

A Novel Web based Multi-Mission Satellite Locator & Coverage Viewer

Shaik Naseepjan, Amit Kumar Singh, Sudhakar Karla, Manikumar Vedantam, M. Manju Sarma
National Remote Sensing Centre
Hyderabad
naseepjan_s@nrsc.gov.in, amitkumar_singh@nrsc.gov.in,
sudhakar_karla@nrsc.gov.in, manikumar_v@nrsc.gov.in, manjusarma_s@nrsc.gov.in

Abstract:

National Remote Sensing Centre (NRSC) is one of the primary centers of Indian Space Research Organization (ISRO), Department of Space (DOS). NRSC has the mandate for establishment of ground stations for receiving satellite data, generation of data products, dissemination to the users, and development of techniques for remote sensing applications including disaster management support, geospatial services for good governance and capacity building for professionals, faculty and students. In Order to suffice the so mentioned requirements Satellite Data Acquisition, it's processing and dissemination of the acquired data plays an important and pivotal role. To meet such a large number of data requirements many IRS (Indian Remote Sensing) satellites are in operation. Together with IRS missions' even large number of non-IRS data is also acquired at NRSC. In order to visualize such a large number of missions there should exist some application utilities to track satellite's current location, its predicted path as well as ground stations location and visibility circle. Satellite Locator and Coverage Viewer application provides these requirements of satellite's current location, ground station visibility with features for showing of planned vs. acquired data on map based interface.

Keywords — Satellite, Ground Station, SGP4, SDP, TLE, LEO, Ephemeris.

I. INTRODUCTION

Satellite Data acquisition, processing and dissemination of EO (Earth Observation) satellite data to the user is of paramount importance for meeting the requirements of the user community for the management of natural resources, disaster monitoring and capacity building. Large number of IRS (Indian Remote Sensing) missions as well as Non-IRS mission's data is being acquired, processed and disseminated from NRSC. In order

to keep precise and accurate management of in operation satellites some sort of application mechanism is required to be in place. Satellite Locator and Coverage Viewer is a web based application that caters to the needs of monitoring the satellite's current location with enhanced features and functionalities to meet the same for accurate tracking and monitoring of In-Orbit IRS and non-IRS satellites.

II. OBJECTIVE

The key technical requirement of the application is to visualize satellite-based traffic system with high accuracy to have the ability to provide three-dimensional position, of satellites using satellite visibility to a ground station, on a selected date and time.

There are several IRS satellites orbiting the earth, observing it continuously. All the satellites are not visible to the given ground station all the time. It is a prime requirement to identify the set of satellites which cover the user's AOI from a specific ground station on a selected date. This web based application Visualizes the current location of the In-orbit ISRO satellites and others with trace providing Global view with their position on map based frontend.

III. TECHNICAL OVERVIEW

The basis of satellite orbital propagation is Keplerian orbit elements [1]. These orbit characteristics define the orbital orientation, shape, speed of orbit and the last known position of the satellite. These can be used to generate a rough estimation of a satellite's position; however, such predictions will fail to reflect reality for extended periods due to disturbances, known as perturbations, in the orbit. These disturbances include atmospheric drag and lunar gravitational pull to name a few. The need for accurate orbital prediction at the dawn of the space age leads to the development of the Simplified General perturbation (SGP) model. SGP models are a set of five mathematical models[5] (SGP, SGP4, SDP4, SGP8 and SDP8) used to calculate orbital state vectors of satellites and space debris relative to the Earth-Centred inertial coordinate system. This set of models is often referred to collectively as SGP4 due to the frequency of use of that model particularly with Two-Line element [2] sets produced by North American Aerospace Defense Command (NORAD) and National Aeronautics and Space Administration.(NASA).These models

predict the effect of perturbations caused by the Earth's shape, drag, radiation, and gravitation effects from other bodies such as the sun and moon. Simplified General Perturbations (SGP) models apply to near earth objects with an orbital period of less than 225 minutes. Simplified Deep Space Perturbations (SDP) models apply to objects with an orbital period greater than 225 minutes, which corresponds to an altitude of 5,877.5 km, assuming a circular orbit.

A. TLE data format

The basic form of SGP4 propagation uses an input file known as a Two Line Elements (TLE). These two lines contain the Keplerian elements information pertaining to the satellite along with identification and time. This format was specified by NORAD and continues to be used today. The details pertaining to the exact format [2] are presented in figure 1. Other data entered into the program include the ground station coordinates, World Geodetic System reference.

