Software Design Document
for
Aerospace Dilution of Precision Automation

Version 2.0 approved

Prepared by: Aaron Simental, Andrew Jarmin, Cesar Salazar, Nathan Gonzales, Pedro Ramirez, Richard Bailon, Scott Sun, William Leung, Xico Blanco, Yuridia Ginez

CSULA/ The Aerospace Corporation

May 12th , 2023

Revision History

Name	Date	Reason For Changes	Version
	11/25/2022	First Draft	1.0
Team	04/07/2023	Updating Document	2.0
Team	05/12/2023	Final Draft	2.0

1. Introduction

1.1 Purpose

The purpose of this document is to:

1. Describe the intended use of the Dilution of Precision Automation (DOPA) Web Application
2. Provide the audience with important information about the technological demands of DOPA
3. Break down the internal works of the web app and explain the thought process and concepts adopted to develop DOPA
4. Promote and explain the current as well as future features of DOPA

1.2 Document Conventions

The Bolding titles indicate a new section and shall demonstrate relevant information below it.

1.3 Intended Audience and Reading Suggestions

The document is for the following:

- The Aerospace Corporation to understand and learn more about the development process of DOPA.
- End-users of the web application that are willing to learn more about the project

1.4 System Overview

DOPA is used to automate the process of obtaining Postion Dilution of Precision (PDOP) values from satellites and visualizing on a web application. Users would be able to query the time and date of the PDOP values on the web application.

2. Design Considerations

2.1 Assumptions and Dependencies

I. Related software or applications:
    A. Docker
    B. SOAP
    C. Space-Track.org
    D. Python
II. Operating Systems:
    A. Windows
    B. macOS

2.2 General Constraints

- Extremely limited to no availability to those unassociated with Aerospace
- SOAP security clearance needed from Aerospace
- Requires a network communication to Space Track

2.3 Goals and Guidelines

- The web application should display DOP values in a world map.
- The goal is to complete the project by the end of the Spring 2023 semester. The team shall commit all of our documentation and repository to Aerospace.
- Web application should be user-friendly and accurately display PDOP data.
- Data from Space-Track.org should be accurate

2.4 Development Methods

The project was broken down into four sub-problems. Each group would be assigned to a problem to which they would individually take on the task and eventually come together to discuss the methods used, learned information, and possible steps that could lead to the end goal. The team used Agile Development and Trello to maintain and monitor project progress.

3. Architectural Strategies

I. Programming languages and products used:
   A. Python
   B. JavaScript
   C. Docker
   D. SpaceTrack API
   E. InfluxDB
   F. Flask
   G. D3.js

II. Reuse of existing software components to implement various parts/features of the system:
   A. SOAP is an internal tool from The Aerospace Corp. provided to us to utilize.

III. User interface paradigms (or system input and output models):
   A. Originally Grafana was used in the beginning of the project. The team has chosen to switch to D3.js for more customizable world maps.

4. System Architecture

The DOPA project pipeline was provided to the CSULA team by The Aerospace Corporation.

