

SpaceCentre: Advanced Web Application for Satellite Mission Control

Wei-Ting Loke, Rakkappan Lakshmanan, Benedict Wen-Yong Chong, Hassan Ali Askari
 Satellite and Airborne Radar Systems Laboratory, ECE Department, National University of Singapore
 Block E4A, #04-03, 3 Engineering Drive 3, Singapore 117582; +65 6516-5154
 elelwt@nus.edu.sg

Cher-Hiang Goh
 DSO National Laboratories, Singapore
 20 Science Park Dr, Singapore 118230; +65 6776-2255
 gcherhia@dso.org.sg

ABSTRACT

In this paper, we present *SpaceCentre*, an advanced web application for satellite mission planning and control. Developed at the Satellite and Airborne Radar Systems Laboratory by undergraduate students and research staff, *SpaceCentre* embraces the latest web technologies and application design philosophies such as mobile readiness, responsiveness, and reactivity. By leveraging NodeJS and the Node Package Manager, developers of *SpaceCentre* have ready access to a huge repository of the latest and most advanced libraries available today, rendering the tool easily extensible. *SpaceCentre* is a feature-packed application equipping mission control staff with the tools needed to plan and execute satellite missions seamlessly. The web application has been integrated with the ground station equipment of the Galassia nanosatellite, and is currently under active use by research staff of the laboratory, as well as for educational purposes for undergraduate students of the institute.

INTRODUCTION

Global spending on the space industry, driven largely by commercial space companies and services, reached an estimated \$323 billion in 2015. Accordingly, satellite missions are becoming increasingly complex. For over a decade, research institutions have largely relied on open-source ground station software, such as Orbitron [1] and GPredict [2], for satellite mission planning and operations. These desktop applications were developed using legacy languages and technology, not actively maintained, and often insufficient to meet the needs of today. Lack of plug-in features also render these tools difficult to extend in functionality. For example, when planning an Earth observation mission, a mission location must first be determined. Depending on the mission, this can be an interesting feature such as land and water masses, or manmade structures, as often is the case in reconnaissance missions. The map features built into existing open-source desktop applications do not present this level of detail. In addition, other important information such as cloud cover or space weather is not available. Planning for an imaging pass can thus be very time-consuming, requiring the use of multiple offline and online tools such as Orbitron/GPredict, Google Maps, cloud cover prediction websites, and space weather information websites.

SpaceCentre is a web-based ground station application developed by students and staff of the Satellite and Airborne Radar Systems Laboratory (SARSL) at the National University of Singapore (NUS) that enables

mission control staff to perform mission planning and satellite operations effectively. The web application is developed using MeteorJS, a reactive full stack framework built upon NodeJS and MongoDB that enables the rapid development of safe and secure web applications. Web applications have the advantage of remote accessibility via standard web browsers and login protocols. A plethora of open-source libraries such as Google Maps, ChartJS, Plotly, Bootstrap etc. allow functionality to be easily extended. *SpaceCentre* provides five key features for mission planning and operations: real-time satellite tracking, pass prediction, weather information, user settings and mission control. These features allow users to plan and operate any satellite with ease. For example, to perform imaging operations, a mission location must first be determined – one usually containing interesting terrestrial features. Pass prediction provides information on the satellite's future passes, filtered according to user preferences (e.g. max elevation, visibility, cloud cover) and displayed in a responsive table whereby each row may be selected to view the orbital path. This allows easy identification of suitable passes with high elevation, good sunlight and minimum cloud cover. Real-time tracking allows users to monitor the satellite's current position before proceeding to communicate with the satellite via the mission control interface, allowing users to communicate with the satellite conveniently. Power users may have the option to input low-level administrative commands. Upon successful imaging, users can verify the image taken against the embedded Google Maps widget to ensure accuracy. Global teams

can easily collaborate on satellite operations via the standard web interface, without needing remote desktop tools. For instance, round-the-clock operations on satellites in near-equatorial orbit can be achieved with one team based in Singapore and another in Brazil. This is of particular importance to commercial satellites looking to maximise every pass. Using mobile deployment in MeteorJS, SpaceCentre is also available on Android and iOS, giving operators convenient mobile access anytime, anywhere. SpaceCentre is a powerful, full-featured satellite mission planning and control tool, and will be deployed for existing as well as future satellites of the institute.

