INTEROPERABILITY OF DIGITAL GEOGRAPHIC INFORMATION IN THE DOMAIN OF MARITIME NAVIGATION

by

Michael R. Rushin
A Thesis
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Master of Science
Geoinformatics & Geospatial Intelligence

Committee:

_______________________________ Dr. Phil Yang, Thesis Director

_______________________________ Dr. Liping Di, Committee Member

_______________________________ Dr. Donglian Sun, Committee Member

_______________________________ Dr. Anthony Stefanidis, Department Chairperson

_______________________________ Dr. Donna M. Fox, Associate Dean, Office of Student Affairs & Special Programs, College of Science

_______________________________ Dr. Peggy Agouris, Dean, College of Science

Date: _________________________ Fall Semester 2017
George Mason University
Fairfax, VA

Interoperability of Digital Geograhic Information in the Domain of Maritime Navigation

A Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at George Mason University

by

Michael R. Rushin
Bachelor of Arts
George Mason University, 2012

Director: Professor Chaowei (Phil) Yang
Department of Geography & Geoinformatics

Fall Semester 2018
George Mason University
Fairfax, VA

PREFACE

This thesis was written to address several hurdles I encountered working as a geospatial analyst in the domain of maritime navigation. For two years I collected and applied Notice to Mariner (NTM) chart corrections to digital geographic information (DGI). I believe the topics discussed herein may serve as an analytical tool for other analysts in the field of geoinformatics and geospatial intelligence.

Michael R. Rushin

Fairfax, VA

December 2017

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Application Programming Interface .... API
Command-Line Interface CLI
Digital Geographic Information .... DGI
Digital Nautical Chart .... DNC
Electronic Navigational Chart .... ENC
Geographic Information System .... GIS
Generalized Search Tree .... GiST
Hypertext Markup Language .... HTML
Hypertext Transfer Protocol .... HTTP
Geographic Information System .... GIS
Military Sealift Command .... MSC
North Atlantic Treaty Organization .... NATO
National Geospatial Intelligence Agency .... NGA
National Oceanic and Atmospheric Administration .... NOAA
Notice to Mariners .... NTM
Open Geospatial Consortium .... OGC
Object-Relational Database Management System .... ORDBMS
Object-Relational Mapping .... ORM
OGC Web Services .... OWS
PostgreSQL .... PSQL
Raster Nautical Chart .... RNC
Uniform Resource Locator .... URL
VPF Database Update .... VDU
Vector Product Format .... VPF
Web Feature Service .... WFS
World Wide Web .... WWW
Extensible Markup Language .... XML

ABSTRACT

INTEROPERABILITY OF DIGITAL GEOGRAPHIC INFORMATION IN THE DOMAIN OF MARITIME NAVIGATION

Michael R. Rushin, M.S.

George Mason University, 2018

Thesis Director: Dr. Chaowei (Phil) Yang

This thesis examines the interoperability of digital geographic information (DGI) in the context of maritime navigation. Maritime navigation is a complex domain that draws on a wide variety of information resources. The diversity and multitude of such information creates challenges with respect to interoperability – the ability of software and hardware systems on multiple machines from multiple vendors to communicate with each other meaningfully. The Open Geospatial Consortium (OGC) is continually developing a series of standards for DGI that may provide solutions for DGI interoperability in the domain of maritime navigation. This thesis examines the OGC Web Services (OWS) and how the Web Feature Service (WFS) can be adopted to serve Notice to Mariners (NTM).

CHAPTER 1

Introduction

The National Geospatial-Intelligence Agency (NGA) produces Digital Nautical Chart (DNC) Vector Product Format (VPF) Database Update (VDU) to support worldwide DNC navigation requirements of the U.S. Navy, Military Sealift Command (MSC), the U.S. Coast Guard, and certain foreign partners. The DNC maintenance system collects new source materials such as bathymetry, imagery, NTM, local notices, new foreign chart sources, etc. for inclusion in the DNC database (Bowditch, 2017). The problem with digital geographic information (DGI) maintenance is accessing and combining information from different sources in a timely manner (Zhang, 2005). This thesis explores the use of open source software and standards to create a Web Feature Service (WFS) for local Notice to Mariners (LNM) to aid in the DNC maintenance process.