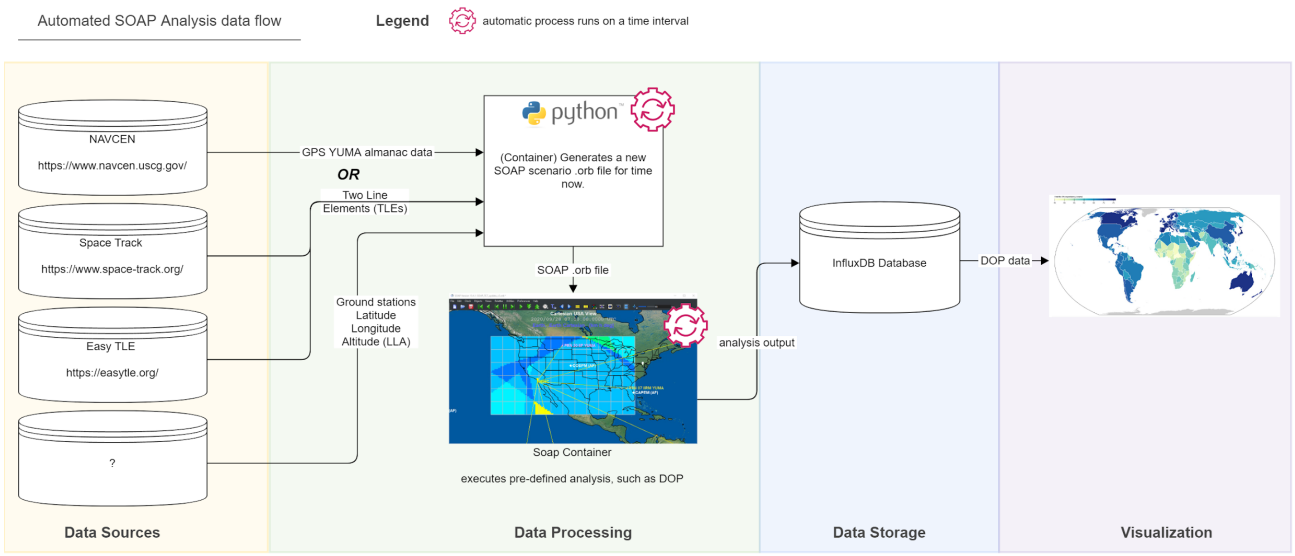

Figure 1 Data Flow Diagram

I. Data Processing
   A. Orb Generation
      1. SpaceTrack.org API is used to obtain satellite data in TLE format
      2. TLE files are formatted into .orb files
   B. SOAP
      1. Process the .orb files and produces analysis output containing PDOP data

II. Data Storage
   A. InfluxDB
      1. Stores the PDOP data into the database

III. Visualization / Web Application
   A. Flask & D3.js
      1. Obtains data from InfluxDB
      2. Visualizes PDOP on a world map

5. Policies and Tactics

5.1 Choice of which specific products used

DOPA mainly uses Docker and Python. Visual Studio Code and Jupyter Notebook was the main IDE used during development. Space Track API provides the satellite data needed for the project. InfluxDB was the database of choice for storing the time-series format of the analysis output. D3.js was chosen for its variety of ways to display data using JavasScript.

5.2 Plans for ensuring requirements traceability

Ensuring project requirements are met by following the project pipeline and project proposal provided by The Aerospace Corporation.

5.3 Plans for testing the software

DOPA retrieves real-time satellite data from Space Track API and goes through the data pipeline testing each component ultimately showing up on the web application for visualization.

5.4 Interfaces for end-users, software, hardware, and communications

The interface will be the web application. The web app will feature user input for the time and date of the PDOP data available from the database to visualize.

5.5 How to build and/or generate the system's deliverables (how to compile, link, load, etc.)

Downloadables are available on DOPA’s project github. To compile and run, use the Docker compose file included.

6. Detailed System Design

6.1 Data Gathering

6.1.1 Responsibilities

The goal is to obtain satellite positioning data from the Space Track API and create an ORB file. The Data Processing is responsible for creating a Python script to interact with the Space Track API, formatting the ORB file, and containerize the process.

6.1.2 Constraints

- TLE data should only be requested no more than once per hour from the API.
- The output should only be the ORB file for SOAP.

6.1.3 Uses/Interactions

The ORB files produced by this team will be passed to the Data Processing.

6.2 Data Processing

6.2.1 Responsibilities

- Containerize the SOAP application.
- Run the ORB files in the SOAP scenarios through the container.
- Generate analysis output in DTT files.

6.2.2 Uses/Interactions

- Analysis output data is passed to the Data Storage.

6.3 Data Storage

6.3.1 Responsibilities

- Store the analysis output data obtained from the Data Processing into InfluxDB.
- Create a way to query data from InfluxDB.

6.3.2 Constraints

- Data storage should only be on local machine.

6.3.3 Uses/Interactions

- The Visualization obtains the DOP data from InfluxDB

6.4 Visualization / Web Application

6.4.1 Responsibilities

- Visual PDOP data obtained from InfluxDB on world map.
- Create a user friendly web application to interact with PDOP data visuals.

6.4.2 Constraints

- Retrieving and displaying data from InfluxDB should be quick.

7. Database Design

Influx Database version 2.7 is used to store PDOP data from SOAP. The epoch is formatted into InfluxDB as a timestamp and is used to index the PDOP data. We are storing six unique satellites under the 'measurement' tag: BEIDOU, COSMOS, GALILEO, IRNSS, NAVSTAR, QZS. The longitude and latitude is stored as a tag acting as a unique key of each DOP value. DOP values are assigned as fields per coordinate. The figure below is a query for NAVSTAR data in a simple table format.

table mean	_measurement group string	_field group string	_value no group double	_time no group dateTime:RFC3339	Latitude group string	Longitude group string
0	NAVSTAR	DOP	1.431198359	2023-05-01T15:30:00.000Z	-102.000000000	-1.500000000
0	NAVSTAR	DOP	1.579864264	2023-05-01T16:30:00.000Z	-102.000000000	-1.500000000
0	NAVSTAR	DOP	1.771507502	2023-05-01T17:30:00.000Z	-102.000000000	-1.500000000
0	NAVSTAR	DOP	1.513965607	2023-05-01T18:32:00.000Z	-102.000000000	-1.500000000

Figure 2: User Inferface of InfluxDB

8. User Interface

8.1 Overview of User Interface

The user will open the web application and see a world map with empty parameters. The user can select different satellites to display the DOP values. The world map then transitions to the corresponding visualization of DOP values. The user can zoom in the world map to see more closely.

