Paper ID: 10104
oral

SPACE SYSTEMS SYMPOSIUM (D1)
System Engineering Tools, Processes & Training (I) (3)

Author: Mr. Damiano Guerrucci
European Space Agency (ESA), Germany, damiano.guerrucci@esa.int

Ms. Rachel Majchrzak
European Space Agency (ESA), Germany, Rachel.Majchrzak@esa.int
Mrs. Emanuela Bordoni
European Space Agency (ESA), Germany, emanuela.bordoni@esa.int

A COMMON MISSION CONTROL SYSTEM FOR THE ESA EARTH OBSERVATION MISSIONS

Abstract

The European Space Agency (ESA) has a long experience in software reuse, keeping in mind the spirit of reducing cost and maintaining the quality. One of the biggest achievements has been the Earth Observation mission control system (EOMCS) kernel software that is implementing a big set of functionality common between missions of the same nature, namely the Earth Observation (EO) missions.

The concept started with the development of Cryosat and Goce mission control systems in the last 2003 and was further developed in the following years. Today not only those missions are flying using the EOMCS kernel, but every new EO mission is basing its development on it. The EOMCS software is based on the ESA Software Infrastructure (SCOS-2000) that alone provides a large and stable system with functionality covering spacecraft monitoring and control. The EOMCS extends the infrastructure with most of the common functionality between EO missions and maximise its reuse through a generic development.

The paper will first explain the motivation and the expectations that made the initiative start, and after giving some historical background and major steps in the process, it will bring the reader to the current status and the associated process. The most interesting aspect is that this software is not an infrastructure as such, instead representing a large part of software that is evolved along the time where the strongest contribution is always coming from the mission preparing for a launch, since this is the most interested in driving the development. On one side the approach can be simpler than an infrastructure software project; on the other hand, it is requiring different innovative approaches for the maintenance and a formal board to coordinate the evolution for a common benefit.

The paper will describe in details the keys of this concept. For example how the integration of a new infrastructure software can be streamlined, or why the implementation of new feature is pushed towards a generic implementation, how generic functionality can be even repatriated to the below infrastructure layer. In general, it will describe the cost benefits that apply to the overall software development cycle.

In conclusion, the paper will also describe the relevance of setting a proper test environment, that shall make use as much as possible of automation, providing a representative environment of each mission using the kernel, but also a generic one where the software can be tested in isolation.