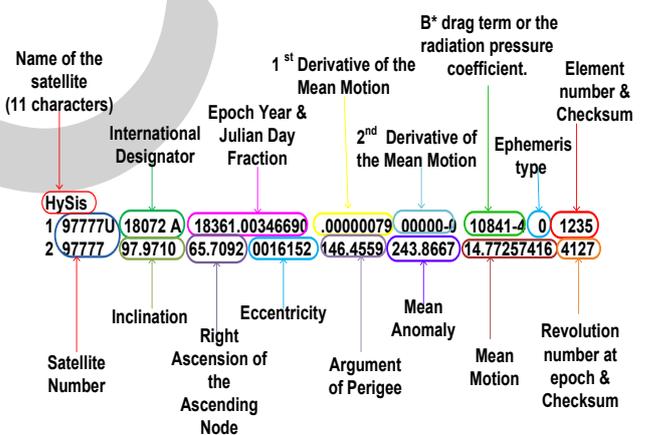


Figure 1: Sample TLE for HySIS satellite

B. System Architecture and Design

Satellite Locator & Coverage Viewer is based on event driven component based layered architecture. The component based architecture focuses on decomposition of design into logical components

that bring forth well-defined interfaces containing methods, events and properties. The layered architecture focuses on grouping related functionality within the application into distinct layers stacked vertically and strong separation of concerns that enables flexibility and maintainability. The availability of TLE's satellite's location tracking module gets invoked as well as based on user selected input event actions like planned, acquired data display on map interface do takes place. The architecture is a client-server architecture[4] pattern in which the user interface (presentation), functional process logic ("business rules"), computer data storage and data access are developed and maintained as independent modules, most often on separate platforms. The different layers of the application include

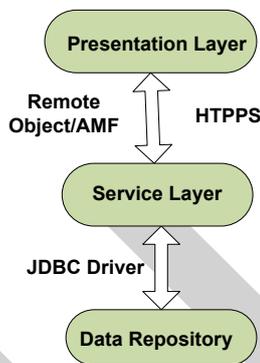


Figure 2: System Architecture

C. Presentation Layer

Presentation layer is the layer through which the user communicates with the application. It is this layer only that provides monitoring and visualization services to the user. This layer facilitates user inputs and presents the data for interactive services through a rich graphical user interface. Flex is used to handle the presentation layer. Flex applications have a client tier which enables the client to offload computation from server minimizing network latency and bringing higher response for interactive user interfaces. Flex brings tightly coupled integration for presentation

to client side systems making the response dependability on the client system. As flex application is state full it can make changes to the view without contacting the server, even though they take more time and resources to develop and test compared to traditional web application they provide better response for interactive intense applications. Map visualization features were built using modest map API which is an open source mapping framework based on Action Script and Flex. The world image and administrative boundaries were displayed as layers with panning and zooming features. Features are also built to show planned vs. acquired path on the map based frontend display.

D. Service Layer

The service layer includes business logic, event processor as well as map server for display on map. The business logic is responsible for processing of the fetched data and presenting it to the presentation layer in user understandable form. The Map Server is responsible for rendering map visualization to the presentation layer. Event processor is responsible for capturing of events and triggering further actions like satellite's location display based on TLE's availability.

E. Repository

The Repository is responsible for storing of various information related used for information display like the planned paths, acquired browse information as well as satellite pass related information in the database which is based on user input is fetched, processed and sent to the presentation layer for display.

IV. IMPLEMENTATION

A. Presentation Layer

Presentation layer provides features to facilitate user's interaction with the services provided by application. Basic needs of a presentation layer should be to provide features like user friendliness, appealing GUI, smooth navigation and accurate validations. We have used Adobe Flex as frontend

design technology. Adobe Flex is based on RIA (Rich Internet Application) technology which provides rich media, resides on client, provides asynchronous communication and data persistence and reduces network consumption. It comprises of MXML, a markup language based on Extensible Markup Language (XML) for application Layout creation and Action Script for user interaction, complex data functionality, and any custom functionality not included in the Flex class library. Communication with the web services is done using the remoting services through AMF interface using BlazeDS at server end which provides the binary encoding of data leading to lower data movement reducing network consumption. This application has used the following features in its frontend design-

- ❖ Separation of modules providing different functionalities.
- ❖ Design of custom components for reusability.
- ❖ Use of configuration files for easy maintenance, configurability and flexibility.
- ❖ Features for easy plug-in capability for adding new components.
- ❖ Features to easily expand the application for newer missions.
- ❖ User friendly and appealing GUI (Graphical User Interface) for the user.
- ❖ Minimum coupling and maximum cohesion to maintain the testability of the application.
- ❖ Use of modest map open source mapping API for map based frontend services which is OGC compliant

The frontend design of the application is done using Adobe Flex that contains mxml and action script. For the purpose of static layout design mxml is used and for logic as well as logic implementation action script is used. Efforts have been made to have minimum user intervention and even if it is provided it is done through combo boxes and checkboxes. Complete application is developed as a single page application to easy use and to resolve more navigation. The Communication between presentation layer and the business logic using

through the remoting services provided by Adobe blazeds.

