF interactions between architecture participants, quickly manipulate parameters of the MCS architecture, and execute spatial analysis algorithms on the simulated data to assess the performance of the architecture variants. The development of the MCS model combined the principles of remote sensing, available GPS monitoring technologies, and the available set of GIS analysis capabilities to provide initial insight into the use of MCS for GPS monitoring. The identification of an applicable set of M&S tools and their initial use to evaluate the MCS architecture addresses the final portion of the third research question in Chapter I. Developing an initial intersection of these tools and utilizing them to evaluate a SOA is the largest contribution this research has to the systems engineering body of knowledge in the area of Geo-spatial and SOA architecture MBSE.

The final section of this document includes the recommendations for future work in the areas of GPS monitoring, enhancement of the MCS architecture model, and refinement of the MCS network and web interactions.

5.2 Future Work

Geo-location Algorithm Enhancement.

The geo-location capability of this research is an initial application of a GIS solution to identify a source of GPS interference. The analysis approach of using a spatial mean location has been shown to work only in specific instances, and still has instances of high degrees of error. One of the first areas of additional research to enhance the MCS architecture's performance should focus on incorporation of more robust GIS processing algorithms and the potential inclusion of machine learning techniques to identify patterns of GPS interference. Additional topics of research in this area could include algorithm's processing efficiency i.e. how long does it take to analyze a given data set. Additionally this research topic should investigate the algorithm or set of algorithms ability operate on different spatial scales, both large and small.

Realistic Receiver Modeling.

As stated in Chapter III the parameters that defined the GPS receivers were extracted from Kaplan and Hegarty's *Understanding GPS Principles and Applications, 2nd edition*. The parameters that were set within the receiver model were the bare minimum to simulate the desired behavior. Each receiver was considered loss-less and noiseless which in practice is impossible. The antenna behavior for each receiver was assumed to be ideal and used an idealized system noise temperature, assumed a perfect impedance match with the receiver input, and antenna orientation was not included as a relevant factor. Additional connection losses such as low noise amplifier to receiver line loss were not included. Also, any loss that might occur due to bad signal noise filtering, or any possible self interference that other microelectronics in the mobile device might cause during operations. The scope of this research was not intended to design a new mobile GPS receiver, and the receiver model used in this research has large room for improvement. Testing could be performed on candidate mobile electronic device to extract some of the steady state normal operational parameters to use as inputs to the M&S activities. This information might already be available through the GPS program office's GPS User Equipment division. Undoubtedly, adding more realistic signal losses, gains, and GPS time tracking performance data will help better define how effective the receivers within the MCS architecture can locate GPS interference.

Web Interface Development.

For this research, the interfaces between the architecture participants was handled largely through representative Python or STK model entities. For example, there was not a formatted sensing request broadcast to the GPS receivers, instead the required message data was extracted from an STK text report. In a traditional SOA a clearly defined message schema between participants is paramount for insuring loose coupling between participants and scalability to new users. Development of the actual web interfaces for communicating data through the architecture, as well as defining the appropriate XML, Resource Description Framework (RDF), or WSDL would allow for a more realistic latency analysis for communicating data between the MCS participants. The World Wide Web Consortium provides a host of resources for defining open source data interfaces for web applications, architecture, and devices [51]. Additional network analysis tools would be required to facilitate this analysis, and the outputs from those tools could then be integrated into the simulation framework developed as a part of this research. Inclusion of the would result in a higher fidelity MBSE effort. However STK is not a native network analysis tool, and may not be able to incorporate the higher fidelity communication protocols on top of the RF communication packages. A more appropriate analysis tool could be the SteelCentral network monitoring program developed by Riverbed Technology, formally known as Opnet. A high level of proficiency and understanding of how web design, network interfaces, and programming would be required for this effort.

Higher Fidelity Environmental Modeling.

The initial intent of this research was to incorporate a three dimensional model of the AoI and enable the STK urban propagation tools during the simulation. However, this was removed from the scope of this initial research due to time constraints.

Including the urban propagation analysis would allow for buildings and other terrain obstructions to obscure receivers from the GPS jammer, and would have reflected more realistic capabilities of the MCS to detect the interfering source. It would have provided addition realistic degradation of GPS access for receivers due to building obstruction; similar to attempting to acquire a GPS signal while parked in a parking garage. The implementation of terrain and building data is supported within STK via the inclusion of the STK ptddx file format and GIS shapefiles respectively. The Los Angeles County GIS community has published a complete data set for all buildings in the county via the Los Angeles Region Imagery Acquisition Consortium (LARIAC) program. The most important aspects of the data set is that it provides all the polygon information, which captures building placement and dimensions including the height information. The LARIAC program has also produced terrain data for the Los Angeles county area. The data is formatted in the California state plane reference coordinate system, but can be projected into the required WGS 1984 format for STK to process. This higher fidelity modeling would require an STK urban propagation extension license which is currently available as a part of AFIT's educational alliance with AGI.