OGC Web Services (OWS)

Web services provide access to information via the World Wide Web (WWW). The service is available over the Internet, uses a standardized Extensible Markup Language (XML) messaging system, and is not dependent on operating system or software. Generally, a client sends a request over the Internet, a server receives that request, processes it, and returns a response. When a browser makes a request for a web page, it receives Hypertext Markup Language (HTML) and other related content in the response (Gassner, 2013). However, when a browser makes a request for data using the Hypertext Transfer Protocol (HTTP), a web service has been used (Li, 2011). The OGC Web Services (OWS) operate in a similar way and fills two roles: accessing remote data sources as a consumer and serving up or sharing data as a provider for others. This allows for the interoperability of data. Interoperability is defined as, "the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units (Di, 2016)."

Digital Nautical Chart (DNC)

The DNC database is made up of 29 geographic regions that provide a worldwide footprint of over 5,000 charts of varying scales resulting in global coverage between 84 degrees North and 81 degrees South (Bowditch, 2017). The 29 regions are further broken down into libraries where each library represents a different geographic area of interest and scale (Bowditch, 2017). A DNC library consists of 12 related thematic layers which are cultural landmarks, data quality, earth cover, environment, hydrography, inland waterways, land cover, limits, aids to navigation, obstructions, port facilities, and relief (Bowditch, 2017). A NTM chart correction will affect one of these layers within it's corresponding chart bounding box. For geographic regions within U.S. territorial waters, NOAA corrections are used to update DNC databases. NOAA chart corrections are referred to as local Notice to Mariners. NOAA local NTM, or LNM, are made available in text file format by NOAA and are the data used in this project.

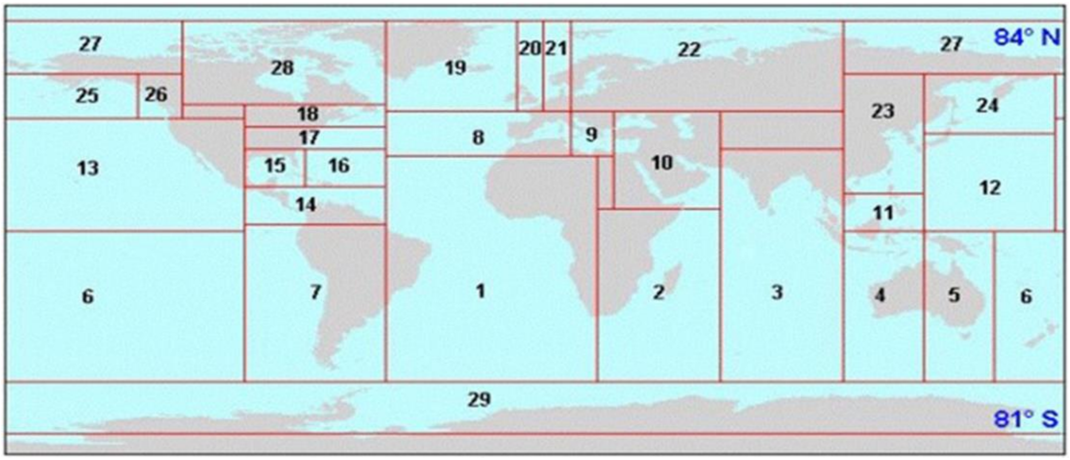

Figure *1* **DNC Regions (Bowditch, 2017)**

The manual approach to DNC maintenance at the file level is problematic for the following reasons. First, often times NTM publications from foreign agencies, as well as within the U.S. government, will utilized different semantics when describing real world phenomena (Peng, 2005). Second, a DNC update at a specific scale cannot be updated in another scale at the same location (Peng, 2005). Lastly, VDU updates make it difficult to provide feature-level data sharing over the Internet in real-time (Peng, 2005). Integrating Web services with geospatial analysis is apart of niche of researchers who are, "...composing dynamic services involving complex geospatial objects and models (Sikder, 2008)," to solve analytical problems. Similarly, this thesis concerns Web services, interoperability, and open source. The next section discusses interoperability in the domain of maritime navigation. The following section discusses the open source software used to create an OGC Web Feature Service. In the concluding section, an analysis of the implementation and a few research questions for future study.

