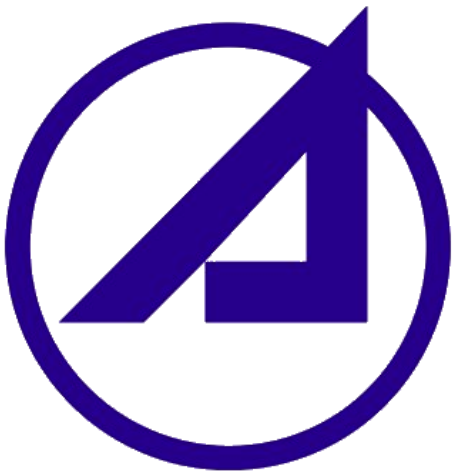
Senior Design Final Report

Aerospace Corporation



Team Members:

Richard Bailon

Xico Blanco

Yuridia Ginez

Nathan Gonzales

Andrew Jarmin

William Leung

Pedro Eduardo Ramirez

Cesar Salazar

Aaron Simental

Scott Yadong Sun

Faculty Advisor:

Dr. Zilong Ye

Liaisons:

Andre Chen

Denny Ly

Karina Martinez

Pablo Settecase

Table of Contents

1. Introduction **2**
1.1. Background 2
1.2. How Our Project Stands Out 2
1.3. Achievements 3
2. Related Technologies **4**
2.1. Existing Solutions 4
2.2. Reused Products 4
3. System Architecture **5**
3.1. Overview 5
3.2. Data Flow 6
4. Conclusions **7**
4.1. Results 7
4.2. Future 7
5. References **8**

1. Introduction

1.1. Background

In September of 2022, Aerospace Corporation approached our team of engineers with a problem to solve. They needed a solution for visualizing Dilution of Precision (DOP) data without much human intervention. DOP is a measure of how well a group of GPS satellites' signal strength is, and Aerospace Corporation currently has the capability to perform this visualization. However, the process they use is manual and time-consuming, requiring their engineers to create correctly formatted files to be analyzed by a program that produces a user interface with the DOP values.

The manual pipeline currently in place at Aerospace Corporation requires significant human intervention, which results in slow turnaround times and a high potential for error. In addition, their visualization tools are antiquated and require significant processing power, which further slows down the process.

The team of engineers tasked with solving this problem are working on creating an automated solution that can take the raw satellite data, analyze it, and produce a user-friendly visualization without the need for expert intervention. The goal is to streamline the process, reduce errors, and create a more efficient and user-friendly system that can be used by a broader audience.

The engineers are exploring various technologies and solutions that can help them achieve this goal. The hope is that by automating the process and implementing more modern tools, Aerospace Corporation can significantly improve their visualization capabilities and better serve their clients.

1.2. How Our Project Stands Out

Our project stands out in several ways. The ability to generate DOP analysis from start to finish with the push of a button is a significant advantage over traditional methods, which may require significant time and effort. This can improve efficiency and reduce the potential for errors. The fact that our team was able to produce DOP analysis for not just NAVSTAR but six foreign satellite constellations, those constellations:

- GALILEO, a European Union (EU) and European Space Agency (ESA) satellite constellation
- CHINASAT, Chinese satellite constellation
- GLONASS, Russian satellite constellation
- IRNSS, Indian Regional Navigation Satellite System
- QZSS, Japanese satellite constellation

This demonstrates a high level of expertise in the field and a willingness to take on challenging tasks. Our project has several unique features that set it apart from others in the field and can provide significant benefits to organizations that rely on DOP analysis.

1.3. Achievements

Our team consisted of 10 members who were divided into four sub-teams, each tasked with solving a smaller problem that would contribute to the overall solution. The first team was responsible for gathering all of the satellite data. They leveraged the Space-Track.org API to collect the necessary data for the 180 satellites that our product focused on. According to Space-Track.org, “Space-Track.org's Application Programming Interface (API) allows users to access data on this site programmatically using custom, stable URLs with configurable parameters.” The team needed to ensure that they were pulling the correct data for each satellite.

The next group spent significant time reverse-engineering how files generate DOP values. They developed a way to pass our previously obtained satellite data through the program creating a new file to generate DOP values. This required a great deal of careful attention to detail, as even the slightest error in calculation could cause significant problems down the line.

After obtaining the DOP values, the third group was tasked with uploading this data into our database in an efficient manner. This was no small task, as the DOP values for all 180 satellites make up approximately 1GB of important data. The team developed a system that would reliably and quickly upload this data without errors or data loss.

