SpaceOps Conferences
28 May - 1 June 2018, Marseille, France
2018 SpaceOps Conference

10.2514/6.2018-2379

Modular Orekit-based Flight Dynamics Software

Maxime Journot,[1] and Nicolas Frouvelle[2]
CS Systèmes d'Information, Toulouse, 31000, France

Cosmin I. Udroiu[3], Lucian F. Barbulescu[4], Oana M. Hogoiu[5], Maria Teodorescu[6]
CS România, Craiova, 200692, Romania

Alexandre Janer[7], Pol Sola Romeu[8]
Thales Alenia Space, Cannes, 06150, France

A Flight Dynamics Software (FDS) is a space mission's ground-segment operational software in charge of controlling the satellite's orbit during its whole lifetime. In the frame of ESA NEOSAT program, CS România and CSSI are developing a new FDS under the responsibility of Thales Alenia Space. This new FDS will be used for the station keeping phase of satellites based on the full-electric SpaceBus Neo platform developed by Thales Alenia Space. This FDS solution is based on the open source library Orekit and its underlying mathematics library Hipparchus for all the space dynamics calculations involved. It uses web-based technologies and a modular architecture that makes it innovative, portable, secured, and extensible.

I. Introduction

A Flight Dynamics Software (FDS) is a mandatory part of a space mission. Its main function is to allow the ground system operators to determine the orbits of satellites, forecast satellites' events, and plan satellites' maneuvers. The FDS is integrated within the Satellite Control Center (SCC), which is itself a part of the ground control system dedicated to the satellite mission.

In the frame of European Space Agency (ESA) Neosat program, Thales Alenia Space is developing its new geostationary satellite platform called Spacebus Neo. This new satellite platform being based on a full electric propulsion sub-system, existing FDS products did not fit with Neosat system requirements. On this basis, Thales Alenia Space has awarded CS România (CSRo) and CS Systèmes d'Information (CSSI) for the development of a fully new FDS product aiming to be used for all On-Orbit Control activities of Spacebus Neo satellites. This development started in 2016 and is being executed under Thales Alenia Space responsibility; with innovation and genericity in mind.

This FDS is based on the open source space dynamics library Orekit [1] and its underlying mathematics library Hipparchus [2]. The software is based on version 9.2 (latest version) of Orekit and it uses the full extent of capabilities provided by this library.

Many FDS solutions were developed so far by the different actors of the space industry; some generic ones, and some specifically dedicated to a space mission. We will emphasize here on the innovations brought by this FDS; which can be summed up by the following points:

[1] Thematic Engineer, Flight Mission and Operation department
[2] Operational Unit Manager, Business Development department
[3] Project Manager
[4] Software Engineer
[5] Software developer
[6] Software Engineer
[7] FDS Products Manager, CCSL/LPCB/C department
[8] FDS Products Technical Responsible, CCSL/LPCB/F department

- Portability: The FDS design and technology stack guarantees product portability to different operating systems and eases product deployment;
- Security: The product embeds an authentication layer and can be run on a closed network. Also, different users' profiles with different access rights are available.
- Modularity: The software is designed to be modular and extensible. It is easy for a user or a customer to add its own space dynamics module.
- Adaptability: Although the first instantiation of this FDS is tailored for SpaceBus Neo and geostationary orbits, the design allows flexibility and adaptation to, for example, low Earth orbits and/or chemical propulsion.

II. The Neosat program, SpacebusNeo & Flight Dynamics Software

A. The Neosat program

The "Next Generation Platform" Neosat program is part of ESA's Advanced Research in Telecommunications System Program (ARTES). Its goal is to support European industrials in developing the next generation geostationary satellite platform. The main objective is to reduce the cost of such missions by developing platforms that are optimized for electric propulsion.

B. Thales Alenia Space's SpaceBus Neo platform

In the frame of the Neosat program, Thales Alenia Space was awarded a contract in September 2015 to develop its new line of geostationary platforms' product. This platform is named SpaceBus Neo. It is an evolution of the former Spacebus product line that will be offering different propulsion configurations, including a full-electric version. SpaceBus Neo will allow operators to choose in a large choice of versions for any kind of geostationary satellites, from small to extra-large ones. Operators will have the possibility to decide the propulsive configuration of their choice, depending on their needs and budget; from 100% electric to 100% chemical with hybrid solutions in between. The payload capacity of SpaceBus Neo will be up to 2000kg and 20kW, for a mass at launch that will be between 3 and 6 tons. The full-electric version will allow electric orbit raising and electric station keeping. Its payload capacity will be up to 1400kg and 16kW.

C. Flight Dynamics Software

In a nutshell, the Flight Dynamics Software is the ground segment operational tool that is in charge of controlling the satellite's orbit. Its goals are to:

- Determine the orbit (using a GNSS on-board device and/or ground-stations measurements) to locate the satellite precisely;
- Predict the future trajectory of the satellite and predict orbital events (eclipses, remarkable orbit points' crossings, sensors' dazzling etc.);
- Keep the satellite on post and working by applying the station keeping strategies, calculating the orbital maneuvers required and monitoring the propulsion system.

A Flight Dynamics Software is used during all phases of a satellite lifetime, from the Launch & Early Orbit Phase (LEOP - Beginning of life) to the Disposal Phase (end of life), passing through the Station Keeping Phase (operational phase).