Table *1* Local Notice to Mariners (LNM) Field Description (NOAA, 2017)

Field	Description
Chart	This column indicated the NOAA chart the NTM coordinates are within.
Action	This column indicates the charting action to be taken to update that specific feature on the NOAA chart. There are only five potential actions, Add, Delete, Relocate, Change and Substitute. If the action field contains an annotation (2) or (3), this indicates that this chart update plots on more than one panel on the same NOAA chart.
Item Name	This column indicates either the name of the aid to navigation or the feature being acted upon. In cases where the chart update affects a NOAA channel depth tabulation or a NOAA chartlet, there will be a hyper-link to a .gif image of that tabulation or chartlet for viewing and download.
Charting Label	This column indicates the charting label or charting action description that needs to be made to the NOAA chart for the specific chart update feature. For example, an aid to navigation charting label characteristics, or a description of hazard to navigation. This is the charting label after the update is made.
Latitude	This column indicates the published latitude in Degrees, Minutes, Seconds (DMS). This is the latitude after the update is made.
Longitude	This column indicates the published longitude in DMS. This is the longitude after the update is made.
Latitude (Decimal Degrees)	This column indicates the published latitude out to three decimal places. This is the latitude after the update is made.
Longitude (Decimal Degrees)	This column indicates the published longitude out to three decimal places. This is the longitude after the update is made.
Published Document	This column indicates the USCG LNM, NGA NM or CNM that the chart correction was published in, and the USCG District where the feature is located. NOAA cartographers have evaluated and applied this information to the nautical chart. In cases where the phrase "Not Yet Published" is listed for the feature, this indicates that the chart correction was identified by NOAA, but has not yet been published in a USCG LNM or NGA NM. In cases where the phrase "Not Published" is listed for the feature, this indicates that the chart correction was not published in a Notice to Mariners. In cases where the phrase "Unknown" is listed for the feature, this indicates that publishing information for this chart correction isn't available.
KAPP	About half of the charts consists of a single panel - The Main Panel. The other half also include supplemental panels - Insets, Extensions, etc. All the different types of panels are known as "kapps" and each has a unique kapp number.
Raster Nautical Chart (RNC) Panel	This is the unique numbering for the chart panels that is used for the RNC in BSB format.
RNC Posted Date	This column indicates when the correction was made available on the RNC in BSB format.

Interoperability

Interoperability is the ability to share data and other resources among different systems (Yao, 2008). The need for interoperability came from the ever existing heterogeneity of digital geographic information. Heterogeneity is defined as, "The quality or state of being diverse in character or content. Heterogeneity occurs in many forms, ranging from the hardware and software platform that a database system is based on, to the data model and schema used to provide logical structure for the stored data, to the very kinds of data and information that are being stored (ISO, 2014)." This is simplified as system, syntactic, structural, and semantic heterogeneity (ISO, 2014). The heterogeneity of DGI in the domain of maritime navigation is primarily structural or semantic. This means that DGI for maritime navigation is different and not interoperable due to the differences related to the conceptual modeling of the data (ISO, 2014). For example, the National Oceanic and Atmospheric Administration's (NOAA) Electronic Navigational Chart is formatted to the International Hydrographic Organization's S-57 standard (IHO, 2000) and the National Geospatial Intelligence Agency's (NGA) Digital Nautical Chart is formatted to the North Atlantic Treaty Organization (NATO) Standardization Agreement (STANAG) Digital Geographic Information Exchange Standard (DIGEST) Vector Product Format (VPF) (NGA, 1997). Singularly, differences in structural heterogeneity often occur with differences in meaning between concepts and data used to represent geographic features (ISO, 2014). For example, a bridge may be stored as a line or polygon depending on the standard. Great efforts have been made to make these data types and others interoperable. However, this thesis does not attempt to solve these issues directly but looks to existing open source softwares that may aid in the maintenance of digital maritime geographic information through the application of OGC standards, specifically NTM, which is one aspect of maritime navigation.