8.2 Screen Frameworks or Images

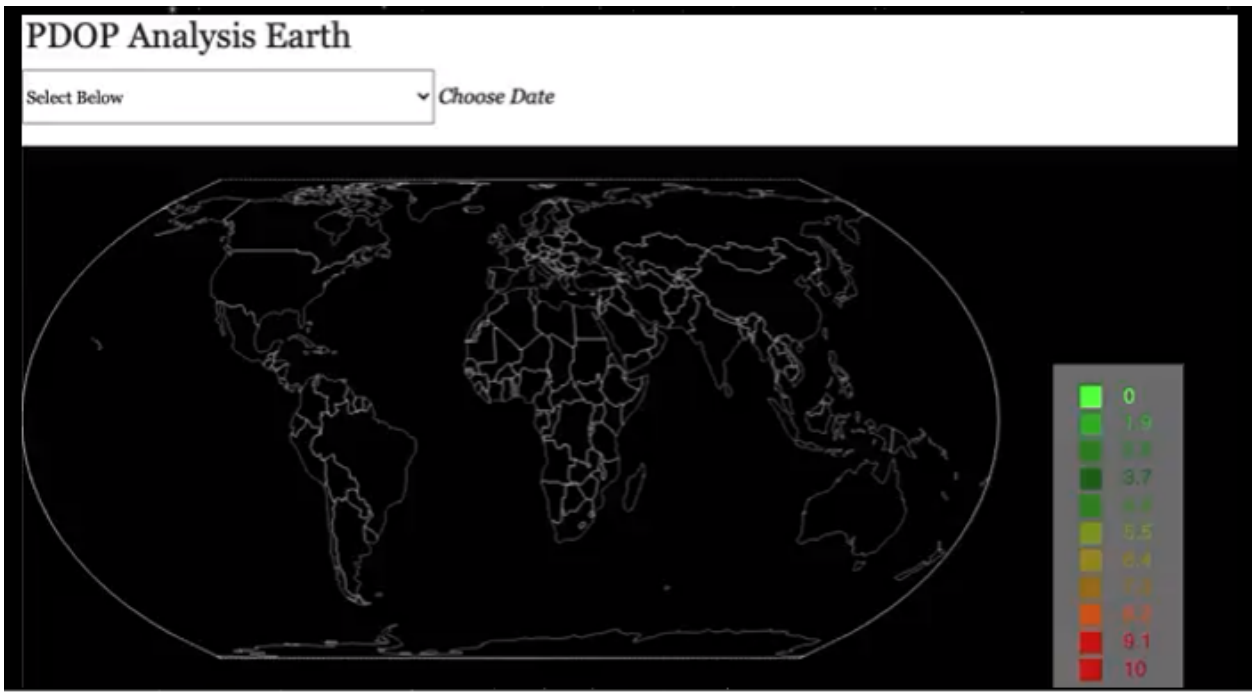

Figure 3: Web Application

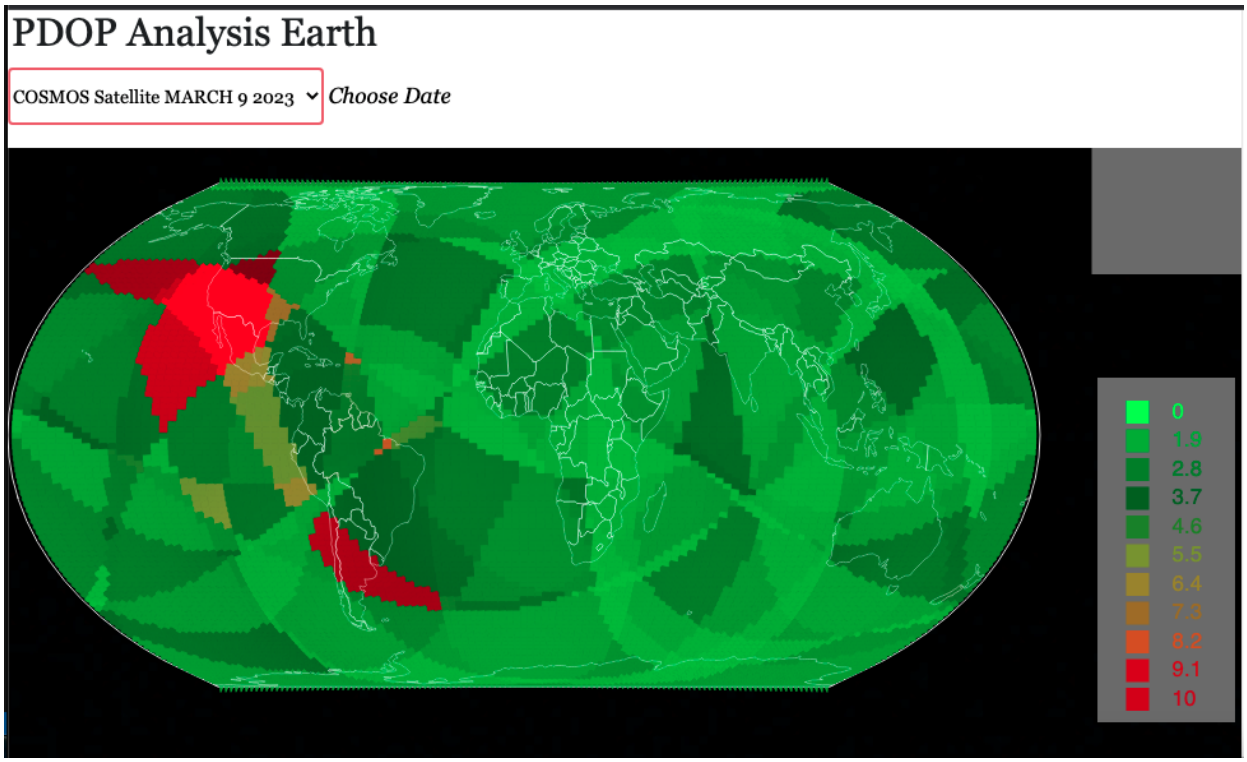

Figure 4: Selecting a specific Satellite

Figure 5: Interacting with Web Application

8.3 User Interface Flow Model

A discussion of screen objects and actions associated with those objects. This should include a flow diagram of the navigation between different pages.

9. Requirements Validation and Verification

SRS Index	Requirements	Satisfied By	Testing Method
4.1.1	Process 1 shall make a call to Space Track API, and output a file containing TLE's	Data Gathering Team	Python scripts in containers request TLE data from Space Track API
4.1.2	Process 2 shall use the output of Process 1 to generate a SOAP scenario and output a DTT file containing SOAP analysis	Data Processing Team	ORB files are generated and run through SOAP producing DTT files.
4.1.3	Process 3 shall use the output of Process 2 and store contents of that file into InfluxDB	Data Storage Team	Analysis output files are stored into InfluxDB
4.1.4	Process 4 shall query from InfluxDB and output visualizations that will make analysis easy to understand	Web Application Team	Data is queried from InfluxDB and visualized on a world map.
4.2.1	Docker provides easy building, testing, and deployment of application	DOPA Team	Containerizing code and applications
4.2.2	Space Track provides TLE information of the Global Positioning System	Data Gathering Team	Python scripts in containers request TLE data from Space Track API
4.2.3	Grafana is a web visualization software which shall provide visualizations to data that has been stored in InfluxDB	Web Application Team	Grafana is replaced by D3.js which is used to visualize PDOP data on world view
4.2.4	InfluxDB Time Series Database which shall be used to store TLE data	Data Storage Team	Storing TLE into InfluxDB no longer required

4.3.1	The source of the information that will be placed into InfluxDB is a dtt file	Data Processing Team	DTT file is generated and handed off to the database