Live Data Integration.

Live data input into the architecture analysis would greatly enhance the validity of utilizing mobile receivers of opportunity to sense the GPS QoS. Both STK and ArcGIS can support inclusion of live data in the loop or live data playback for analysis. There is an increasing number of studies and efforts being undertaken by industry to gather GPS data form mobile receivers. Most cities include GPS capability and tracking information for taxi companies or other public transportation means [52]. As indicated in chapter II many GIS studies and companies have begun to prioritize floating car data. The use of this data could provide a high degree of insight into the movement patterns of actual receivers within an AoI. This data could be a starting point for utilizing real data in conjunction with the STK jamming scenario. At best, continued pursuit of mobile receiver data from cell phone companies such as AT&T, Verizon, Sprint, etc. would be of great benefit. It provides the scenario with more realistic receiver numbers, movement patterns, dwelling periods in locations, and distributions within an AoI. A long term goal if this research is to be continued should be to establish a strategic partnership between AFIT, AFSPC/GP, and a commercial entity interested in understanding the reliability of GPS for a AoI; such as the FAA or the telecommunication companies mentioned above. If a partnership cannot be established with a company or institution with access to this type of data then it may be necessary to purchase the needed data.

Data Latency and Storage Analysis.

For the purposes of this research, the required data storage, transport, and analysis was performed in a closed loop fashion. The data was extracted from STK via Python, and Python immediately parsed the text reports to generate the required message sets. However, in reality there would be a substantial delay for the transport of data through a telecommunications network and into an organized storage configuration. The concepts of requesting, storing, and managing "Big Data" have become driving forces in the area of data analytics. The use of large database storage and access schema such as Hadoop or SQL might be well suited to organizing and querying a larger MCS data set. Additionally, if possible the MITRE TADA application could be used to understand the latency of sending a sensing request through a cellular network and receiving a response from a mobile device. These experiments could be done across heterogeneous devices and locations to establish some basic statistical parameters such as average response time, average processing time, etc. that could then be used in a larger scale simulation. It is recommended that any research in this area be conducted by someone with an exposure to computer programming and/or network engineering. The combination of a data storage server and the ESRI hosted GIS web application capability could be used to explore the scalability of the GPS MCS architecture. The data access and processing time can be measured in these circumstances to understand the impacts on the architecture due to the more realistic access method. STK does support data routing latency analysis, but on large scales these computations are more appropriately executed utilizing a dedicated network analysis tool.

Multiple Jammer Scenario.

For this research the assumption of one jammer made the analysis simpler by being assured that any cold spots were either caused directly by the jammer or due to the distribution of the receivers. The jammers position was able to be calculated based on the average location of the statistically less populated sub-areas in the AoI. This method would not be effective for a scenario with two jammers. The mean location of the total ”cold” areas would just produce a point midway between the two jammer locations. More robust feature matching and analysis would be required in the ESRI ArcGIS software to try and identify the location of two or more jammers in a large AoI. For this follow on research area it is recommend that the initial analysis assume homogeneous jammer types, and should focus on the MOE of identifying the multiple jammer locations. A potential starting point would be to identify two stationary jammers in an AoI, and then apply movement to the jammers.

Heterogeneous GPS Jammer Analysis.

This research did not explore the effects on the MCS architecture vs different types of jammers. Additional research should be conducted to understand the ability of an MCS architecture to respond to jammers that utilize varying power levels, are not constantly active, and/or utilize different transmitter antenna patterns. Research in this area should also relax the jammer type constraint and explore the ability of an MCS architecture to detect GPS spoofing jammers. Being able to detect potential degradation effects of GPS timing is often more important than monitoring the position data. Timing drives bank transactions, stock market trades, and emergency vehicle estimated travel times. It is not apparent that STK or ESRI support the capability to model GPS spoofing, so a separate tool or custom objects and scripts would need to be developed to define the properties of the spoofing jammer in each program. There are other GPS M&S tools such as the GPS Interference and Navigation Tool (GIANT) by the company LinQuest [53] that specialize in GPS jamming and could provide a more appropriate solution for modeling GPS spoofing jammers. Additionally incorporation of an operationally relevant spoofing capability into the analysis might force the research to a higher classification level. This is not prohibitive but full understanding of the security requirements should be complete before beginning this research, especially confirmation that any M&S tools are cleared for the classified environment.

Advanced Detection Capability Augmentation.

