Software architecture for the automatization of the ground segment of the OUFTI-1 CubeSat

Ngoc An Nguyen*, Laurent Chiarello*, Amandine Denis†, Jonathan Pisane*, Jacques G. Verly*

**Dept. of EECS, University of Liège, Grande Traverse 10, 4000, Liège, Belgium*
Tel: +32 4 366 2642, Fax: +32 4 366 2989, Email: jpisane@ulg.ac.be
†*LTAS, University of Liège, Chemin des Chevreuils 3, 4000, Liège, Belgium*

The context is the educational nanosatellite OUFTI-1 under development at the University of Liège. The project comprises the usual ground and space segments. The ground segment is divided into a Mission Control Center (MCC, mainly consisting of software) and a Ground Station (GS, mainly consisting of hardware, such as transceivers and antennas). We focus here on the software components (or tools) of the MCC, and their interactions. The GS must be contollable both manually and automatically.

The GS (principally the MCC) uses a client-server software architecture. The server software, called the Mission Control Software (MCS), is the center of the system. The MCS ultimately sends all the telecommand (TC) messages up to the satellite, and is the first to see the telemetry (TM) messages coming down from it. The rest of the software tools handle the TCs and TMs to and from the MCS. We now briefly discuss these clients.

Users with little programming experience and only a basic knowledge of the operation of OUFTI-1 can use the TC planner (tool) to produce scripts of TCs, this under tight, supervisory control. This tool can handle advanced control strategies, such as conditional execution. The TCs so generated are passed to the Scheduler, which decides when to send them to the MCS for immediate formatting and sending to the satellite. Advanced users can use the TC sender to directly send TCs to the MCS for immediate processing.

The MCS is connected to two databases. The TC/TM database stores a log of the complete history of all TCs and TMs. The Management Information Database (MIB) contains complete information about all aspects of the mission, such as the orbit parameters and the range of acceptable battery temperatures.

When TCs are generated via the TC planner, the user can follow the history of TCs and TMs via the Mission Data Viewer. It is of course essential that the MCS be always active. This means that it should be located in a server computer, at a fixed location. All of the other software tools are clients, and can thus be located anywhere.

Much of the above architecture and has been implemented and tested on synthetic examples. Development and tests are continuing.