Open Source Software

Django is a high-level Python Web Development Framework that encourages rapid development and simple design (Django, 2017). GeoDjango is the spatial counterpart to Django and allows for the creation of GIS Web applications (GeoDjango, 2017). The NTM Web Feature Service (WFS) uses GeoDjango and is built upon different models that contain the essential fields of the NTM data and other applications necessary for the web service. The Django documentation (Django, 2017) states the Django model is, "...an object-relational mapper in which you describe your database layout in Python code." The field descriptions of an NTM provided by NOAA are used to build a Python subclass of the Django API database model.

```
# -*- coding: utf-8 -*-
from __future__ import unicode_literals

from django.contrib.gis.db import models

class NTM(models.Model):
    chart = models.CharField(max_length=1000)
    action = models.CharField(max_length=1000)
    itemname = models.CharField(max_length=1000)
    chartingla = models.CharField(max_length=1000)
    latitude = models.CharField(max_length=1000)
    longitude = models.CharField(max_length=1000)
    latdd = models.CharField(max_length=1000)
    longdd = models.CharField(max_length=1000)
    publishedd = models.CharField(max_length=1000)
    kapp = models.CharField(max_length=1000)
    rncpanel = models.CharField(max_length=1000)
    rncposted = models.CharField(max_length=1000)
    geom = models.PointField(srid=4326)
    objects = models.GeoManager()

    # Returns the string representation of the model.
    def __str__(self):              # __unicode__ on Python 2
        return self.itemname

    class Meta:

        verbose_name_plural = "NTMS"
```

Figure *2* Django NTM Model

The NTM model and others are then migrated, or translated into a database to hold the data from which a Web service will be created. PostgreSQL (PSQL) is an Object-Relational Database Management System (ORDBMS) and is widely considered to be the most advanced open source database system in the world (Worsely, 2002). PSQL was the ORDBMS used for this project along with the PSQL spatial extension, PostGIS. PSQL PostGIS supports vector, raster, TIN and is ideal for geospatial web services (Parr, 2015). Using the Django command-line utility, the NTM WFS models can be migrated to a PSQL database automatically.

```
$ CREATE DATABASE thesis;
$ CREATE EXTENSION postgis;
$ CREATE EXTENSION postgis_topology;
```

Figure 3 PSQL Database Creation

Acquiring the NTM data required the creation of a data mining Python script. The URL for each NTM text file, organized by chart, is stored in a Python list data structure. The list is then processed, downloading each text file, cleaning the data noise, and converting to an ESRI shapefile format. Each key in the NTM ORM dictionary corresponds to a field in the NTM model and the value of each dimension is the same name of the shapefile field that the data will be loaded from. Once the data has been preprocessed and formated the individual NTM features can be loaded into the database.

```python
import os
import sys
import codecs
sys.stdout = codecs.getwriter("iso-8859-1")(sys.stdout, 'xmlcharrefreplace')
from django.contrib.gis.utils import LayerMapping
from models import NTM

### Auto-generated `LayerMapping` dictionary for NTM model
ntm_mapping = {
    'chart' : 'Chart',
    'action' : 'Action',
    'itemname' : 'ItemName',
    'chartingla' : 'ChartingLa',
    'latitude' : 'Latitude',
    'longitude' : 'Longitude',
    'latdd' : 'LatDD',
    'longdd' : 'LongDD',
    'publishedd' : 'PublishedD',
    'kapp' : 'Kapp',
    'rncpanel' : 'RNCPanel',
    'rncposted' : 'RNCPosted',
    'geom' : 'POINT',
}

ntm_shp = os.path.abspath(
    os.path.join(os.path.dirname(__file__), 'data', 'NTM.shp'),
)

def run(verbose=True):
    lm = LayerMapping(
        NTM, ntm_shp, ntm_mapping,
        transform=False, encoding='iso-8859-1',
    )
lm.save(strict=True, verbose=verbose)
```

Figure 4 Django NTM ORM Dictionary

This is how the data is manage on a local machine. However, to make the data accessible in almost real-time it must be deployed to the World Wide Web (WWW). The Heroku Internet service and the Heroku Command-Line Interface (CLI) allows developers to deploy, run, and manage applications. The collection of source code for the NTM WFS and the application software dependencies, stored in the requirements text file, provides Heroku everything needed to make an active website. Heroku uses the distributed version control system Git to deploy applications. Therefore, similar to pushing source code to GitHub, source code is pushed to Heroku where the application can be deployed. Once the source code has been pushed to Heroku, the Heroku PSQL database must also be migrated, spatial extensions must be added using Heroku Buildpacks, and a domain name should be assigned. Buildpacks are scripts that vendor main Python spatial libraries like Geos, Proj, and GDAL. When Heroku runs an application it installs the necessary libraries outlined in the requirements text file and often meets errors due to system differences. Buildpacks solve this problem and allows Heroku to run a Python application identically to the local machine. The Heroku-Geo-Buildpack allows for the deployment of GeoDjango applications and is necessary to run the Django WFS over the WWW.