WEB VS DESKTOP

Web applications today have a distinctive edge over traditional desktop applications, as the former may be accessed remotely from different platforms, even mobile, without the hassle of platform-dependent installation or updates. Similarly, web applications tend to be highly scalable due to the availability of open source libraries and source codes that can be integrated with relative ease while requiring no compilation. Unlike traditional application programming languages such as C++, Java and C# that often require a dedicated IDE, run-time environment and proprietary libraries for development and deployment, web applications today can almost always be developed using free open-source tools, keeping development costs low. Web applications are therefore cost effective, following the “write once, run everywhere” concept. Web applications may be deployed either on dedicated servers or on the cloud, and typically support simultaneous access by multiple users. Most importantly, due to the rapid advancement in and proliferation of web technologies over the past decade, the typical user of a desktop terminal or mobile device today would be knowledgeable in the operation of web browsers, with set expectations of how a

reasonably well-designed web applications should behave.

CLIENT-SERVER STACK

A good backend stack is critical in order to implement advanced business logic and to interface with other middleware and database programs, such as MongoDB. Node.js was chosen as the server runtime environment, not because heavy real-time access is expected (as not more than a few people are expected to be operating a satellite at any one time) but because Node.js and the accompanying Node Package Manager (npm) lends itself to a huge repository of the latest and most advanced libraries available to web developers today. To this end, MeteorJS [3] was picked as the full stack framework of choice. MeteorJS comes pre-integrated with MongoDB and uses the Distributed Data Protocol and publish-subscribe pattern to automatically propagate data changes between the frontend and backend without requiring the developers to write the synchronization code themselves. As a NoSQL database, MongoDB is able to handle unstructured data natively, allowing custom data to be added easily.

On the client side, native support for reactive templates using Blaze and React.js enables the frontend to update automatically each time data is changed on the backend, providing the end-user with a seamless interactive experience. MeteorJS enjoys a huge user base, with plenty of well-documented libraries and large community support. In contrast, the MEAN stack [4] requires that the developer write the code and define the REST endpoints to connect all components of the stack together, while still lacking many of the data synchronization, server-side autocomplete and reactive features. Finally, MeteorJS also supports mobile deployment to the iOS and Android platforms.

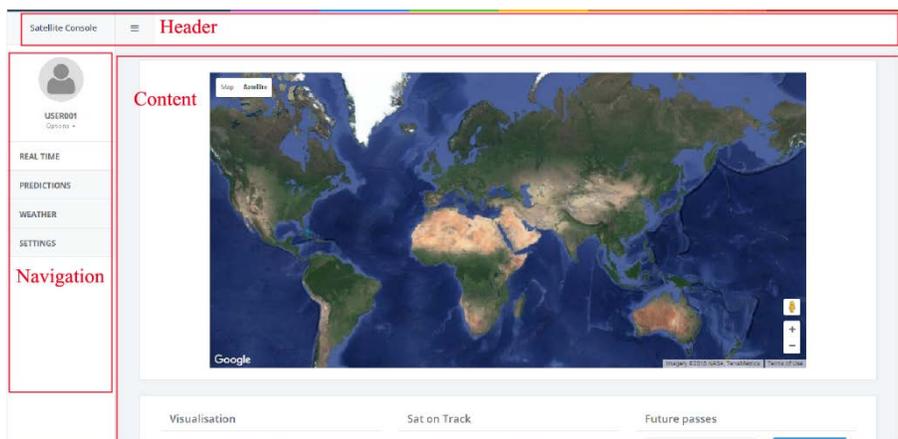


Figure 1: UI Layout of SpaceCentre

SPACECENTRE FEATURES

The user interface of SpaceCentre is structured as an administrative dashboard comprising the header, footer, navigation side panel and content sections (Fig. 1).

The header displays the web application's main labels, while the footer shows miscellaneous information such as contact and license information. Within the collapsible navigation side panel are clearly labelled function tabs. The content is a reactive template that changes as information is being updated in real-time. The content panel is further divided into clearly labelled subsections for intuitive operation. Content panels are responsive and reposition themselves according to the size of the viewport.