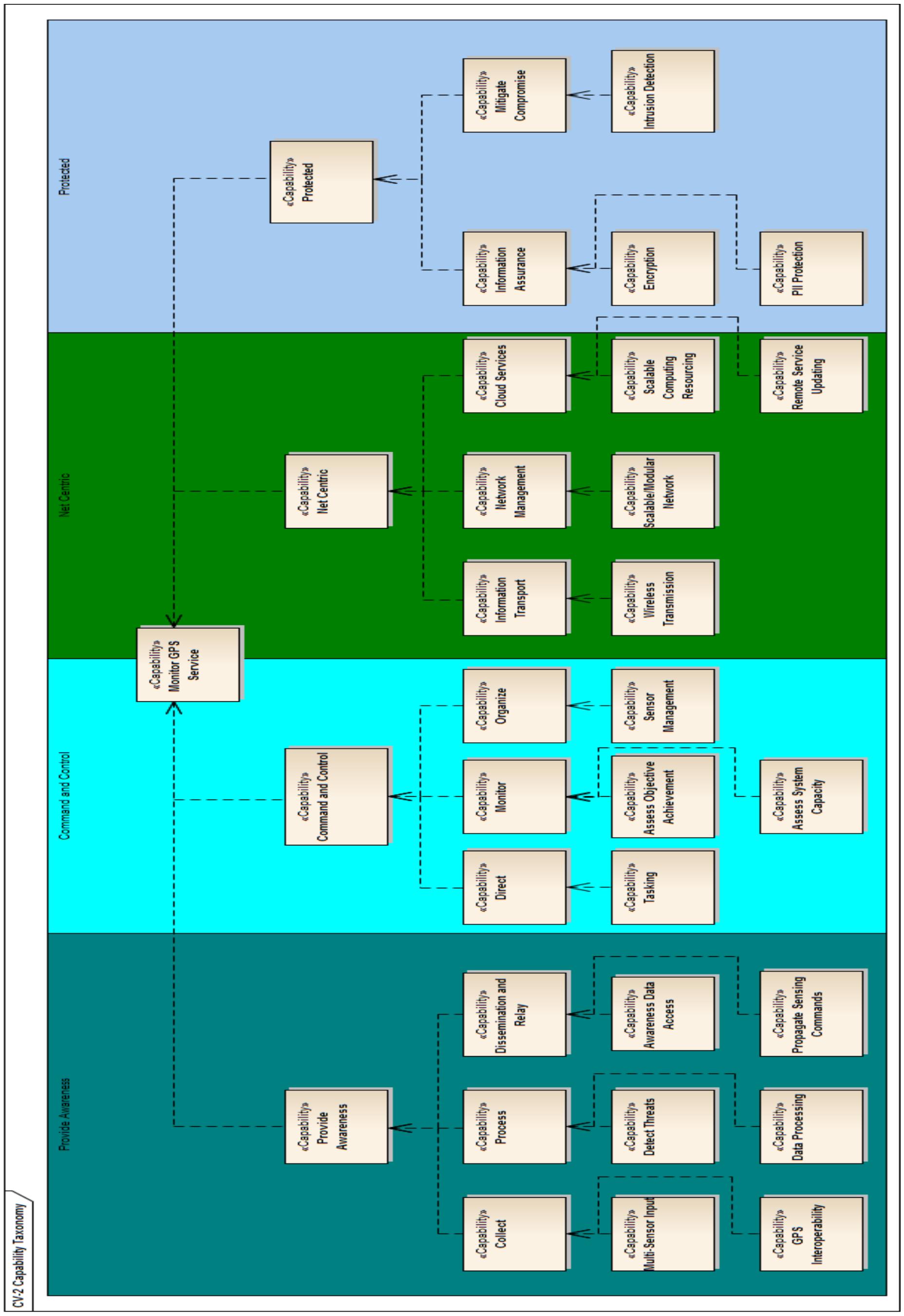
As identified in the scope section of this research the MCS architecture only considers standard GPS sensors as inputs into the model. However, given the overall goal of the monitoring network is to protect the CI of the United States research into a hybrid MCS architecture with dedicated detection sensor could provide a more robust capability for detection and identification of GPS threats near these critical areas. Research would need to be conducted to determine the capabilities of jammer detection technology, scalability analysis performed to help develop a viable business case, and a method for including the advanced sensor data into the jammer location assessment techniques explored in this thesis. The Chronos Technology Limited report is a realistic place to begin analysis for the capabilities of jammer detection technology. Incorporation of an advanced detection capability into the MCS architecture simulation would not be trivial and consideration for dedicated support from the M&S tool developer should be included.

Inclusion of Military Receiver Capabilities.