Modest map is used for map display on frontend which is an open source mapping API used for developing rich map based applications and it's rendering on frontend. It provides features like drawing, zooming, overlaying functionalities on map.

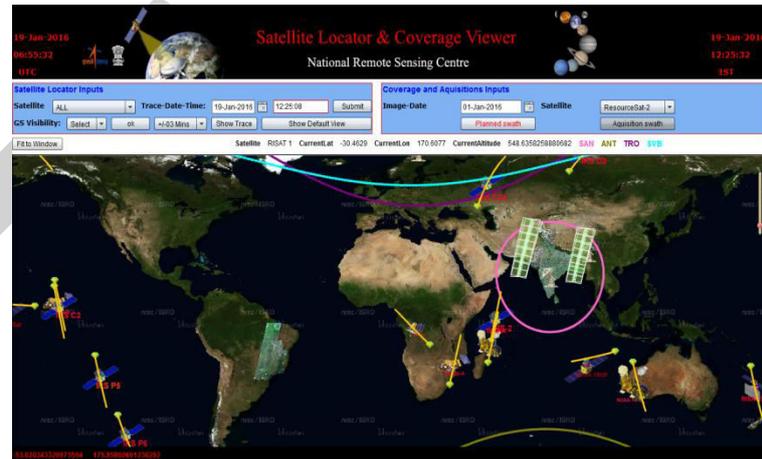


Figure 3: Satellite locator & Coverage Viewer Application

B. Service Layer

Adobe BlazeDS is used in our application which is an open source server based Java remoting and web messaging technology that helps in effective integration of flex and Java. BlazeDS provides remote procedure calls and messages exchanged between different platforms. Our application uses Remote Objects to communicate with BlazeDS-enabled server. Remote Object s returns a serialized data from POJO (Plain Old Java Object) back to the applications and data Binders present those data. Business Logic contains various functions like fetching of TLE's, its processing, application of satellite location prediction algorithm over it, its formatting and its dissemination to the frontend for showing satellites current position. It also uses features to take fetch planned, acquired details based on user inputs from the database, process it based on selected satellite's swath and other parameter, formatting it and finally sending it to the presentation layer in user understandable form. This

layer also provides features to trigger location calculation module and its dissemination to the frontend as soon as TLE's (Two Line Element) for the current day are made available in the operational area. Automatically system will be dispatching the events with input information of satellites, date and time to trace the location of the satellite on the map. There are two generic software packages has been developed.

- Two-line element service
- Orbit determination

Two-line element service, It will read the Two-line element set form NORAD website or ISRO and stored in data repository for generation ephemeris for real-time satellite tracking for all satellites acquired by NRSC.

Orbit determination is a Satellite tracking program written in Java. It allows you to predict the position of any satellite in real time or in the past or future based on events received from presentation layer. It uses advanced SGP4/SDP4 algorithms [3] developed by NASA/NORAD or customizable high precision solvers to propagate satellite orbits. The program also allows for easy updating of current satellite tracking data via CelesTrak.com/ISRO. This application was written in Java, it should run on almost any operating system. The ephemeris package returns position and velocity into latitude and longitude format for presentation layer to plot the data on map display.

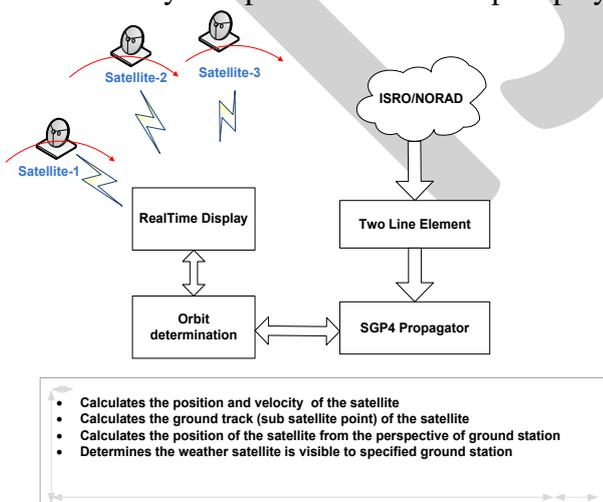


Figure 4: Overview of Satellite Trace System

C. Repository

Normalization is done up to 3rd level to remove redundancy and easy maintenance of database. Optimization of queries is also done for better performance as faster fetching of the values from database. Join Queries are also used to fetch data across the tables. Data Repository contains diversified data comprising of satellite ephemeris information planned and acquired AOI information details. As far as information related to storage of information is concerned Oracle 12g is used as the database