Mission Planning

One key feature in SpaceCentre is the integrated Google Maps widget. Unlike the static map in GPredict, users can zoom in to particular locations, revealing detailed information about the terrain. Also embedded are UI elements such as markers with custom icons and coloured lines differentiating flight passes. Images captured during a pass can be verified against the terrain view in Google Maps. Night shadow is implemented using the Rrossengeorgiev Nite-Overlay library [5], which shades the areas currently in eclipse. The library is adapted from NOAA's solar calculator, based on equations from Astronomical Algorithms [6]. A night shadow terminator is added to serve as a further visual aid.



Figure 2: Real-time Google Maps View with Satellite, Footprint, Primary Orbital Path, Ground Station, Sun Marker and Night Shadow Position Overlay

The satellite radar is created using Plotly, a MeteorJS charting library. The radar shows where a satellite is or will be at each point in time, and displays its flight path with respect to its azimuth, elevation and observed location in both prediction and real-time views. Azimuth starts at the North line representing 0° in a clockwise direction, while elevation starts with a maximum of 90° at the centre and ends with a minimum of 0° at the radial edge. The satellite radar is especially useful for determining pass direction and estimating the communication time left.

SpaceCentre has four key features for mission planning: *real-time satellite tracking*, *pass prediction*, *weather prediction* and *user settings*. For real-time satellite tracking, the user can monitor a particular satellite's current location using the embedded Google Maps widget and satellite radar described above (Fig. 2). Orbital parameters such as elevation, azimuth, latitude, longitude and Doppler Shift are also calculated using SGP4 [7] (PredictLib by Andrew T. West [8]) and reported in real time (Fig. 3). Additional options are available for the user to toggle between satellites and display preferences.

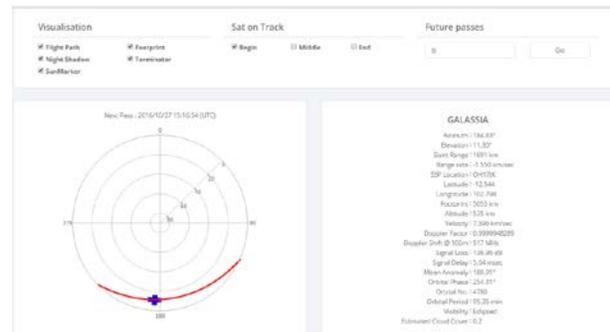


Figure 3: Real-time Satellite Information with Satellite Radar

For pass prediction, users can obtain a satellite's future passes from an observed location. On top of the orbital parameters provided by GPredict and Orbitron, SpaceCentre also provides information on cloud cover and visibility (sunlight), with the results presented in a responsive table filtered according to user preferences such as max elevation, visibility, or cloud cover (Fig. 4 and 5). Each result row is highlighted on mouseover, and may be selected to view its orbital path, as well as future passes, on the Google Map widget (Fig. 6). Separate icons are used to identify the sun (sun marker), the ground station and target mission locations (Fig. 2), allowing the user to distinguish between them easily.