Lastly, the inclusion of GPS augmentation capability would provide valuable insight to understand the full potential of the MCS architecture. Another augmentation to the MCS architecture could be to focus on modeling an AoI with a heterogeneous mix of military and civilian receiver types. Additional capability for the military receivers could be modeled to account for the benefits of P(Y) and M-code capabilities. Incorporating technical performance information of these advanced military receivers could result in a higher security classification level and full understanding of the security requirements should be complete before beginning this research. This research could also include a shift of the AoI from a Continental United States (CONUS) application to an area that is more combat operationally representative.

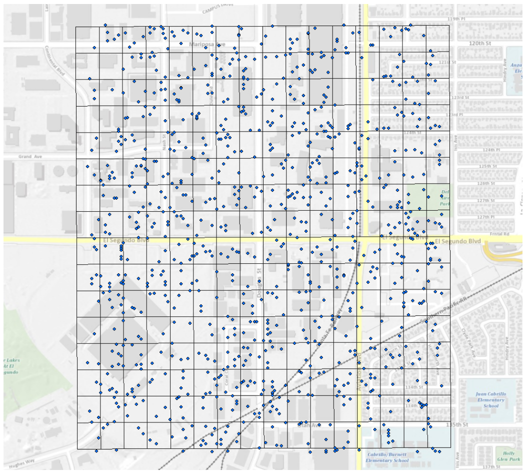
Appendix A. MCS Capability Taxonomy

Figure 40. Tier 3 Capability Taxonomy

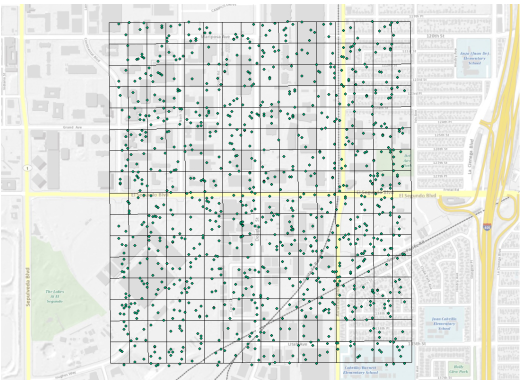

CV-2 Capability Taxonomy
Protected
Net Centric
Command and Control
Provide Awareness
«Capability» Monitor GPS Service
«Capability» Protected
«Capability» Mitigate Compromise
«Capability» Intrusion Detection
«Capability» Information Assurance
«Capability» Encryption
«Capability» PII Protection
«Capability» Net Centric
«Capability» Cloud Services
«Capability» Scalable Computing Resourcing
«Capability» Remote Service Updating
«Capability» Network Management
«Capability» Scalable/Modular Network
«Capability» Information Transport
«Capability» Wireless Transmission
«Capability» Command and Control
«Capability» Organize
«Capability» Sensor Management
«Capability» Monitor
«Capability» Assess Objective Achievement
«Capability» Assess System Capacity
«Capability» Direct
«Capability» Tasking
«Capability» Provide Awareness
«Capability» Dissemination and Relay
«Capability» Awareness Data Access
«Capability» Propagate Sensing Commands
«Capability» Process
«Capability» Detect Threats
«Capability» Data Processing
«Capability» Collect
«Capability» Multi-Sensor Input
«Capability» GPS Interoperability

Appendix B. Architecture Variant Initial Raw Receiver Distribution

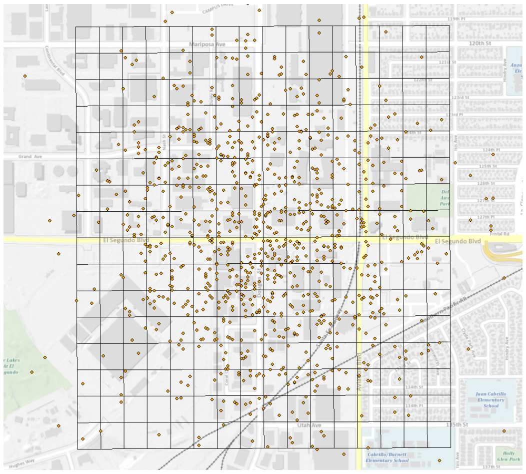
This appendix contains the unfiltered receiver data sets for each of the architecture variants explored in Chapter IV. If the variant is a mobile variant the data set is initial distribution of the receivers.



