AN OPEN SOURCE WEB-GIS BASED PRECISE SATELLITE TRACKING AND VISUALIZATION TOOL USING TWO LINE ELEMENT DATA

NAME	ENROLLMENT NUMBER	COURSE
ABHISEK MAITI	mgi17-10930	M.Sc
SAYANTAN MAJUMDAR	mgi17-10933	M.Sc
SHASHWAT SHUKLA	mgi17-10934	M.Sc
SAURABH GUPTA	pgd17-10945	PGD

Month, Year: May, 2018
Submitted to: Mr. Shiva Reddy Koti

FACULTY OF GEO-INFORMATION SCIENCE AND EARTH OBSERVATION
UNIVERSITY OF TWENTE, ENSCHEDE, THE NETHERLANDS

INDIAN INSTITUTE OF REMOTE SENSING,
INDIAN SPACE RESEARCH ORGANISATION, DEHRADUN, INDIA

Contents

List of Figures

1. Introduction

1.1. Background

Accurate monitoring of satellites plays a pivotal role in analysing critical mission specific parameters for estimating orbital position uncertainties. An appropriate DBMS at the software end, could prove its potential as a convenient solution to the existing file based two line element (TLE) [1] data structure. The existing web based satellite tracking systems, e.g., n2yo [2], satview [3] and satflare [4]. are unable to provide satellite monitoring based on a particular location. The users need to zoom in to the world map for obtaining information of the satellites that are currently over the respective area. Also satellite searching is a cumbersome task in these web based systems.

A two-line element set (TLE) is a data format which encodes a list of orbital elements of an Earth-orbiting object (in this case satellite) for a particular epoch. Using suitable python libraries (e.g. Skyfield [5] and Orbit-Predictor [6]), the position and velocity at any point of time can be estimated to some accuracy. TLEs are widely used as input for projecting the future orbital tracks of space debris for purposes of characterizing "future debris events to support risk analysis, close approach analysis, collision avoidance manoeuvring" and forensic analysis [7].

1.2. Goals

In this project, a systematic approach has been utilized for developing a generic open-source web-GIS based tool for addressing the aforementioned issues. This tool incorporates SQLite3 database for storing the parsed TLE data procured from CelesTrack (NORAD) [5], which being a portable and light-weight database application, offers added advantages over other resource intensive databases.

1.3. Methodology

Initially, the TLE data are scraped from CelesTrack server as raw ASCII file format. This data format is the de facto standard for distributing the list of orbital parameters of an orbiting satellite for a given epoch. Here, each set of elements are written to two 70-column ASCII lines.

- **Building a prototype desktop tool**: In the desktop tool, the Python based Skyfield library has been used for the high precision (results agree with that of the United States Naval Observatory and their Astronomical Almanac to be within 0.0005 arc seconds) estimation of satellite trajectory, which in turn, takes the TLE data as input. However, the provided arguments have to be parsed beforehand for subsequent query processing. Most of the existing TLE parsers are written in Haskell and JavaScript. Currently, Python being the most popular and extensively used scripting language in the space industries, a pure python based TLE parser has been developed for facilitating the application development process in a convenient and hassle-free way.

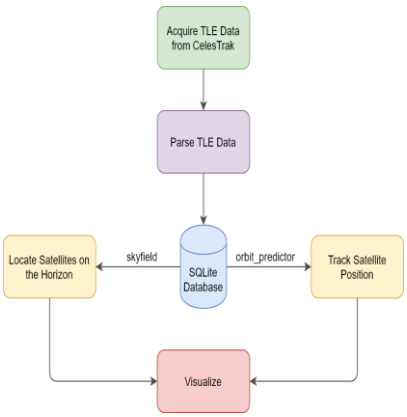


Figure 1.1 Desktop Tool Workflow

- **Linking with Web-GIS**: In this phase of the workflow, the entire desktop tool has been made available for the Web-GIS environment wherein the user is able to search and track different satellites. The SQLite DB is kept at the server side and a suitable UI/UX has been developed.