As existing FDS products did not fit with Neosat full-electric station keeping requirements, Thales Alenia Space has awarded CS România and CS-SI a contract for the development of a new FDS product line that will be used for the station keeping of SpaceBus Neo satellites. The solution proposed by CS is a web-based architecture using open source tools that will be described further here. The core flight dynamics algorithms are based on the open source libraries Orekit and Hipparchus.

III. Orekit & Hipparchus

A. Orekit

Orekit (https://www.orekit.org) is a space dynamics library that provides accurate and efficient low level components for the development of flight dynamics applications. It is implemented in Java and designed to be used in very different contexts, from quick studies up to critical operations. Its development was started by CSSI in 2002

on capital funds and it was then distributed under the open source Apache License version 2.0 in 2008 [3]. Since that day Orekit has been extensively used and validated by reference companies ([4], [5]) and schools in the European space industry: ESA (European Space Agency), CNES (Centre National d'Etudes Spatiales - the French Space Agency), Thales Alenia Space, ADS (Airbus Defence and Space), EUMETSAT (European Organisation for the ExploiTation of METeorological SATellites), TeleSpazio (Italy), ISAE (Institut Supérieur de l'Aeronautique et de l'Espace - National Higher French Institute of Aeronautics and Space), NRL (Naval Research Laboratory), ADS (Applied Defense Solution) etc.

Orekit is, as of today; in version 9.2 and the latest versions (since 8.0) need a Java platform that is at least 1.8.

Orekit provides many ways to handle spacecraft, orbits and models of physical processes occurring in space. Below is a non-exhaustive list of the features implemented in Orekit. For more details, please visit the website of Orekit:

- Time: High accuracy absolute dates, handling of many common time systems (UTC, TAI, UT1, TT, GPS etc.)
- Geometry:
    - Frames hierarchy supporting fixed and time dependent frames,
    - Predefined frames: J2000, GCRF, ICRF, all ITRF from 1998 to 2014 and intermediate frames (TOD, MOD, GTOD and TEME), Veis, topocentric, local orbital frames, Moon, Sun and planets based frames…,
    - Transparent handling of IERS Earth Orientation Parameters, of JPL DE 4xx ephemerides,
    - Transforms including kinematics effects, composition transforms reduction and caching for efficiency.
- Earth models: Tropospheric and ionospheric models, geomagnetic fields (IGRF, WMM), geoid from any gravity field, tessellation of zones of interest as tiles.
- Spacecraft:
    - Cartesian, elliptical Keplerian, circular and equinoctial parameters, Two-Line Elements, and transparent conversion between parameters,
    - Attitude state and derivatives, Jacobians computation, mass management etc.
- Maneuvers: Analytical models for small maneuvers, impulse maneuvers, continuous maneuvers
- Propagation:
    - Several analytical propagation models (Kepler, Eckstein-Heschler, SDP4/SGP4 with 2006 corrections),
    - Numerical propagator with:
        - customizable force models (central attraction, gravity models, atmospheric drag (DTM2000, JB2008, NRL MSIS 2000…), third body attraction, radiation pressure with eclipses, solid and ocean tides, general relativity, multiple maneuvers…),
        - State of the art ODE integrators,
        - Jacobian computations.
    - Semi-analytical propagation model (DSST [6]) with customizable force models,
    - Tabulated ephemerides,
    - GNSS propagation,
    - Taylor-algebra and automatic differentiation for very fast Monte-Carlo analyzes [7],
    - Discrete events handling during propagation (eclipse, node crossings, stations' visibility, dates, alignments etc.),
    - Ability to run several propagators in parallel and manage their states simultaneously throughout propagation.
- Attitude: Extensible attitude evolution models, predefined laws such as nadir pointing, yaw compensation, yaw-steering, LOF aligned, inertial and many more.
- Orbit determination:
    - Batch least-square filtering for estimation of orbital parameters, force models' parameters (drag or solar radiation pressure coefficients, maneuvers thrusts or flow rates) and measurements parameters (biases, station position, pole motion and rate, prime meridian correction and rate),
    - Extended Kalman filtering for sequential estimation of the same parameters,
    - Multi-satellites orbit determination,
    - Several predefined measurements (range, range rate, azimuth/elevation, right-ascension/declination, position/velocity,

    - Several predefined modifiers (tropospheric and ionospheric effects, station offsets, biases, delays, antenna phase center),
    - Parsing of CCSDS Tracking Data Message file.
- Orbit file handling: Loading of SP3-a, SP3-c and SP3-d orbit files, loading of CCSDS Orbit Data Messages (OPM, OEM, OMM), loading of SEM and YUMA files for GPS constellation, exporting of ephemeris in CCSDS OEM file format.

B. Hipparchus

Hipparchus (https://hipparchus.org) is a library of low level, lightweight and self-contained mathematics and statistics components. It addresses the most common problems not available in the Java programing language. It is a large library that has been split in several modules, allowing users to select only the modules they need. The different modules available are given below:

- Core: Core elements of Hipparchus, all modules depend on it. It contains analysis functions, complex numbers, probability, fractions, linear algebra, random numbers etc;
- Statistics: Basic descriptive statistics, frequency distributions, linear regression, analysis of variance etc;
- Geometry: Classes useful for many physical simulations in Euclidean spaces, like vectors and rotations in 3D, as well as on the sphere;
- Ordinary Differential Equation (ODE): Solvers and integrators for ODE, event handling;
- Optimization: General optimization algorithms for solving linear or non-linear problems;
- Fitting: Curve fitting implementation for different functions;
- Filtering: Implementation of common filters such as Kalman filters;
- Fast Fourier Transform: Transformers for signal analysis;
- Cluster: Operations to cluster data sets based on a user-defined distance measurement.