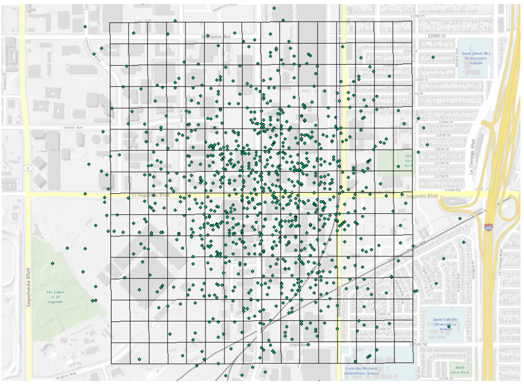
(a) 1000 Static Uniform



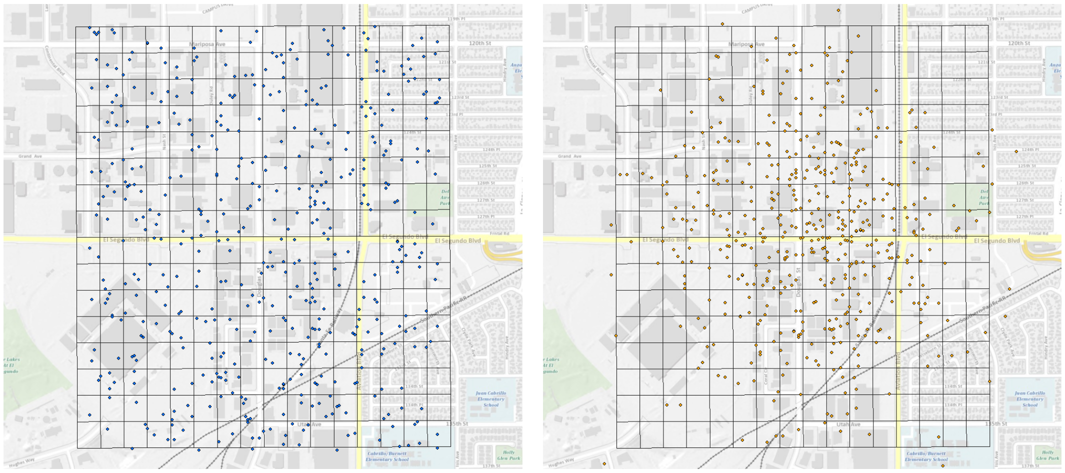
(b) 1000 Mobile Uniform



(c) 1000 Static Normal



(d) 1000 Mobile Normal



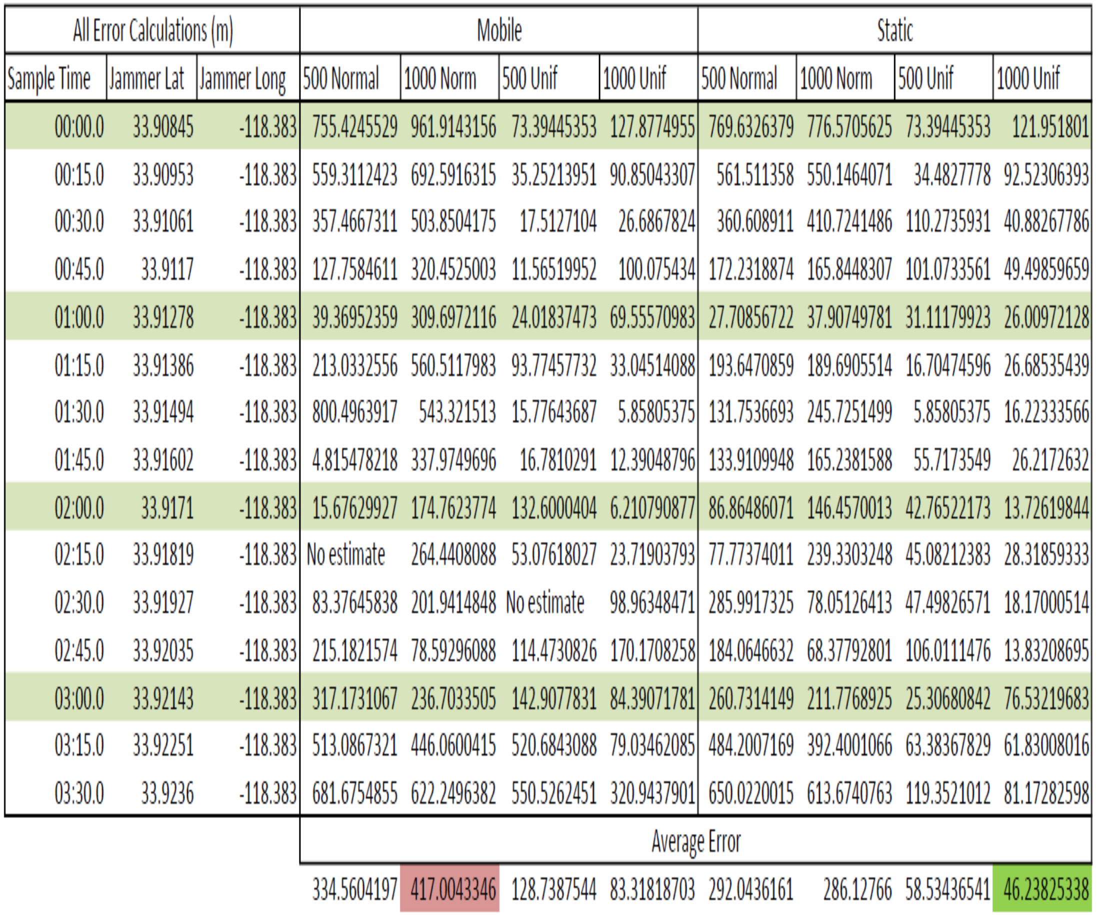
(a) 500 Static Uniform

(b) 500 Static Normal

Figure 41. Unfiltered Raw Receiver Distributions

Appendix C. Architecture Analysis Appendix

Table 11. Architecture Variant Jammer Location Estimates



All Error Calculations (m)			Mobile				Static			
Sample Time	Jammer Lat	Jammer Long	500 Normal	1000 Norm	500 Unif	1000 Unif	500 Normal	1000 Norm	500 Unif	1000 Unif
00:00.0	33.90845	-118.383	755.4245529	961.9143156	73.39445353	127.8774955	769.6326379	776.5705625	73.39445353	121.951801
00:15.0	33.90953	-118.383	559.3112423	692.5916315	35.25213951	90.85043307	561.511358	550.1464071	34.4827778	92.52306393
00:30.0	33.91061	-118.383	357.4667311	503.8504175	17.5127104	26.6867824	360.608911	410.7241486	110.2735931	40.88267786
00:45.0	33.9117	-118.383	127.7584611	320.4525003	11.56519952	100.075434	172.2318874	165.8448307	101.0733561	49.49859659
01:00.0	33.91278	-118.383	39.36952359	309.6972116	24.01837473	69.55570983	27.70856722	37.90749781	31.11179923	26.00972128
01:15.0	33.91386	-118.383	213.0332556	560.5117983	93.77457732	33.04514088	193.6470859	189.6905514	16.70474596	26.68535439
01:30.0	33.91494	-118.383	800.4963917	543.321513	15.77643687	5.85805375	131.7536693	245.7251499	5.85805375	16.22333566
01:45.0	33.91602	-118.383	4.815478218	337.9749696	16.7810291	12.39048796	133.9109948	165.2381588	55.7173549	26.2172632
02:00.0	33.9171	-118.383	15.67629927	174.7623774	132.6000404	6.210790877	86.86486071	146.4570013	42.76522173	13.72619844
02:15.0	33.91819	-118.383	No estimate	264.4408088	53.07618027	23.71903793	77.77374011	239.3303248	45.08212383	28.31859333