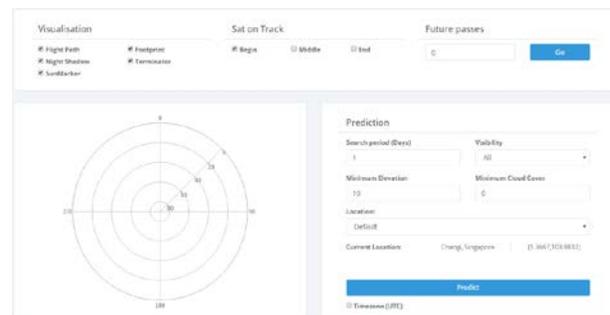


Figure 4: Pass Prediction Options and Filters

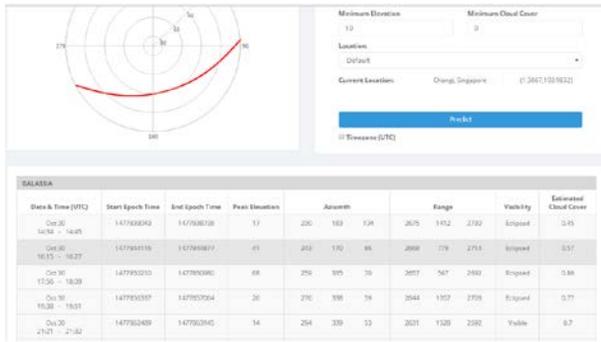


Figure 5: Selecting a Predicted Pass

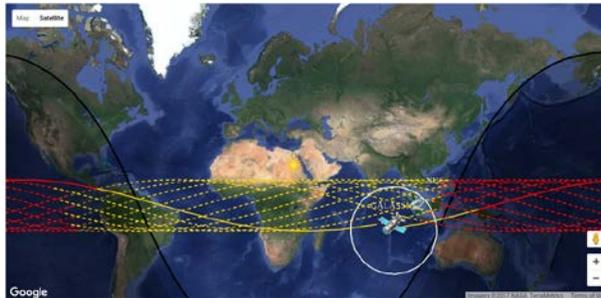


Figure 6: Plotting Future Passes

Weather prediction, a feature not found in open-source mission planning tools, reports to the user the latest solar, geomagnetic and cloud cover forecasts (Fig. 7). Cloud cover prediction is especially useful for optical imaging missions as users can determine whether it is possible to have a clear view of the terrain during the pass. Solar flare prediction allows users to preempt the satellite from harmful space weather. With this feature, satellite operators no longer need to obtain these information from multiple sources.



Figure 7: Solar and Geomagnetic Forecast

Finally, a user setting interface is available for saving custom satellite groups, ground stations and mission locations (Fig. 8). These are reflected throughout the application and associated with the user's account. Users can operate the application easily from any location, which was not possible before without a separate collaboration tool such as TeamViewer or

VNC. For enhanced security, access to the application may be limited to within the institute's private network, as is currently the case for SpaceCentre in NUS.

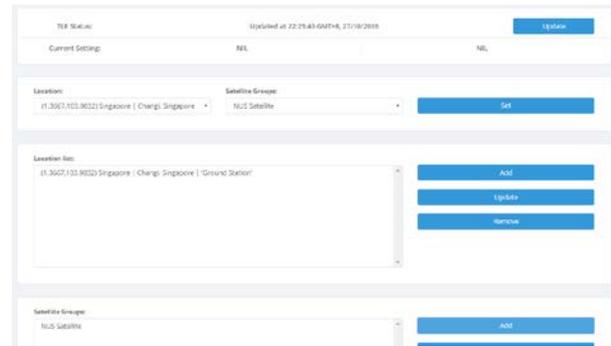


Figure 8: User Settings Interface

Mission Control

In addition to satellite mission planning, SpaceCentre also implements features for actual satellite operations. At SARSL, SpaceCentre has been integrated with the Terminal Node Controller (TNC) and antenna rotator controller for communication with the Galassia nanosatellite. (Galassia is a 2U CubeSat designed by undergraduate students and staff of NUS, launched on 16 December 2015. The reader is referred to [9] for further details.)

Galassia's ground station setup comprises the following equipment (note that this is not the full BOM list):

1. 1 x PC for main ground station
2. 1 x PC for camera live feed and TEC payload mission and analysis
3. 2 x large LCDs connected to the two PCs
4. 1 x iCOM IC-9100 HF/VHF/UHF Transceiver
5. 1 x GSV3000 DC Power Supply for iCOM
6. 1 x GOMspace TNC
7. 1 x Yaesu G-5500 Elevation-Azimuth Dual Controller
8. 1 x Yaesu GS-232B Computer Controller
9. 2 x DC Power Supplies for UHF and VHF LNAs
10. 1 x National Instruments USRP-2920
11. 1 x DC Power Supply for USRP

SpaceCentre interfaces the Ham Radio Control Libraries (Hamlib) [10] on the MeteorJS backend to

Further to the command interface, a data visualisation tool was developed for the Galassia mission. The operator may download housekeeping and telemetry data, such as EPS logs, from the satellite for visualization and storage in SpaceCentre. Similar to the satellite radar, visualization is accomplished using the line charting library in Plotly. Because Galassia stores on-board data in SQLite format, a conversion to MongoDB's native JSON document format was required after downloading into SpaceCentre. Fig. 13 below shows a sample plot of the EPS temperature log for the Galassia nanosatellite.