CHAPTER 2

Django Web Feature Service (WFS)

The OGC Web Feature Service is officially defined as a service allowing, "... a client to retrieve and update geospatial data encoded in Geography Markup Language (GML) from multiple Web Feature Services (Vretanos, 2002)." The Django WFS implementation by Vasco Pinho loosely follows the OGC Web Feature Service Implementation Specification Version 1.0.0 and is successful through use of the Django view function. The Django view function is defined as, "...a Python function that takes a Web request and returns a Web response. This response can be the HTML contents of … an XML document...(Django, 2017)".

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:FeatureCollection
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs http://schemas.opengis.net/wfs/1.0.0/WFS-basic.xsd">
   {% for ftype, feature in features %}
     {% if feature.gml %}
       <gml:featureMember>
         <{{ ftype.name }} fid="{{ ftype.name }}.{{ feature.id }}">
           {{ feature.xml|safe }}
         </{{ ftype.name }}>
       </gml:featureMember>
     {% endif %}
   {% endfor %}
</wfs:FeatureCollection>
```

Figure *5* **Django WFS XML Example**

The Django WFS uses the Django view to declare an operation and return it's corresponding XML template. The XML template utilizes the OGC WFS name space and OGC WFS XML schema and encodes the data in GML. GML is used to communicate the NTM data over the WWW and promotes interoperability (Peng, 2004). However, the Django WFS only offers the GetCapabilities, DescribeFeatureType, and GetFeature operations which is considered a READ-ONLY WFS or Basic WFS (Vretanos, 2002). The GetCapabilities operation describes the capabilities of the WFS, the DescribeFeatureType operation describes the structure of the feature type it can service, and the GetFeature operation retrieves feature instances. The Basic WFS operations allow for an maritime geospatial analyst to layer NTM chart corrections automatically; saving time and reducing error because the coordinates are no longer manually entered. The OGC Web Services (OWS) and other emerging developments in distributed geoinformatics offer new analytical tools which can reshape the spatial data infrastructure (SDI) of maritime navigation.

Table 2 Django Web Feature Service (WFS) Operations

Operation	Uniform Resource Locator (URL)
GetCapabilities	http://www.michaelrrushin.com/1/?service=wfs&request=getcapabilities
DescribeFeatureType	http://www.michaelrrushin.com/1/?service=wfs&request=describefeaturetype
GetFeature	http://www.michaelrrushin.com/1/?service=wfs&request=getfeature&typename=ntm

Analysis

A complete functioning website was created to visualize and serve NTM data. The web GIS section of the website is created with Leaflet. Leaflet is a JavaScript (JS) library for interactive maps (Leaflet, 2017). The Leaflet JS library can be declared in the head portion of an HTML file and depending on the structure of the document, a GIS window can be initiated. The map window contains a legend showing three base map options of OpenStreetMap, OpenTopoMap, and Raster Nautical Cart (RNC). There can be an unlimited number and type of base map added to a Leaflet window. The only requirement is the tiles are 256 x 256 pixel PNG files, each zoom level is a directory, each column is a subdirectory, each file in that column is a file, and the URL format is zoom, X, and Y. The layers section of the table of contents shows NTM and Clusters. The NTM collection is each individual NTM and the Cluster section shows clusters at different scales with the number of NTM in each cluster. When the user clicks on the individual NTM the feature attributes will show in a pop-up window. Also, when a user clicks the Download NTM button, the PSQL database is populated with the newest NTM published by NOAA.

Conclusion

This thesis shows Python, Django, GeoDjango, and PSQL can be used to create a web GIS for atmospheric and oceanographic data sciences utilizing OGC implementations specifications (Woolf, 2005). Similar and future research is being done with a DjangoREST Framework implementation of the OGC WFS. Django REST framework is a Django toolkit for building Web APIs. The toolkit offers developers the ability to serialize data in a geographic format such as GML. More importantly, DjangoREST framework has long term support (LTS) and is used and recognized internationally by companies such as Mozilla, Red Hat, Heroku, and Eventbrite (Christie, 2017).