Lastly, our data visualization group was tasked with pulling the necessary data from our database and visualizing the DOP values. This was a crucial step, because if this part of the process didn't work correctly, all of our hard work would have been for nothing. The team worked hard to develop a user-friendly and visually appealing interface that would allow users to analyze the DOP of a constellation of GPS satellites with one press of a button.

Throughout the eight months of development, our team showed tremendous dedication to getting all of the individual solutions to work together autonomously, without the need of aerospace professionals. Our four smaller solutions now work together seamlessly, allowing anyone to analyze the DOP of a group of GPS satellites with ease. We are proud of the work we have accomplished and believe that our solution will be a valuable tool for those in the aerospace industry and their stakeholders.

2. Related Technologies

2.1. Existing Solutions

Our project involved the development of a DOP analysis web app that is able to automatically perform the full DOP generation pipeline (reference **Figure 3.1**). To the best of our knowledge, there were no existing solutions available that are able to perform this task automatically. Therefore, we took on the challenge of creating a solution that would streamline the process and save time and effort for those who require DOP analysis.

To achieve this, we developed our own custom python scripts and database to populate and store the DOP results. Additionally, we leveraged Aerospace's SOAP program to produce the DOP analysis. Satellite Orbit Analysis Program (SOAP) is a graphical user interface developed by The Aerospace Corporation. SOAP is capable of performing more than 150 different analysis types. By combining these technologies and building our own custom components, we were able to create an efficient and reliable DOP analysis web app that was capable of performing the full DOP generation pipeline automatically.

2.2. Reused Products

In regards to reused technologies, it is important to note that our group decided to leverage aerospace's SOAP program to produce the DOP analysis. By using this existing technology, we were able to streamline the development process and reduce the overall workload of our developers.

During the development phase, we used various websites to quickly test the results produced by our pipeline. We leveraged geojson[.]io (geojson) to plot the DOP values that we stored in our database, as our web app was still under development. An example of United State's NAVSTAR constellation DOP values using geojson can be seen in **Figure 2.2** below.

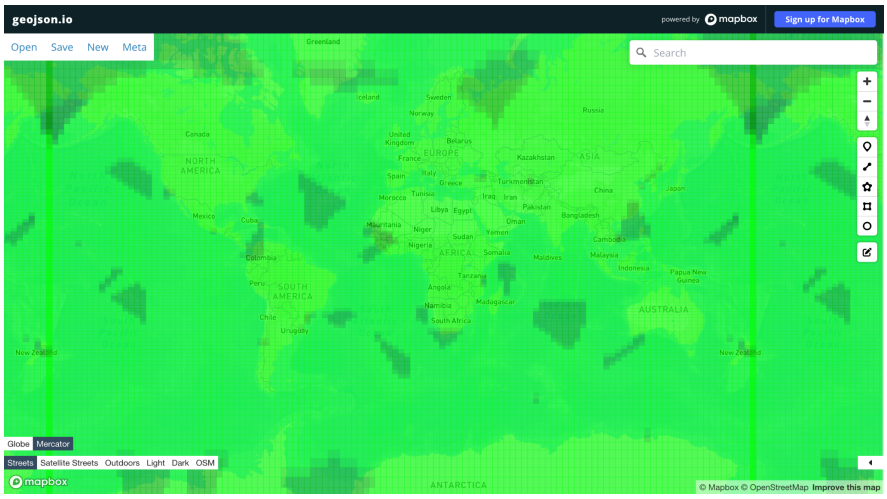


Figure 2.2: *NAVSTAR Constellation Geojson View*

We also utilized Docker technology in our project to ensure that our application runs consistently across different environments. Docker allowed us to package our application and all of

its dependencies into a container, which can be easily deployed on any machine with Docker installed. This helped us to avoid issues caused by differences in operating systems, libraries, or other dependencies. Using Docker also made it easier for us to manage and scale our application, as we could easily spin up new containers and distribute the workload. Docker was an essential tool in our project, enabling us to streamline our development and deployment processes and ensure the reliability of our application.

Other technologies were used to achieve our goals, some of which were considered trivial in their application. For instance, we utilized Python for all of our data pulling, parsing, and formatting needs. Additionally, we used InfluxDB as a database, which is a popular time-series database designed to handle high write and query loads. Lastly, for visualization, we used Node.js.While these technologies may be considered trivial in their application, they played a crucial role in the success of our project.

3. System Architecture

- Describe system architecture. (Include DFD graph).
- Chart signal workflow or the data exchange flow between different software components.
- Software/Algorithm/System Development and Implementation.
    - Describe implementation steps and details.
    - Describe algorithms implemented.