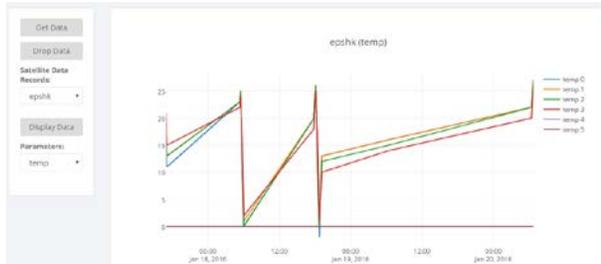


Figure 13: Subsystem Options

The 'Get Data' button performs the parsing of data from one of the 9 different telemetries available: eps_hk, error_log, flash_var, global_var, hk_data, nanocom_data, nanocom_hk, nanocom_rssi, nanohub_hk. The user may switch between the different categories and access the parameters that they wish to analyse. The Plotly line chart offers a multitude of functions such as panning, zooming in and out, or viewing within a range, giving the user the freedom to navigate through the data effortlessly. The 'Drop Data' button clears the satellite's database within SpaceCentre, should this be desired.

CONCLUSION AND FUTURE WORK

Integration of SpaceCentre with the ground station equipment at the Satellite and Airborne Radar Systems Laboratory, National University of Singapore has completed, and the web application is now actively being used for mission planning and operation of the Galassia nanosatellite. The application is also being used for educational purposes with our undergraduate and exchange students.

Given SpaceCentre's ease of extensibility, we envision the application to be the choice mission planning and control framework for future satellites of the institute. Work is currently underway to support more complex operational features, such as the concurrent remote control of multiple ground station antennae, which has foreseeable application in formation flying and constellation missions, as well as high speed downlink protocols (e.g. S/L/X-band).

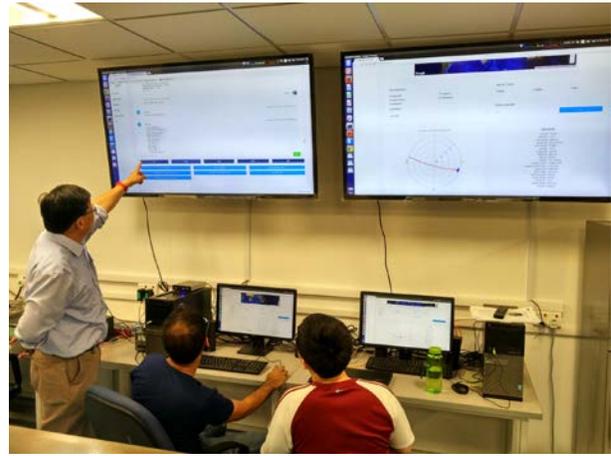


Figure 13: Operating Galassia using SpaceCentre

References

1. Orbitron - Satellite Tracking System. Available online at: <http://www.stoff.pl/> (accessed Mar 2016).
2. GPredict. Available online at: <http://gpredict.oz9aec.net/> (accessed Mar 2016).
3. MeteorJS. Available online at: <https://www.meteor.com/> (accessed Aug 2016).
4. The MEAN Stack. Available online at: <https://www.codeschool.com/mean> (accessed Jan 2017).
5. Rossengeorgiev Nighttime overlay with all twilight zones for Google Maps API 3. Available online at: <https://github.com/rossengeorgiev/nite-overlay> (accessed October 2016)
6. Earth System Research Laboratory (ESRL). Available online at: <http://www.esrl.noaa.gov/gmd/grad/solcalc/calcdetails.html> (accessed October 2016)
7. D. Vallado & P. Crawford. SGP4 orbit determination. AIAA/AAS Astrodynamic Specialist Conference and Exhibit, 6770 (2008).
8. PredictLib. Available online at: <https://andrewwest.com/orbtrak/> (accessed September 2016).
9. E. W. H. Ee, A. N. Nikicio, D. Feng, H. Kumar, H. A. Askari, S. Luo, R. Zhang, C. H. Goh, and H. C. Liaw, "Design, AIT, Launch & Early-Operations of Galassia Nano-Satellite", Small Satellite Conference, Logan, Utah, USA, Aug 6-11, 2016.
10. HamLib (2017), Retrieved 21 March 2017, from HamLib: <https://sourceforge.net/p/hamlib/wiki/Hamlib/>
11. libCSP. Available online at: <https://github.com/libcsp/libcsp> (accessed 21 March 2017)