29th IAA SYMPOSIUM ON SMALL SATELLITE MISSIONS (B4)
Small Satellite Operations (3)

Author: Mr. Yuri Matheus Dias Pereira
EnduroSat AD, Bulgaria

Mr. Lubomir Toshev
EnduroSat AD, Bulgaria

SHARING MISSION DATABASES ON A SOFTWARE-DEFINED SATELLITE

Abstract

Satellite platforms often consist of subsystems developed by different parties, who may have their own way of sending commands and parsing data from their payload, with different needs, formats, and expectations. Coordinating and integrating these in a single satellite platform is a challenge to any provider, which is acknowledged by the satellite operator community and led to the development of the XML Telemetric and Command Exchange (XTCE) standard for sharing mission databases between providers. XTCE is an OMG standard and is adopted by the CCSDS, but it has some limitations. The tools to validate and use the XTCE XML files to parse mission data are only available on the Java ecosystem, forcing engineers to either incorporate those Java tools and libraries or make their own ad-hoc parsers, which is not trivial. This leads to XTCE having poor open-source support. In this work, we show an alternative way of solving this problem, by using a tool that allows for the definition of mission databases, and also the generation of data parsers in different languages from those definitions. This tool is based on the Franca IDL language used by the automotive industry and allows for the generation of software code for the serialization and deserialization of mission data, either on the satellite, on the ground for mission operations and even in cloud-based solutions for multiple satellites. The performance and ease of use of this tool on a large-scale database of small satellite missions is compared with similar open-source alternatives. We also compare the use of Franca to the effort of creating an open-source XTCE based parser for the JavaScript ecosystem. The results show that the use of Franca shortens the development of new parsers, as they only need to be written once for working parsers in multiple languages to be generated. Furthermore, being able to customize the code allows for increased flexibility to add features, customization of different parsers for specific needs and even performance improvements. However, it has the drawback of needing the creation of the code generators for new languages, which may not support all parser needs equally.