74th International Astronautical Congress 2023

Paper ID: 77932
oral
student

IAF SPACE SYSTEMS SYMPOSIUM (D1)
Space Systems Architectures (2)

Author: Prof. Rene Hexel
Griffith University, Australia, r.hexel@griffith.edu.au

Prof. Paulo de Souza
Griffith University, Australia, paulo.desouza@griffith.edu.au
Mr. Callum McColl
Gilmour Space Technologies, Australia, callum.mccoll@gspace.com
Mr. Morgan McColl
Griffith University, Australia, morgan.mccoll@griffithuni.edu.au
Dr. Aaron Pereira
Griffith University, Australia, aaron.pereira@griffith.edu.au
Dr. Gervase Tuxworth
Griffith University, Australia, g.tuxworth@griffith.edu.au

HIGHLY FLEXIBLE AND EASILY SCALABLE SOFTWARE ARCHITECTURES FOR SPACE APPLICATIONS

Abstract

Software-defined architectures are at the forefront of sophisticated industrial applications. The Software-Defined Vehicle has revolutionised the automotive industry by allowing the evolution of the system after manufacture. Space as an industry can benefit tremendously from this concept as launch deadlines become decoupled from payload requirements through in-flight software updates. This process is particularly beneficial in the satellite market, where leveraging the existing hardware can realise new capabilities during flight through links to software-deployment services on the ground.

This concept is not without its challenges, as software must skew towards scalable architectures to support a dynamic codebase. We thus show the ability to create reusable and composable software from the base hardware abstraction to the application-level implementation. Software of this form is deployable to multiple hardware architectures, provided the platform contains compatible hardware.

We first demonstrate our make infrastructure that removes the hardware specification from the needs of software interfacing with it. We show how to create scalable software integrating multiple hardware architectures, including ARM-based CPUs such as the M4 and FPGAs. We show that such a software architecture can integrate easily with CI/CD pipelines promoting continuous hardware-in-the-loop testing and integration. Lastly, we show how our infrastructure can perform service-based software updates that upload only the minimum amount of software required to target hardware. This procedure includes partially modifying compiled artefacts to reduce the bandwidth required for software updates, drastically reducing the amount of data transmitted.