3.1. Overview

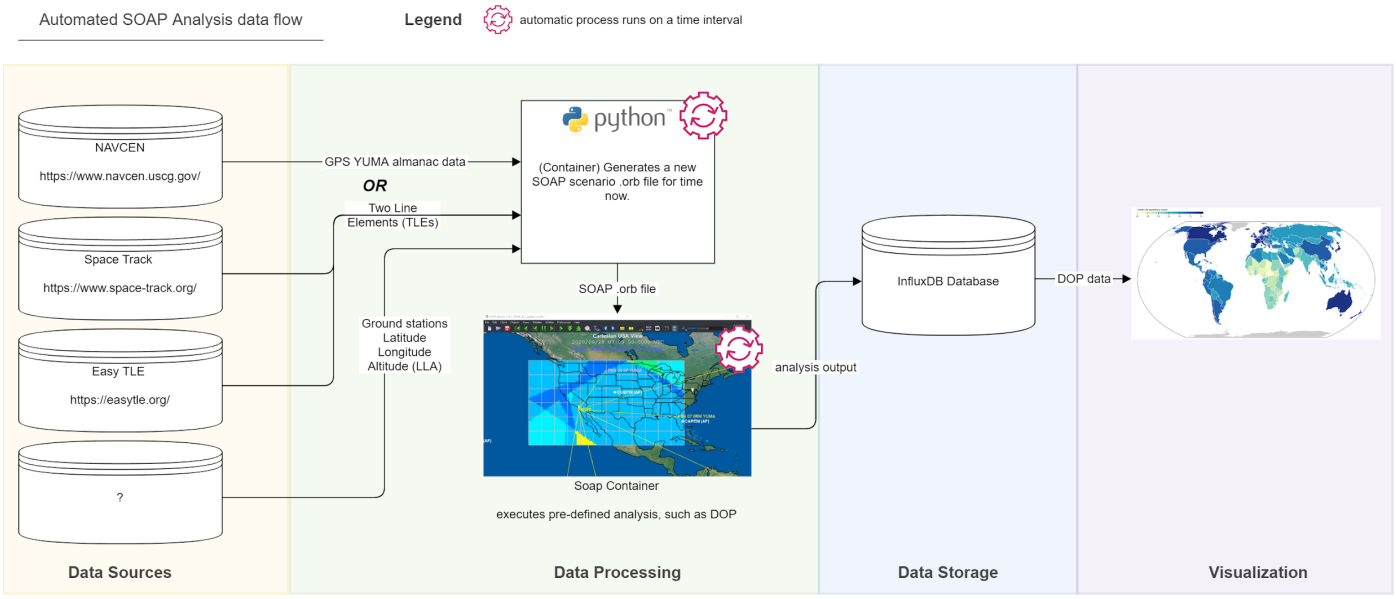


Figure 3.1 : *Pipeline*

Our team consisted of 10 members and was divided into four sub-teams, each assigned to a specific problem: data sources, processing, storage, and visualization, as shown in **Figure 3.1** above.

3.2. Data Flow

The first team (Backend Engineers) was responsible for gathering all of the satellite data. They leveraged space-track API data found online to collect the necessary data for the 180 satellites that our product focused on. It was essential that the team ensured that they were pulling the correct data for each satellite.

The next group (Data Formatters) spent significant time reverse-engineering how files generate DOP values. They developed a way to pass our previously obtained satellite data through the program creating a new file to generate DOP values. This required a great deal of careful attention to detail, as even the slightest error in calculation could cause significant problems down the line. Once this file is complete, it is executed through SOAP, automatically generating hundreds of DOP values along with their longitude and latitude.

After obtaining the DOP values, the third team (Database Engineers) was tasked with uploading this data into our InfluxDB database in an efficient manner. This was no small task, as the DOP values for all 180 satellites make up approximately 1GB of important data. The team developed a system that would reliably and quickly upload this data without errors or data loss.

Lastly, our data visualization team (Frontend Engineers) was tasked with pulling the necessary data from our database and visualizing the DOP values. This was a crucial step, because if this part of the process didn't work correctly, all of our hard work would have been for nothing. The team worked hard to develop a user-friendly and visually appealing interface that would allow users to analyze the DOP of a constellation of GPS satellites with one press of a button.

Our four smaller solutions now work together seamlessly, allowing anyone to analyze the DOP of a group of GPS satellites with ease.

4. Conclusions

4.1. Results

Our team achieved several significant successes during the project. Firstly, we were able to automate the entire DOP generation process, which was the main goal of this project. Secondly, we successfully retrieved open-source satellite data from SpaceTrack, which was a crucial component of the project. Thirdly, we automated the generation of .orb file creation. Fourthly, we generated .dtt files and uploaded the .dtt produced data to Influx DB, which required considerable technical expertise. Finally, we successfully plotted the data on a web app that was made in-house, which created an easy-to-use interface for displaying the DOP results.