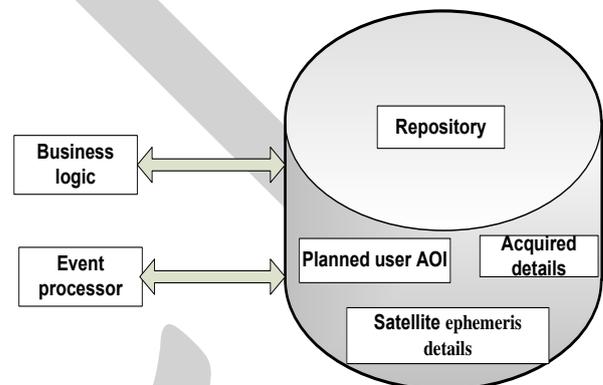


Figure 5: Repository

V. TESTING & RESULT

This application is tested for the conformance of functional and performance requirements including security aspects. This software has pass through software quality assurance (QA) test to validate both the functional and non-functional requirements. The application is tested to ensure the security of data and resources from unauthorized access. Compatibility of the application with several browsers, operating systems and screen resolutions is manually verified. The user interface is thoroughly evaluated to uncover the errors pertaining to navigation, aesthetics, syntax and semantics.

At present this application is enabled for real time tracking of 28 Indian and foreign satellites. The

average response time for the application tested using Rational Performance Tester .The following figure depict the average response time of application with respect to concurrent number of users.

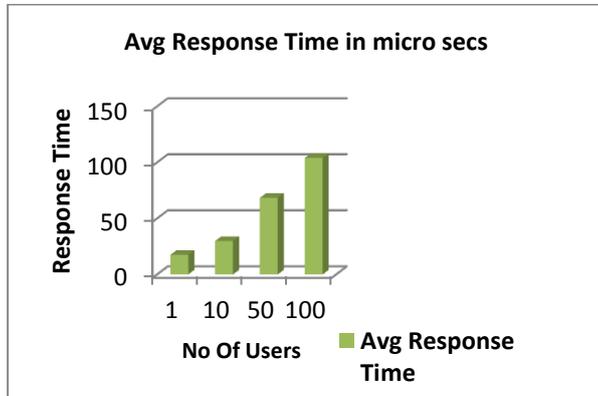


Figure 6: Load testing result of application

Application is flexible enough to be extended for future mission with minimum configuration changes.

The major salient features of the application are.

- ❖ Provides the current location of the In-orbit IRS ISRO satellites and others satellites data acquired at NRSC.
- ❖ Features to predict the future position of Satellite at any given time entered by user.
- ❖ Features to provide NRSC acquisition Ground station location and foot print visualization on map.
- ❖ Features to select, deselect, freeze the satellite that user wants to display.
- ❖ Plot on the map of planned area versus acquired for selected satellite and date.
- ❖ Provides brief description about satellite based on user action event on satellite image.

VI. CONCLUSIONS & FUTURE WORK

The whole application is realized keeping in mind user friendliness, seamless navigation and faster and efficient response to the user. The future plan is to extend this application on mobile devices and geo-tagging of places covered by satellite pass, systematic coverage display is also planned in subsequent releases.

ACKNOWLEDGMENT

The authors would like to thank Director, NRSC for giving us the opportunity to realize the application. The authors are grateful to Deputy Director, DPPA&WAA and NRSC data centre for providing the inputs. The authors would also thank quality assurance group for materializing the workflow.

REFERENCES

- [1] Hoots, Felix R.; Ronald L. Roehrich (31 December 1988). "Models for Propagation of NORAD Element Sets" (PDF). United States Department of Defense Spacetrack Report (3). Retrieved 16 June 2010.
- [2] Kelso, T. S., "Frequently Asked Questions: Two-Line Element Set Format," *Satellite Times*, vol. 4, no. 1, pp. 68-69, Sept 1997.
- [3] Vallado, David A.; Paul Crawford; Richard Hujsak; T. S. Kelso (August 2006). "Revisiting Spacetrack Report #3" (PDF). *Astrodynamics Specialist Conference*. Retrieved 29 April 2017
- [4] Eckerson, Wayne W. "Three Tier Client/Server Architecture: Achieving Scalability, Performance, and Efficiency in Client Server Applications." *Open Information Systems*.
- [5] https://en.wikipedia.org/wiki/Simplified_perturbations_models