Hipparchus emphasizes small, easily integrated components rather than large libraries. Its dependencies are limited to the core Java platform.
Hipparchus is, as of today, in version 1.3 and the latest versions need a Java platform that is at least 1.8.
Note that Hipparchus started as a fork of Apache Commons Math that was initiated by most of the main developers and a few contributors of Apache Commons Math.

IV. An Orekit-based Flight Dynamics Software for SpaceBus Neo

As stated before, Thales Alenia Space has entrusted CS România and CSSI the task of developing a fully new FDS product based on Orekit and aiming to be used for all On-Orbit Control activities of Spacebus Neo satellites. This development, started in 2016, is being executed under Thales Alenia Space responsibility, with innovation and genericity in mind.

CS România, as the prime contractor, is in charge of developing the architecture of the FDS and the main core flight dynamics module. CSSI is providing expertise in space mechanics to CS Rômania while also developing the orbit determination module and the co-location and collision avoidance module. Thales Alenia Space, on its side, is developing all the flight dynamics modules related to electric propulsion (station keeping module, relocation and disposal module, maneuver commanding and reconstruction module etc…).

In this part we will describe further this Orekit-based FDS for Spacebus Neo and show the different tasks performed by the actors of the project. We will emphasize on the following aspects of the FDS: its portability, its modularity, its extensive use of Orekit functionalities, its key security features and finally its adaptability or extensibility.

A. A web-based architecture for a portable Flight Dynamics Software

The FDS design and technology guarantees product portability to different operating systems and eases product deployment. The technology stack of the FDS is shown on Fig. 1. Below is a quick list of the key technologies used in the FDS:

- Development is based on Java for the space dynamics functions using the Orekit library,
- The Graphical User Interface (GUI) is web-based and uses Angular JS, JavaScript, HTML and CSS,
- The rendering on screen of data and plots is ensured through the use of COTS (Commercial Off-The-Shelf) products such as jqPlot, d3js or jsPDF,
- The server part uses Apache Tomcat and the Spring framework,

- The data are handled on the server side using PostgreSQL and, of course, an underlying file system to store the results and communicate with the Spacecraft Control Center (SCC),
- Note that most of the space dynamics data exchanged between the SCC and the FDS are based on CCSDS formats. Thus, the data is easily shareable with other industrials or institutional satellite operators.

Built on a web-based design, product deployment is easier: satellite operators simply have to connect to a server via a web-browser to use the FDS, thus avoiding any issue regarding their operating system model, software libraries version or administration rights on their computers. Technically, the operator machine can be very light as all the heavy calculations are done on the server side.

This architecture allows the centralization of data and calculations, which in turn improves sharing information; a new dataset produced and validated by an operator is immediately available to all operators working on the same mission and with the proper visibility rights.

Fig. 1 shows the distribution of the task between Thales Alenia Space and CS (CS România and CSSI); distribution that is detailed below:

- Thales Alenia Space provides its "Mainframe GUI". It is a set of tools and widgets in Angular JS that is used in most of the web-based products developed by Thales Alenia Space. On their side, it allows consistency in the GUI of their different software; an operator using several of their web-based products would immediately be comfortable using them as the look-and-feel, the main widgets and the sets of color would be identical.
- Thales Alenia Space also provides the flight dynamics modules related to the propulsion system and station keeping strategies. These modules will be detailed in section B of this part. They are provided as black-boxes to CS;
- CS is in charge of developing the architecture of the software and linking all the different COTS together to deliver a fully functional product;
- CS is also developing the GUI of the FDS for all the modules;
- Finally, CS is developing all the flight dynamics modules that are not related to the propulsion system. Once again, the modules will be detailed in section B of this part.