REFERENCES

Bowditch, N. (2017). *Pub. no. 9 American practical navigator: an epitome of navigation.* Springfield, VA: National Geospatial-Intelligence Agency.

Christie, T. (2017). Django REST Framework. Retrieved August 31, 2017, from http://www.django-rest-framework.org/

Di. L., & Zhao, P. (2016). Geospatial Semantic Web, Interoperability. *Encyclopedia of GIS,* 398-403. doi:10.1007/978-3-319-23519-6_516-2

Django Documentation (2017). Retrieved August 31, 2017, from https://www.djangoproject.com/start/overview/

Gassner, D. (2013, August 21). Programming Foundations: Web Services [Video file]. Retrieved from https://www.lynda.com/

GeoDjango Documentation (2017). Retrieved August 31, 2017, from https://docs.djangoproject.com/en/1.11/ref/contrib/gis/

IHO (2000). S-57 IHO Transfer Standard for Digital Hydrographic Data (3rd ed.). Monaco: International Hydrographic Bureau. Retrieved from https://www.iho.int/iho_pubs/standard/S-57Ed3.1/31Main.pdf

ISO (2014). ISO 19101-1:2014 Geographic information - Reference model – Part1: Fundamentals, International Organization for Standardization (ISO)

Leaflet Documentation (2017). Retrieved August 31, 2017, from http://leafletjs.com/reference-1.2.0.html

Li, Z. Yang, C.P., Wu, H., Li, W., & Miao, L. (2011). An optimized framework for seamlessly integrating OGC Web Services to support geospatial sciences. *International Journal of Geographical Information Science*, 25(4), 595-613. doi:10.1080/13658816.2010.484811

NGA (1997). DNC Performance Specification – Digital Nautical Chart MIL-PRF-89023. *National Geospatial Intelligence Agency.* Retrieved November 17th, 2017, from http://earth-info.nga.mil/publications/specs/printed/89023/89023.pdf

NOAA. (2017). NOAA Office of Coast Survey. *U.S. Department of Commerce.* Retrieved November 17th, 2017, from https://ocsdata.ncd.noaa.gov/ntm/Info.aspx

Parr, D.A., & Scholz, M. (2015). Building a Low-Cost Geographic Website for Collecting Citizen Science Contributions. *Papers in Applied Geography*, 1(3), 205-211. doi:10.1080/23754931.2015.1014263

Pinho, V. (2015). Django WFS. Retrieved August 31, 2017, from https://github.com/vascop/django-wfs

Peng, Z., & Zhang, C. (2004). The roles of geography markup language (GML), scalable vector graphics (SVG), and Web feature service (WFS) specifications in the development of Internet geographic information systems (GIS). *Journal of Geographical Systems*, 6(2). doi:10.1007/s10109-004-0129-0

Peng, Z. (2005). A proposed framework for feature-level geospatial data sharing: a case study for transportation network data. *International Journal of Geographical Information Science*, 19(4), 459-481. doi:10.1080/13658810512331319127

Sikder, I. U., Gangopadhyay, A., & Shampur, N. V. (2008). Interoperability in Web-Based Geospatial Applications. I*nternational Journal of Information Technology and Web Engineering*. 3(3), 66-88. doi:10.4018/jitwe.2008070105

Vretanos, P.A. (Ed.) (2002). Web Feature Service Implementation Specification (1.0.0). *Open GIS Consortium Inc*. http://portal.opengeospatial.org/files/?artifact_id=7176

Woolf, A., Cramer, R., Gutierrez, M., Dam, K. K., Kondapalli, S., Latham, S., . . . Oneill, K. (2005). Standards-based data interoperability in the climate sciences. *Meteorological Applications, 12*(1), 9-22. doi:10.1017/s1350482705001556

Worsely, J.C., & Drake, J.D. (2002). *Practical PostgreSQL*. Sebastopol, Calif: O Reilly.

Yao, X., & Zou, L. (2008). Interoperable Internet Mapping—An Open Source Approach. *Cartography and Geographic Information Science*, 35(4), 279-293. doi:10.1559/152304008786140560

Zhang, C., & Li, W. (2005). The Roles of Web Feature and Web Map Services in Real-time Geospatial Data Sharing for Time-critical Applications. *Cartography and Geographic Information Science*, 32(4), 269-283. doi:10.1559/152304005775194728

BIOGRAPHY

Michael Rushin graduated from Robert E. Lee High School, Springfield, Virginia, in 2007. He received his Bachelor of Arts from George Mason University in 2012. He received his Graduate Certificate in 2014. He was employed as a Geospatial Analyst in the private sector for two years and received his Master of Science in Geoinformatics & Geospatial Intelligence from George Mason University in 2017.