02:30.0	33.91927	-118.383	83.37645838	201.9414848	No estimate	98.96348471	285.9917325	78.05126413	47.49826571	18.17000514
02:45.0	33.92035	-118.383	215.1821574	78.59296088	114.4730826	170.1708258	184.0646632	68.37792801	106.0111476	13.83208695
03:00.0	33.92143	-118.383	317.1731067	236.7033505	142.9077831	84.39071781	260.7314149	211.7768925	25.30680842	76.53219683
03:15.0	33.92251	-118.383	513.0867321	446.0600415	520.6843088	79.03462085	484.2007169	392.4001066	63.38367829	61.83008016
03:30.0	33.9236	-118.383	681.6754855	622.2496382	550.5262451	320.9437901	650.0220015	613.6740763	119.3521012	81.17282598
			Average Error							
			334.5604197	417.0043346	128.7387544	83.31818703	292.0436161	286.12766	58.53436541	46.23825338

Bibliography

1. Department of Defense, Global Positioning System Program Office. *Global Positioning System Standard Positioning Service Performance Standard*, 4th edition, September 2008.

2. Prof. Charles Curry. Sentinel project - reprot on gnss vulnerabilities. Technical Report 001, Chronos Technology, April 2014.

3. Thomas Erickson. Some thoughts on a framework for crowdsourcing. In *Workshop on Crowdsourcing and Human Computation*, pages 1–4, 2011.

4. Alexander Kotsev, Francesco Pantisano, Sven Schade, and Simon Jirka. Architecture of a service-enabled sensing platform for the environment. *Sensors*, 15(2):4470–4495, 2015.

5. Xiang Sheng, Jian Tang, Xuejie Xiao, and Guoliang Xue. Sensing as a service: Challenges, solutions and future directions. *Sensors Journal, IEEE*, 13(10):3733–3741, 2013.

6. Jianpeng Hu, Linpeng Huang, Xuling Chang, and Bei Cao. A model driven service engineering approach to system of systems. In *Systems Conference (SysCon), 2014 8th Annual IEEE*, pages 136–145. IEEE, 2014.