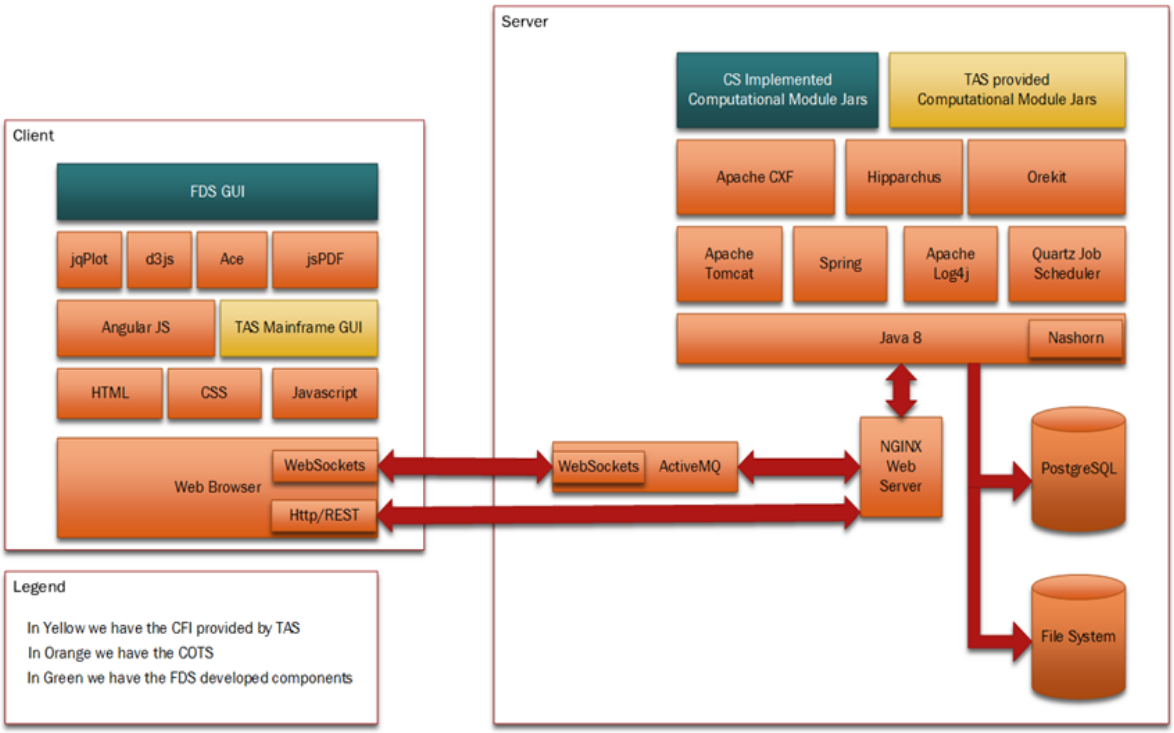

Fig. 1 – Technology stack of the Orekit-based FDS

An example screenshot of the FDS is shown in Fig. 2. The screenshot is taken from the orbit propagation module of the FDS, it shows:

- A black banner on the top with the name of the tool in the center. On the right there is info on the date, the user name and a help button. On the left is a drop-down menu allowing quick access to other modules of the FDS.
- Below the black banner is another top banner showing the current flight dynamics module being opened and allowing for the selection of the satellite that is being operated.
- The inputs/outputs sections of the module occupy respectively the left side and right side of the screen. A control arrow allows for expanding either the inputs or the outputs for full screen visualization.
- Inputs are organized in thematic tabs; their name and content depend on the purpose of the module. Generally the input tabs organization follows a logic that is based on a sequential selection of the inputs, from the first obvious thing to select on the left (example: a date here, a starting point for the propagation, a duration of propagation), to the last fine tuning of the algorithm used on the right (here the propagator and force models selection). Also, the first input tabs on the left are usually inputs that will always have to be filled by the operators (typically dates, input files selection etc.); whereas the last tabs on the right contain inputs that will seldom be changed by operators (fine tuning of space dynamics functionalities for which usually the default configuration is enough).
- The visibility of the inputs of the different modules and the possibility to modify them will depend on the operator profile. The profiles are described in details in section D of this part.
- The outputs are also organized in tabs. They are always the same:
    - Data: Contains the data outputs of the module, which is a synthetic view of the calculations performed by the module.
    - Plots: The plots available for the module. They are based on jqPlot. The user can display all of them or just a subset, open them in a new tab, zoom on them, print them or save them on his machine.
    - Files: The files produced by the execution of the module. In the case of the orbit propagation module it will be an ephemeris file, in the CCSDS OEM format.
- Above the inputs some control buttons allow the operators to:
    - "Run" the module;
    - "Cancel" the execution of the module;
    - "Validate" the calculations of the module. This last operation implies that the results obtained are sent to the persistence database and made available for all the other operators working on the FDS of the same satellite.
    - "Load or Save Context": A context is defined by a set of inputs and outputs representing a former run or configuration of the module. The operator can save its own contexts "locally" when he is using for example multiple iterations of a module to fine tune some results.
- Beneath all this is a logbook that can be extended or collapsed. It contains high level information on the execution of the module (start, stop, status etc…)