4.3.2	Dtt file is the analysis output from SOAP	Data Storage Team	SOAP generates the analysis output into dtt file
4.3.3	Type of information being stored in InfluxDB include a Geographical location in the form of longitude and latitude, as well as a PDOP value	Data Storage Team	InfluxDB stores the epoch, coordinates, and PDOP values
4.4.1	The application shall accommodate different platforms including Windows and MacOS.	DOPA Team	DOPA accommodates for Windows and macOS platforms
4.4.2	The SOAP image should be preinstalled. The request process to download SOAP shall be involved signing an NDA with Aerospace	DOPA Team	DOPA Team went through the process of obtaining and installing SOAP
4.4.3	The limitation of requesting satellite data per user from space-track.org shall be considered.	Data Gathering Team	Space Track API rules indicate 1 request per hour

10. Glossary

Acronym	Term	Definition
SOAP	Satellite Orbit Analysis Program	An analysis software used heavily within Aerospace to manually build and analyze satellite scenarios. Capable of performing more than 150 different analysis types
DOP	Dilution of Precision	is a geometric measure of GPS accuracy
TLE	Two Line Element	is a data format encoding a list of orbital elements of an Earth-orbiting object for a given point in time
	Docker	A platform to package software to make it capable of running on any software on any machine
	Epoch	A moment in time used as a reference point
API	Application Programming Interface	is a set of definitions and protocols to build and integrate application software
	InfluxDB	A database platform that allows to store and manage large volumes of time series data

11. References

Reference Name:	Source:
1. Software Design Document (SDD) 2. Software Requirement Document (SRD)	CSNS Website https://ascent.cysun.org/project/project/view/197
Space Track API	Spack Track API Documentation https://www.space-track.org/documentation
Influx Database	https://www.influxdata.com/
D3.js	https://d3js.org/