7. Ashraf M. Abusharekh, Lawrence E. Gloss, and Alexander H. Levis. Evaluation of service oriented architecture-based federated architectures. *System Engineering*, 14:56–72, 2010.

8. Satheesh Gopi, R. Sathikumar, and N. Madhu. *Advanced Surveying: Total Station, GIS and Remote sensing.* Pearson India, 2006.

9. Pascal Neis and Alexander Zipf. Analyzing the contributor activity of a volunteered geographic information project - the case of openstreetmap. *ISPRS International Journal of Geo-Information*, (ISSN 2220-9964), 2012.

10. Xintao Liu and Yifang Ban. Uncovering spatio-temporal cluster patters using massive floating car data. *ISPRS International Journal of Geo-Information*, (ISSN 2220-9964), 2013.

11. Analytical Graphics Inc. Stk programming guide. `http://help.agi.com/stkdevkit/index.html`, 2015.

12. *National Space-Based Positioning, Navigation, & Timing Advisory Board Meeting minutes, December 10-11, 2014*, number 14th Meeting, The Omni Shoreham Hotel, Hampton Ballroom, 2500 Calvert Street NW, Washington, DC 20008, December 2014.

13. Department of Homeland Security. What is critical infrastructure. `www.dhs.gov/what-critical-infrastructure`, October 2013.

14. Mark W Maier. *The art of systems architecting.* CRC press, 2009.

15. GPS.gov. Official u.s. government information about the global positioning system (gps) and related topics. `www.gps.gov/systems/gps/modernization`, July 2014.

16. E. Kaplan and C. Hegarty. *Understanding GPS: Principles and Applications, Second Edition.* Artech House, 2006.

17. Jamming the global positioning system - a national security threat: Recent events and potential cures. White paper, National PNT Advisory Board, Novemeber 2010.

18. Michael Behrendt, Bernard Glasner, Petra Kop, Robert Dieckmann, Gerd Breiter, Stefan Pappe, Heather Kreger, and Ali Arsanjani. Introduction and architecture overview ibm cloud computing reference architecture 2.0. *Draft Version V*, 2011.

19. Federal Communications Commission. Communications act of 1934: as amended by the telecom act of 1996. `https://transition.fcc.gov/Reports/1934new.pdf`, 1996.

20. WAAS T&E Team. Global positioning system (gps) standard positioning service (sps) performance analysis report. Technical Report Report 89, William J. Hughes Technical Center, Atlantic City International Airport, NJ 08405, 2015.

21. Jeff Coffed. The threat of gps jamming: The risk to an information utility. *Report of EXELIS, Jan. Chicago*, 2014.

22. WAAS T&E Team. Waas performance analysis report. Technical Report Report 46, William J. Hughes Technical Center, Atlantic City International Airport, NJ 08405, 2013.

23. Sam Pullen, Per Enge, Stuart Shaw, Charles Frey, John Frye, and Michael Souder. The impact of gps modernization on standalone user performance and integrity with araim. In *Proceedings of the 26th International Technical Meeting of the Satellite Division of the Institue of Navication (ION GNSS 2013)*, 2013.

24. J Robble. Timing anomaly detection applique, android application datasheet. Technical report, The MITRE Corporation, 2015.

25. Chronos Technology. Corporate profile. `www.chronos.co.uk/files/pdfs/corp/Chronos_Corporate_Profile.pdf`, 2014.

26. Nicholas D Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T Campbell. A survey of mobile phone sensing. *Communications Magazine, IEEE*, 48(9):140–150, 2010.

27. Khan Wazir Zada, Xiang Yang, Mohammed Y Aalsalem, and Quratulain Arshad. Mobile phone sensing systems: A survey. *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, 15(1), 2013.

28. ESRI. What is gis? `http://www.esri.com/what-is-gis/howgisworks`, 2015.

29. Peter Ranacher, Richard Brunauer, Stefan van der Spek, and Siegfried Reich. What is an appropriate temporal sampling rate to record floating car data with a gps? *isprs International Journal of Geo-Information*, 2016.

30. DoDAF Working Group. Dod architecture framework version 2.02; dod deputy chief information officer. `dodcio.defense.gov/Library/DoDArchitectureFramework.aspx`, 2015.

31. Thom Erl. Service-oriented architecture (soa) concepts, technology and design. 2005.

32. Beta Specification. Service oriented architecture Modeling Language (SoaML) Specification for the UML Profile and Metamodel for Services (UPMS). *Object Management Group OMG*, FTF Beta 2(April):1–182, 2009.

33. Tor Dietrichs, Richard Griffin, Adrian Schuettke, and Mark Slocum. Integrated architecture study for weapon borne battle damage assessment system evaluation. Capstone graduate research project, Air Force Institue of Technology, 2006.

34. Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl 3):7280–7287, 2002.

