# SIRIUS-DV: The new Flight Dynamics algorithms for the future CNES missions

Iván Llamas,[1] Jean-Jacques Wasbauer,[2] Yannick Tanguy,[2] and Michel Lacotte[2]
*[1] GMV, Spain; [2] CNES, France;*
*illamas@gmv.com, Jean-Jacques.Wasbauer@cnes.fr, Yannick.Tanguy@cnes.fr, Michel.Lacotte@cnes.fr,*

**Keyword :** SIRIUS, Flight Dynamics, Algorithms

The SIRIUS project aims to develop a set of Flight Dynamics products that will be used operationally in the control centers of the upcoming CNES missions. It mainly covers three different layers: the mathematical low level libraries (PATRIUS), intended to be used either in an operational environment or in expert studies; the flight dynamics algorithms, implementing the operational functionalities (SIRIUS-DV); and the FD applications, that include the assembly of the algorithms to build stand-alone applications – with dedicated GUI - and the infrastructure services (such as time, messages, logging, …) needed in an operational FDS.

Due to its unique architectural conception, SIRIUS provides a higher level of flexibility (so as to be easily adapted to any future mission, almost in a “plug and play” manner) and scalability (the effort to add new functionalities is reduced) with respect to other state-of-the-art systems. The choice of technologies used in the line of products also guarantees its non-obsolescence up to - at least - twenty years from now.

This paper focuses in the second layer, the software applications implementing the flight dynamics algorithms that are divided in several technical domains:

- Conversion: Dates conversion, orbital/attitude parameters conversion.
- Ephemeris Generation: orbit propagation and ephemeris generation.
- Events: events/phenomena calculation.
- Scenario: Functionalities dealing with the data scenario, which represents the whole mission of a given satellite.
- Orbitography: measurement treatment, orbit determination, collision risk assessment.
- Orbital Maneuvers: orbital maneuvers computation, station keeping.
- Guidance & Programming: AOCS programming, guidance, constraints checking.
- Mission: reference orbit calculation
- Monitoring: monitoring of TM parameters, thresholds checking.
- Interfaces: External data retrieval/production.
- Scenario Processings: treatment of the different parts of the data scenario, such as trajectory, attitude, maneuvers, MCI, thrusters, tanks and solar arrays.

These algorithms rely on a data model managed by the CNES domain experts and which is updated gradually as the development advances. It contains the definition of all the data that are used in the algorithms, the definition of the interfaces (inputs/outputs) of each algorithm and the software requirements that the implementations must meet. Using this model as input, the implementations of both the data and algorithms interfaces are automatically generated (using a code generator that is also part of the SIRIUS line of products), which serve as starting point for the development carried out by the team.

The SIRIUS-DV applications are developed in Java using an Agile/SCRUM methodology with sprints (realization iterations) lasting four weeks. The functionalities to be developed in a given sprint are presented (at the beginning of each sprint) to the team by the CNES domain experts. During the sprint a constant communication flow is established between both parties in order to ensure the understanding – and hence the quality – of the tasks to be done (the development team being physically located at CNES premises). At the end of each sprint, those functionalities that are finished are presented to the users by the development team, so a fully usable product is available once a month, with increased functionalities over time.

This paper gives a brief description of the development process of SIRIUS-DV and describes the key concepts of this new line of flight dynamics algorithms, the data model and its impact in the developed software and several of the most representative applications, paying special attention to the architectural design of the propagator and its link with the data scenario, since it constitutes the core of the system.