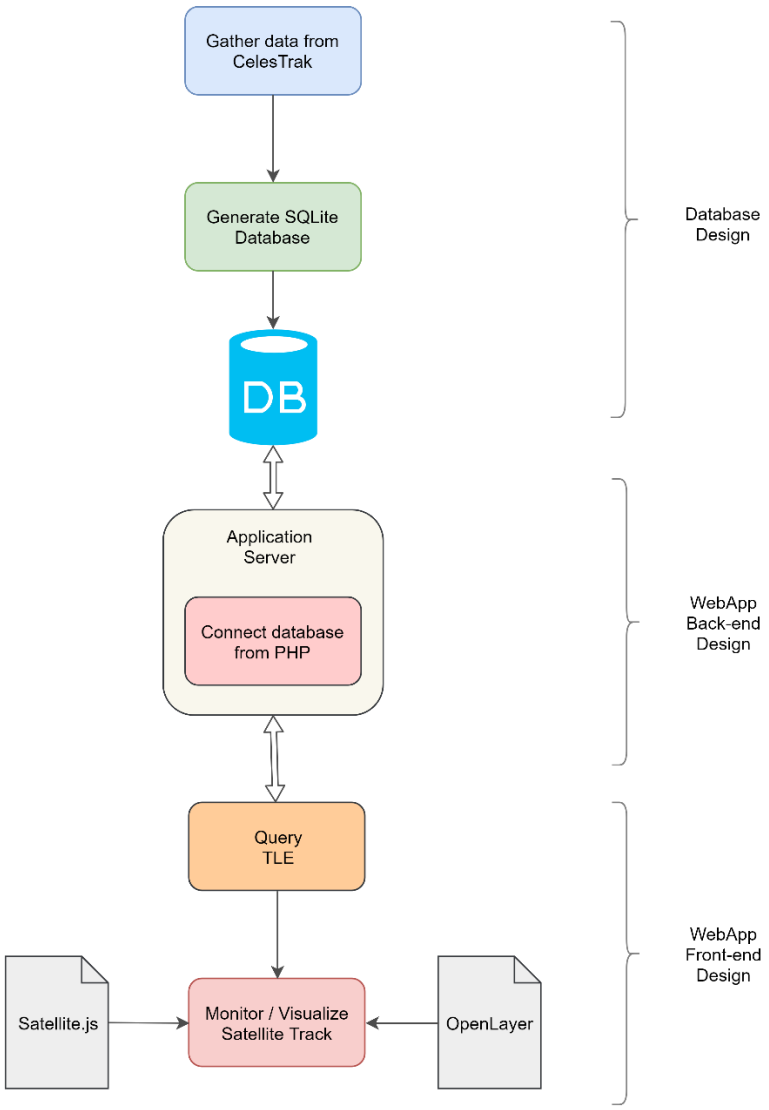


Figure 1.2 Project Workflow

2. Results

In the figures 2.1-2.4, the time interval is taken to be one minute. Iridium33 is a space debris and is shown in Figure 2.4. In the real time scenario illustrated in Figure 2.5, the time interval is taken as 500 milliseconds.