We also encountered several challenges. One of the biggest challenges was ensuring the accuracy of the data generated through the automated processes. It required a lot of attention to detail and rigorous testing to ensure that the data was reliable. Another challenge was designing and implementing the web app to display the results effectively. This required a combination of technical skills and creativity to create a user-friendly interface that could handle large amounts of data.

Although our team did not fail, we did encounter some difficulties during the project. One of the biggest challenges was reverse engineering the Aerospace SOAP program. We had a difficult time understanding how to initially pass values through our Python script to SOAP. This required ample research and testing to understand the underlying architecture and programming language. Although this was a significant challenge, we were eventually able to overcome it through collaboration and persistence.

Overall, our team achieved many important milestones during the project, including automating several processes and generating useful data that could be accessed through a web app. Although we encountered several challenges, we were able to overcome them through careful planning, attention to detail, and teamwork. The project was a great success, and we believe that it will have important applications in the future.

4.2. Future

In the future, it will be crucial for developers to prioritize the scalability, efficiency, and functionality of our project. To achieve this, we have already decided to implement several important additions, such as pushing our product to a hosted server and creating one central place for the satellite data. These changes will allow more users to easily access and use our project, while also enabling us to better manage the data and avoid any inconsistencies or errors that may arise from having multiple copies of the same information.

As we continue to grow and expand, it will be essential to further improve the data pipeline to handle larger volumes of data and meet the growing demands of our users. This can be achieved through various measures, such as optimizing the data ingestion process, improving data processing algorithms, and implementing more robust data storage and retrieval mechanisms. By focusing on

these aspects, we can ensure that our project provides a more reliable and consistent experience for our users, which is crucial for achieving long-term success.

Another feature that would be beneficial to work on would be a DOP compression algorithm for the visual side. Currently our front end plots 15,000 DOP rectangular polygons on a world map, although this method works it is computationally intensive on the users machine. Our data engineers proposed a solution that would help fix this problem, DOP compression. Essentially it boils down to a simple recursive problem with these following steps:

- Select a starting DOP (Center)
- Check if the DOP above is the same
    - If so merge the two polygons into one
    - If they aren't the same ignore
- Check if the DOP below is the same DOP
    - If so merge the two polygons into one
    - If they aren't the same ignore
- *... left polygon and right polygon*
- Do this recursively until the entire world grid has been compared

We essentially want to find polygons of the same value that happen to be next to each other and combine those to create large shapes of the same value.

Figure 4.2 shows a visual of what the DOP compression algorithm should aim to do, each world map has different DOP values. That means that the DOP compression algorithm won't always compress the data to a minimum value but the algorithm in most cases should greatly reduce the number of polygons stored in the world view, leading to a faster user experience and less data having to be stored.

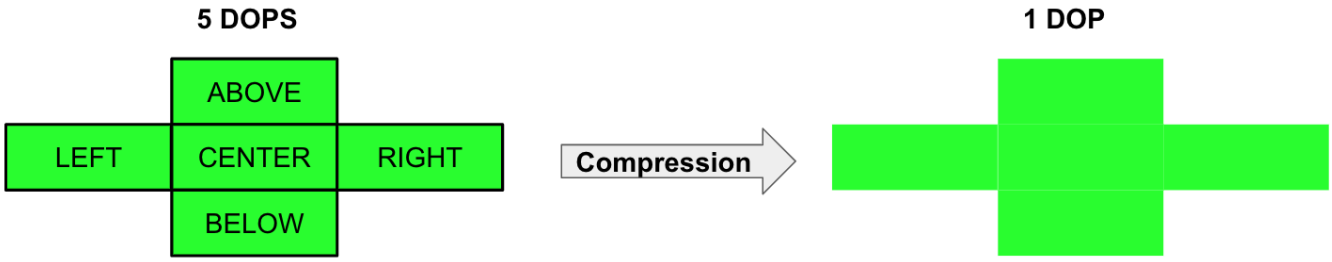


Figure 4.2 : *Compression*

5. References

SpaceTrack. (n.d.). Home. Retrieved April 5, 2023, from https://www.space-track.org/

GeoJSON. (n.d.). The GeoJSON Format. Retrieved April 5, 2023, from https://geojson.io/

InfluxData. (n.d.). InfluxData | The Modern Time Series Platform Built for Metrics and Events. Retrieved April 5, 2023, from https://www.influxdata.com/

Docker. (n.d.). Docker - Build, Ship, and Run Any App, Anywhere. Retrieved April 5, 2023, from https://www.docker.com/

Aerospace Corporation. (n.d.). Software & Specialized Applications. Retrieved April 5, 2023, from https://aerospace.org/software-specialized-applications