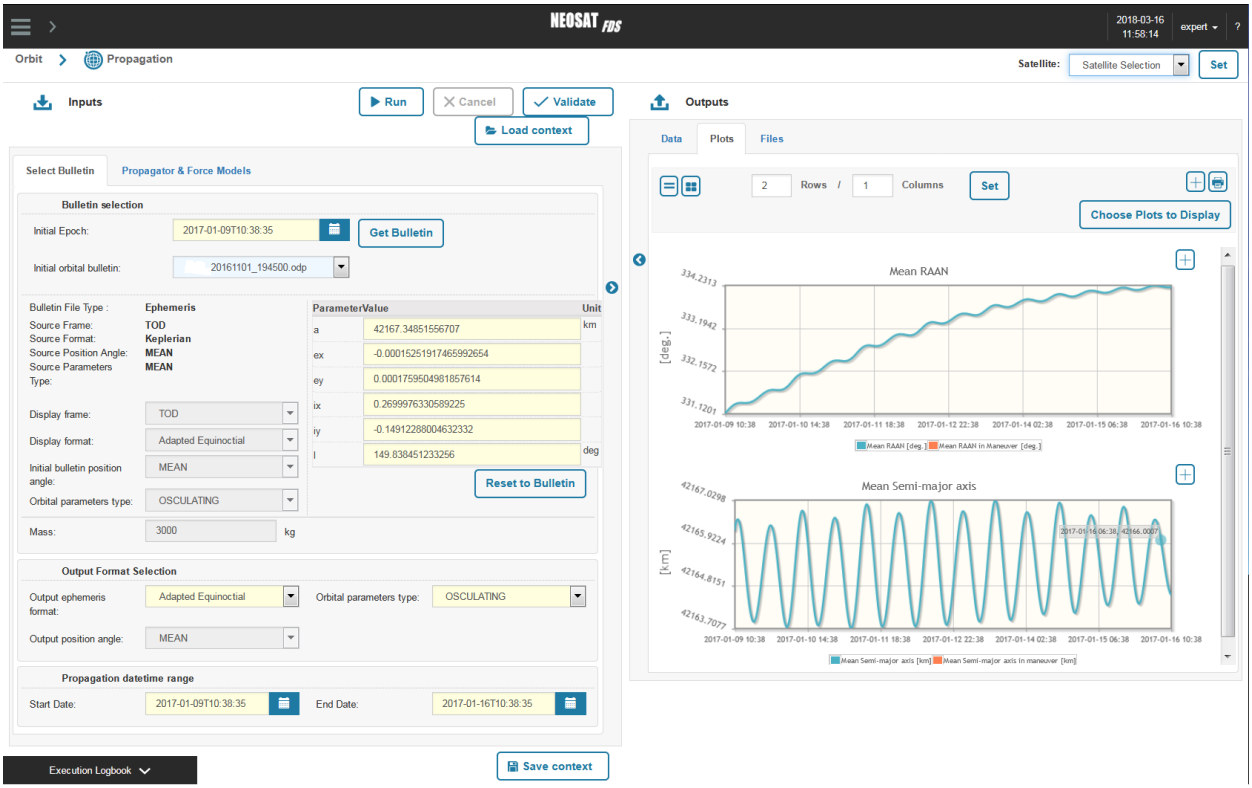

Fig. 2 – Example Screenshot of the FDS – Orbit Propagation Module input/output

B. A modular Flight Dynamics Software

The FDS is designed to be modular and extensible. It is very simple to add a new module on a customer demand. As decided with Thales Alenia Space, capability to integrate Customer Furnished modules as black boxes is also possible. Through this mechanism, the customer is ensured that his most sensitive technology will not be disclosed to any third party.

That is essentially what Thales Alenia Space is already doing on this version of the FDS tailored for Spacebus Neo. Thales Alenia Space is indeed developing its own modules for everything that is related to the electric propulsion. Thales Alenia Space wants to keep its algorithms and technology hidden from the public, as they required a lot of R&D investment to be developed.

Thus they start with developing their own flight dynamics modules on their side. Then they deliver them to CS România (as obfuscated jar binaries) who integrates them as black boxes in the global software. In the meantime, CS România has developed the GUI related to this module. All the integrator needs to know are the interfaces of the module (inputs and outputs); the core algorithm of the functions stays hidden and never has to be disclosed.

Fig. 3 presents the different modules that have been created for the FDS of Spacebus Neo. The color code shows the different entities in charge of developing each module. This is a good example of the modularity and flexibility of the architecture. We have three customers or developers (Thales Alenia Space, CS România and CSSI) and one integrator (CS România). The three developing entities are in three different locations (Cannes/France, Craiova/România, and Toulouse/France) and yet they can easily develop their own modules according to the structure of the FDS (capabilities, interfaces…).

Fig. 4 also shows the different modules of the FDS. It is a screenshot of the home page of the FDS, available right after the operator has logged in.

CSSI and CS România are developing the flight dynamics modules dealing with:

- Orbit determination - using ground stations' or GNSS measurements, through a least-square method or a Kalman filter;
- Orbit propagation and ephemeris generation - using a numerical propagator or a semi-analytical propagator (DSST: Draper Semi-analytical Satellite Theory) for the computation of mean orbital elements;
- Orbital events prediction – satellite's apsides, nodes, maximum latitude or longitude, eclipses, transit of the Sun/Moon for a ground station or any satellite's sensor etc.;
- Ground antenna visibility calculation;
- Maneuvers' calibration;
- Colocation's strategy survey and close-approach management (collision avoidance);
- Orbit import/export and conversion functions.
- Historical Data Management: handling, storing, presenting and organizing all the data produced by the FDS during a space mission time;
- Batch Dashboard: this module allows the user to script its own sequence of FDS modules execution without requiring the GUI of those modules. It is used for automation of the routine tasks usually performed in satellite operations.

On its side, Thales Alenia Space is developing all modules related to the electrical station keeping, meaning:

- Electric station keeping and momentum dumping maneuvers calculation;
- Relocation and disposal maneuvers;
- Maneuver reconstruction and consumption computation;
- On-board orbit propagator initialization.

Based on the modular architecture of the product, these modules are fully integrated as black boxes within the FDS, making it a fully integrated solution for all on-orbit control activities of Spacebus Neo satellites.