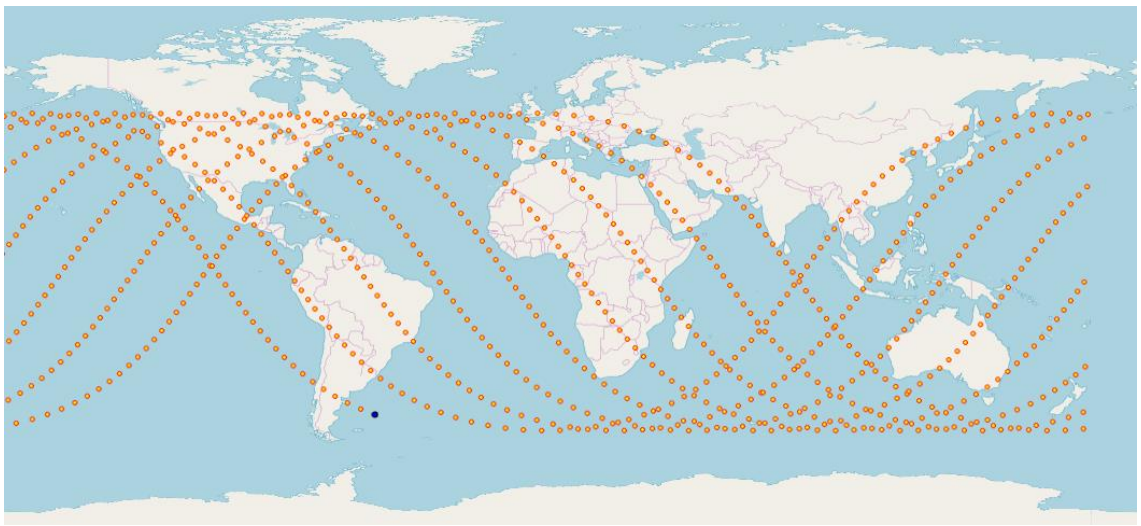


Figure 2.1 ISS Simulated Ground Track

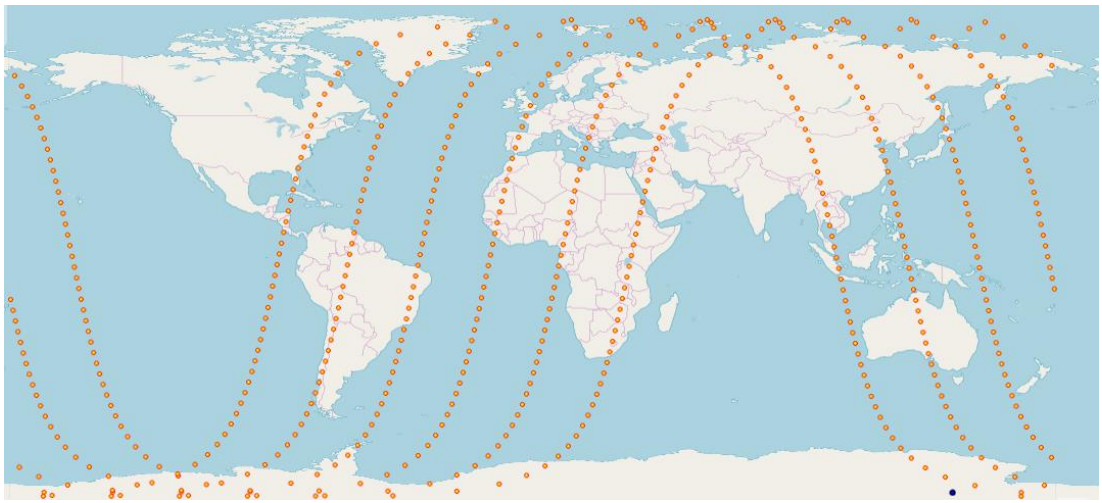


Figure 2.2 WorldView-4 Ground Track

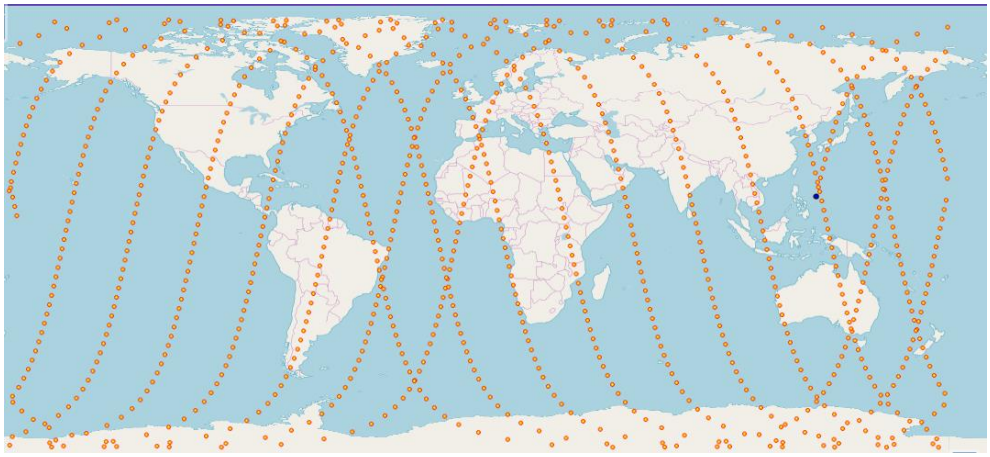


Figure 2.3 ENVISAT Simulated Ground Track

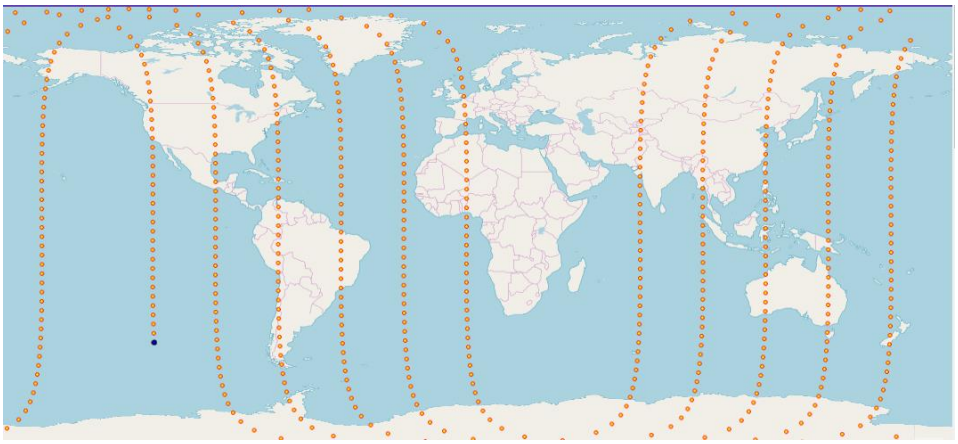
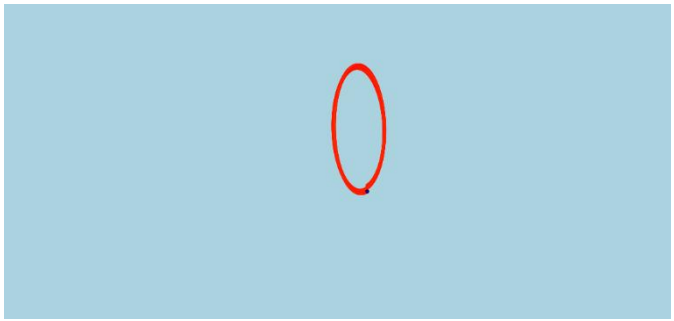
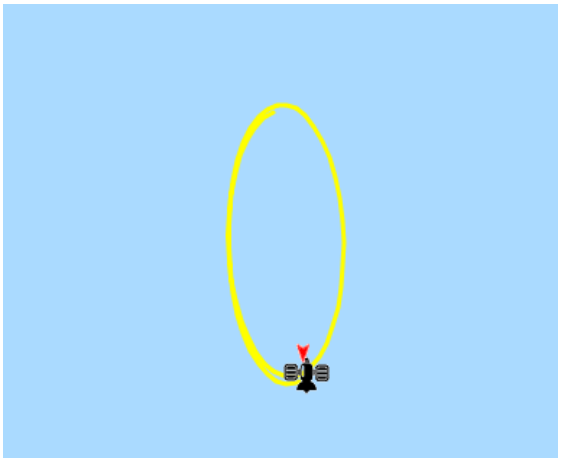


Figure 2.4 IRIDIUM33 Simulated Ground Track

Figure 2.5 ISS Real Time Ground Track



(a)



(b)

Figure 2.6 (a) TrackSat INSAT 4CR (b) N2YO INSAT 4CR

3. Conclusion

Due to time constraints, a prototyping based SDLC (Software Development Life Cycle) approach has been followed in this project. As part of the future software development process, the proposed webapp could be integrated with smartphone environments wherein the user would be able to search and track different satellites. Moreover a user friendly interface is to be built, such that, the respective user would be able to choose the geo-temporal variables. At the back end, these would be used as binding variables to a SQLite query, which would then be fired to display a 2D animation showing the current trajectories of near-by orbiting satellites. In this case, the temporal attribute provided by the user can represent either past or future satellite orbiting scenarios. Apart from this, a provision for adding future satellite orbital information would be kept, which can be further