35. Aaron Burns. Accessing gps Constellation Resiliency in an Urban Canyon Enviornment. Master's thesis, Air Force Institute of Technology, 2015.

36. ExoAnalytic Solutions Inc. Seas overview. `https://www.teamseas.com/about/overview`, 2014.

37. Analytical Graphics Inc. Agi home page. www.agi.com/products/.

38. Thomas Ford, David Meyer, John Colombi, Brian Scheller, and Cody Palmer. A method of assessing time variant value of multi-domain architectures. *Joint Defense Modeling and Simulation (submitted)*, 2016.

39. LCDR. USN (Ret) David Meyer. *The CSRA Guide to Advanced Modeling, Wargaming, and Simulation*. The Air Force Institute of Technology.

40. Python Software Foundation. The python tutorial. `https://docs.python.org/3/tutorial/index.html`, 2015.

41. Rischard W. Bohannon. Comfortable and maximum walking speed of adults aged 20-79 years: reference values and determinants. Technical report, School of Allied Health, University of Connecticut, Hartford Hospital, Hartford CT, 1997.

42. U.S. Census Bureau. Population, housing units, area, and density: 2010 - county – county subdivision and place 2010 census summary file 1, 2010.

43. Jonathan Gross and Shannon Cosgrove. Crime analysis that goes beyond dots on a map. `www.esri.com/esri-news/arcnews/spring15articles/crime-analysis-that-goes-beyond-dots-on-amap`, 2015.

44. Esri news for state & local government. `www.esri.com/~/media/Files/Pdfs/library/newsletters/state-local/fall-2014.pdf`, 2014.

45. Mark Greninger. Los angeles county gis data portal, 2015.

46. Arthur Getis and J.K. Ord. The analysis of spatial association by use of distance statistics. *Geographical Analysis*, 24(3), 1992.

47. ESRI. How optimized hot spot analysis works, 2016.

48. Lu Liu, Russell Duncan, Jie Xu, David Webster, Zongyang Luo, Colin Venters, and John Davies. Modelling and simulation of network enabled capabiolity on service-oriented architecture. *Simulation Modelling Practice and Theory*, 2009.

49. Philipp Helle, Ian Giblett, and Pascal Levier. An integrated executable architecture framework for system of systems development. *Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 2013.

50. Defense Aquisition University. Joint capability areas definitions. `https://acc.dau.mil/CommunityBrowser.aspx?id=410029`, 2015.

51. World Wide Web Consortium. W3c standards. `https://www.w3.org/standards`, 2016.

52. Jeannine Brands and Thomas Drischler. Los angeles department of transportation bureau of franchise and taxicab regulation, April 2013.

53. LinQuest Corporation. Gps interference and navigation tool (giant). `http://www.linquest.com/government/products/gps-interference-and-navigation-tool`, 2015.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704–0188*

1. REPORT DATE *(DD–MM–YYYY)* 24–03–2016	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From — To)* Sept 2014 — Mar 2016

4. TITLE AND SUBTITLE A Service Oriented Architecture Approach for Global Positioning System Quality of Service Monitoring	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Stuart A. Everson	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Space and Missile Center GPS Program Office Mr. James Horejsi 483 N. Aviation Blvd. Los Angeles AFB, CA, 90245-2808 DSN 633-3226, COMM 310-653-3448 Email: james.horejsi.1@us.af.mil	**10. SPONSOR/MONITOR'S ACRONYM(S)** SMC/GPE **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This research focuses on the development of a Service Oriented Architecture (SOA) for monitoring the Global Positioning System (GPS) Standard Positioning Service in near real time utilizing a Mobile Crowd Sensing technique. This SOA development employed a Model Based System Engineering (MBSE) approach to evaluate a set of MCS architecture variants. Ultimately this research provides a feasibility analysis for utilization of mobile distributed sensors to provide situational awareness of the GPS Quality of Service (QoS). First this research provides justification for development of a new monitoring architecture and defines the scope of the SOA. Then an exploration of current SOA, MBSE, and Geospacial System Information (GIS) research was conducted. Next a Discrete Event Simulation (DES) of the MCS participant interactions was developed and simulated within AGI's Systems Toolkit. The architecture performance analysis was executed using a GIS software package known as ArcMap. Finally, this research concludes with an analysis if the proposed architecture would be suitable for detecting sources of GPS interference within an AoI.

15. SUBJECT TERMS

Service Oriented Architecture, Mobile Crowd Sensing, Geographic System Information, Model Based System Engineering, GPS Monitoring, Global Positioning System

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Lt Col. Thomas Ford, AFIT/ENV
U	U	U	U	133	**19b. TELEPHONE NUMBER** *(include area code)* (937) 255-3636, x4747; thomas.c.ford1.mil@mail.mil