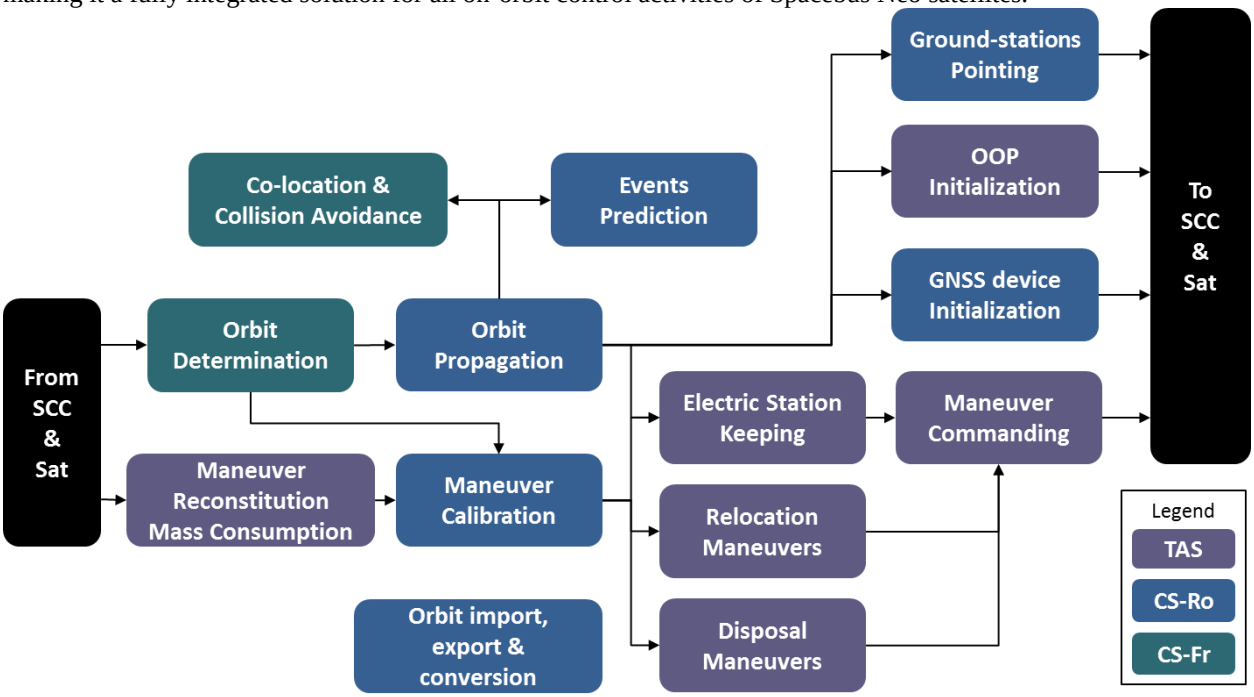

Fig. 3 – General view of the FDS modules and their sequence of activation

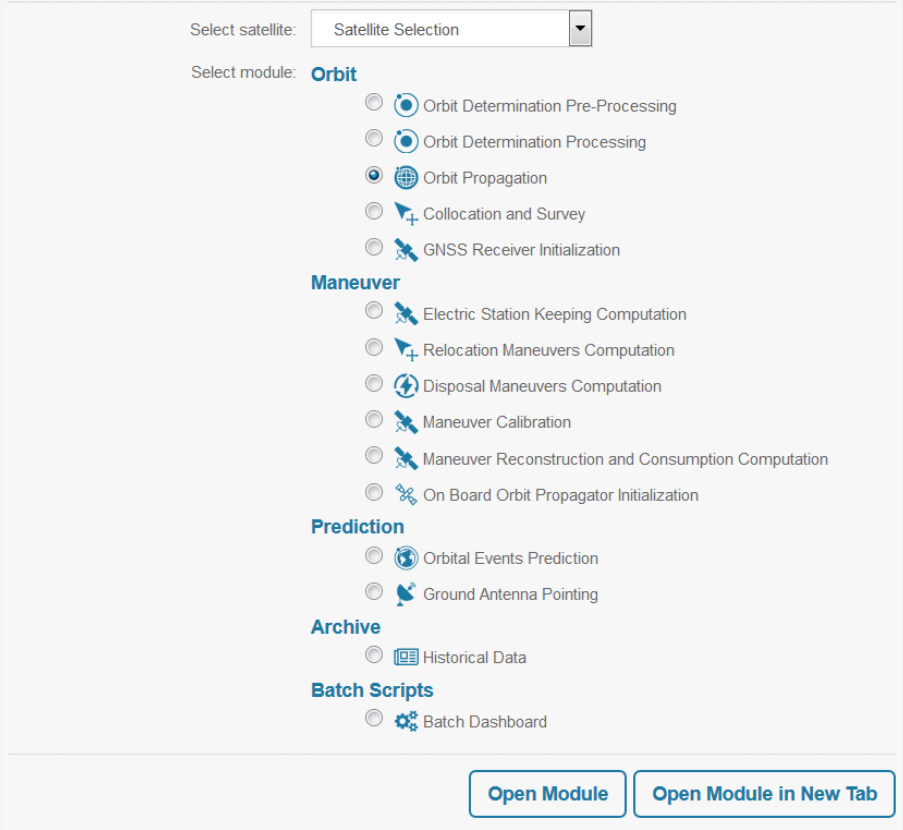

Fig. 4 – FDS homepage, the different modules available to an operator

C. Orekit usage in the Flight Dynamics Software

In this part, the usage that is made of Orekit in the FDS is briefly described, as a bullet list with some explanations, as we do not want to enter in a detailed implementation. We simply want to show here that (almost) the full extent of the library is being used to implement the space dynamics functions.