used for trajectory simulation studies. In order to perform software verification and validation (V&V) the generated outputs could be cross verified with a ground based survey which would involve the use of DGPS or mobile GPS for obtaining orbital arrangement of the existing satellites at a given instance of time which would be matched to that the tool. The TLE database would also be updated automatically whenever there is any change of the TLE data on the CelesTrack server.

References

[1] Dr. T.S. Kelso, “CelesTrak.” [Online]. Available: http://celestrak.com/. [Accessed: 11-Apr-2018].
[2] N2YO, “N2YO.” [Online]. Available: http://www.n2yo.com/. [Accessed: 18-Apr-2018].
[3] SatView, “SATVIEW - Tracking satellites and Spacejunk in Real time.” [Online]. Available: http://www.satview.org/. [Accessed: 18-Apr-2018].
[4] SATFLARE, “SATFLARE.” [Online]. Available: http://www.satflare.com/. [Accessed: 18-Apr-2018].
[5] Brandon Rhodes, “Skyfield.” [Online]. Available: http://rhodesmill.org/skyfield/. [Accessed: 18-Apr-2018].
[6] N. Demarchi, J. Rodriguez, I. Malerba, “orbit-predictor” [Online]. Available: https://github.com/satellogic/orbit-predictor/. [Accessed: 19-Apr-2018]
[7] T. Carrico, J. Carrico, L. Policastri, and M. Loucks, “Investigating orbital debris events using numerical methods with full force model orbit propagation,” *Adv. Astronaut. Sci.*, vol. 130 PART 1, no. September 2017, pp. 407–426, 2008.