- Orbits:
  - Type: Cartesian, Keplerian, Equinoctial,
  - Frames: All frames defined in IERS 2010 conventions
    - Inertial: EME2000, TOD, Veis…,
    - ECEF: All ITRF, WGS84,
    - Topocentric frames for ground stations.
- Orbit conversions:
  - Read/Write NORAD Two-Line-Elements,
  - Read JSpOC Conjunction Data Messages (CDM).
- CCSDS formats:
  - TDM - Tracking Data Messages: Antennas/Ground stations measurements,
  - ODM - Orbit Data Messages
    - OEM - Orbit Ephemeris Message,
    - OPM – Orbit Parameters Message.
- Orbit Determination:
  - Batch least-square orbit determination ,
  - Real-time orbit determination with an Extended Kalman Filter,
  - Ground stations measurements supported:
    - Distance: Range and turn-around range,
    - Angular: Azimuth/Elevation or Right-Ascension/Declination.

    - Raw data measurements from an eventual on-board GNSS device can be treated by the orbit determination functions,
    - Atmospheric corrections: Ionospheric and tropospheric delays handled for all ground-stations' measurements,
    - Estimations:
        - Orbital parameters (all or a subset),
        - Force models parameters: solar radiation pressure and electric maneuvers' thrusts,
        - Measurements parameters: stations' biases, stations' offset in position, satellite range bias etc.,
        - Covariance's estimation is performed along with the estimation of all the parameters.
- Orbit Propagation:
    - Mean element propagation with the DSST (Draper Semi-analytical Satellite Theory),
    - Osculating element propagation performed either with DSST or a numerical propagator,
    - Force models include: Spherical harmonics Earth potential (configurable), Luni-Solar gravity perturbations, solar radiation pressure modeling, electric maneuvers modeling.
- Orbital Events:
    - Apogee & perigee crossings,
    - Nodes crossing,
    - User defined solar time crossings,
    - Eclipses of the Sun an obscuration ratio (by the Earth or the Moon),
    - Transits of the Sun and the Earth in satellites' sensors.
- Ground Antenna Pointing - Stations visibility with:
    - Minimum elevation,
    - Azimuth/elevation masks,
    - Atmospheric refraction.
- Co-location and collision avoidance:
    - Multi-satellite optimized propagation,
    - Close approach detection,
    - Avoidance maneuvers calculation using analytical models as first guess,
    - Co-location strategies: longitude separation, eccentricity and inclination separation,
    - Set for a max of 10 co-located satellites (in first instance).

D. A secured Flight Dynamics Software

Here we emphasize on the security features embedded in our Orekit-based FDS. Although the architecture and GUI are web-based there is of course no need to be connected to the Internet to run the FDS. It can be run on a closed network inside the Satellite Control Center (SCC) with no interface with the outside world.

1. Authentication

An authentication layer is embedded in the server part, ensuring customers that their system is safe from external intrusions. The access rights are managed by the administrator of the system on the customer side.
Fig. 5 shows the splash window of the FDS with the authentication login.

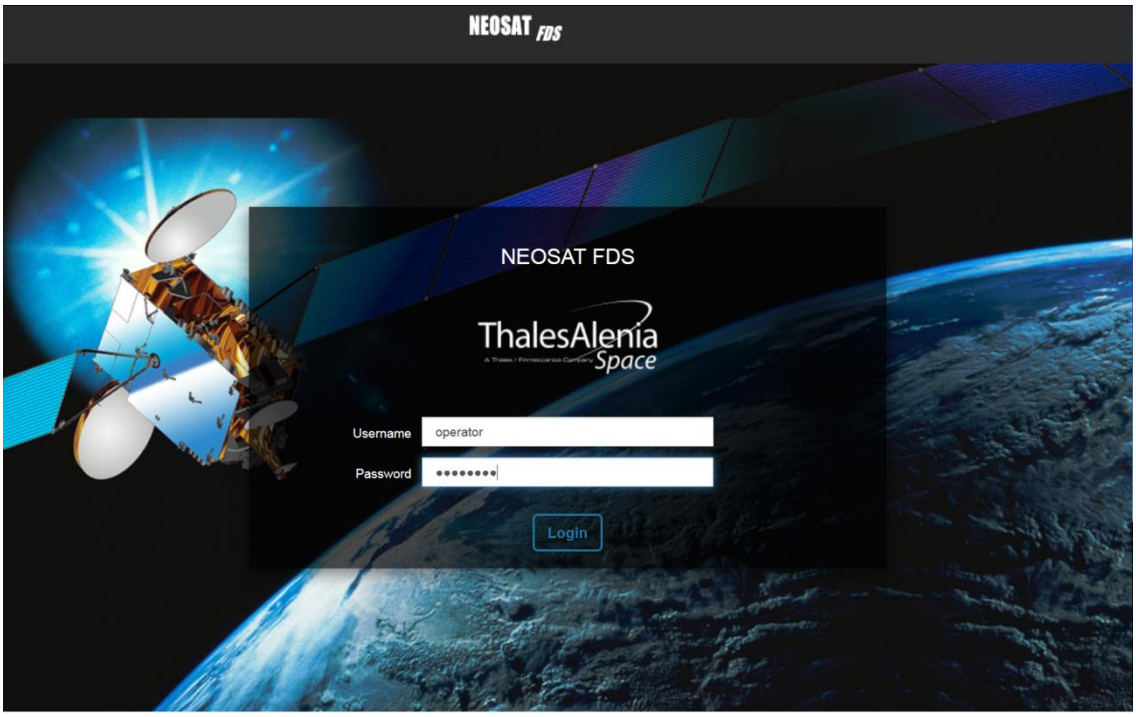

Fig. 5 - Splash window of the FDS showing the authentication layer

2. Different profiles for different tasks

The FDS exposes different kind of users' profiles that will allow them to carry different tasks out:

- **The operator**: He/She has access to the very basic functionalities of the FDS. It is the type of profile that will be the most common in the FDS. The operator can determine an orbit, propagate it, calculates maneuvers, performs some routine tasks etc.…
- **The expert**: The expert is either from the company that is operating the FDS or detached from another company (typically the one that built the satellite). He/She usually is more knowledgeable in space dynamics than the operator. He has the same access than an operator plus access to an extensive capacity of tuning the space dynamics algorithm of the FDS through the GUI. The expert will generally be here at the beginning of life of the spacecraft, during the commissioning phase. He can also be called in case a problem is detected on a satellite and the operators are not able to sort it out themselves.
- **The administrator**: There will usually be only one (or two for backups) administrator on a given mission. He has an extensive access to the database and historical data management. He is also the one who can add new users' profiles or delete old ones. He has been given proper clearance and authority by the industrial operating the FDS.

E. An extensible Flight Dynamics Software

We have seen that the first instantiation of this Orekit-based FDS is tailored for geostationary orbits and a spacecraft with electric propulsion. Nevertheless the software design makes it adaptable and extensible to almost any other types of mission. Developing a version for Low Earth's orbits (LEO) or a chemical or hybrid propulsion system is made easy by the built-in modularity of the FDS.

Implementing a LEO version implies configuring the database for handling the drag force; and Orekit already implements several models of atmosphere and satellites geometry just for this purpose. Also, the station keeping strategy, the relocation and disposal maneuvers have to be adapted. But it only implies replacing three modules of the FDS with three new ones implementing a new algorithm that can also be based on Orekit. Scripting new GUI and implementing new interfaces is quite easy with the architecture and the technologies chosen, and it does not require an entire new software development process.

Adapting the FDS for chemical propulsion also requires changing the station keeping strategy, along with most of the models dealing with the propulsion system (maneuver reconstruction and mass consumption, maneuver calibration, maneuver commanding). But again the core of the FDS will not fundamentally be changed by this evolution. Moreover, the FDS is also designed to manage three-axis stabilized and spin-stabilized spacecrafts.

Also, the software being designed for easily switching between different satellites, it has been thought to be extended in the future for orbit determination and maneuver planning of a constellation of satellites. It is to be noted on that aspect that Orekit already supports multi-satellite orbit propagation and orbit determination.

V. Conclusion

A new Flight Dynamics Software (FDS) product is presented here. It is developed by CS România and CSSI under Thales Alenia Space responsibility. It is based on the open source library Orekit for all the space dynamics calculations. It has a web-based architecture using client-server communication and is portable to any type of operating systems. It is secured by an embedded authentication layer and through the availability of different users' profiles with different access rights to its functionalities. It is modular by design and allows for quick development and integration of new space dynamics functions. The first instantiation of this FDS is tailored for geostationary satellites with full-electric propulsion. It is destined to be used for the station keeping of satellites based on the SpaceBus Neo platform developed by Thales Alenia Space in the frame of ESA's Neosat program. However, the modularity and extensibility of the FDS makes it usable in the future for Low Earth's Orbits mission and satellites with a chemical or hybrid propulsion system.

References

[1] Maisonobe, L., et al., "Orekit, An accurate and efficient core layer for space flight dynamics applications,", 2008. URL https://orekit.org/.

[2] Steitz, P., et al., "Hipparchus: a mathematics Library,", 2016. URL https://hipparchus.org/.

[3] Maisonobe, L., Cefola, P. J., Frouvelle, N., Herbinière, S., Laffont, F.-X., Lizy-Destrez, S., and Neidhart, T., "Open Governance of the Orekit Space Flight Dynamics Library," International Conference on Astrodynamics Tools and Techniques, ESA/ESTEC, 2012.

[4] Ward, E. M., Warner, J. G., and Maisonobe, L., "Do Open Source Tools Rival Heritage Systems? A comparison of tide models in OCEAN and Orekit," AIAA/AAS Astrodynamics Specialist Conference, American Institute of Aeronautics and Astronautics, 2014.
doi:10.2514/6.2014-4429.

[5] Bernard, N., Maisonobe, L., Barbulescu, L., Scortan, S., Cefola, P. J., Casasco, M., and Merz, K., "Validating Short Periodics Contributions in a Draper Semi-Analytical Satellite Theory Implementation: the Orekit Example," ISSFD, 2015.

[6] McClain, W. D., "A Semianalytic Artificial Satellite Theory," Tech. rep., Charles Stark Draper Laboratory, 1992.

[7] Antolino, A., and Maisonobe, L., "Automatic differentiation for propagation of orbit uncertainties," Stardust final conference, ESA/ESTEC, 2016.