7th ESA Workshop on Avionics, Data, Control and Software Systems - ADCSS 2013

Tuesday, 22 October 2013 - Thursday, 24 October 2013

European Space Research and Technology Centre (ESTEC)

Scope & Topics

ADCSS 2013 will address a set of key topics related to the design and implementation of present and future avionics systems. The programme is structured in 5 main tracks, as follows:

SAVOIR (22 October AM)

A significant effort is being deployed by Agencies and Industry to streamline the development, validation and operation phases of spacecraft, with particular focus on the Avionics. This effort is being coordinated by ESA in the form of the "Spacecraft Avionics Open Interface Architecture" (SAVOIR) initiative. SAVOIR brings together ESA and industry experts in an open forum and is gaining significant momentum. Based on establisment of reference architectures, it provides the ground for the identification of building blocks interacting through standardised interfaces, service access points and protocols across hardware and software boundaries.

The session will be introduced by a general presentation of the overall concept complemented by more focused presentations dealing with lower level elements such as equipment specifications, interfaces to AOCS sensors and actuators together with advanced software concepts. The interaction between ground and space systems is tackled as well by considering implementation of file based operations and underlying concepts.

CAN Bus in Space (22 October PM & 23 October AM)

In space avionics we are witnessing a change from highly centralized intelligence to distributed autonomous functions, thanks to the availability of high capacity FPGAs and microcontrollers that offload tasks alternatively concentrated in the on- board computer.
The glue of this change are the command and control buses, and a similar process led in the late 80s to the development and successive adoption of CAN as an automotive and industrial automation bus.

The ADCSS CAN track will host keynote from the Managing Director of CAN in Automation, Holger Zeltwanger. It will include tutorials, exhibits together with presentations on CAN bus physical and protocol layers. Additionally an overview of modern design tools and method commonly used by automotive industry in design, development, production and maintenance of CAN bus networks for safety critical applications will be given.

Avionics Based on Ethernet Networks (23 October PM)

Ethernet is one of the most widely used communication interfaces for non-real-time terrestrial applications. Its use is still expanding in the industrial and automotive domains in part due to the introduction of enhancements to the original Ethernet specification to meet real time requirements. There are many variants of real-time Ethernet protocols (e.g. Profibus, Powerlink or AFDX) and recently the TTEthernet implementation from TTTech has received interest from the space environment by being adopted for the NASA Orion capsule. This session will overview the concepts behind TTEthernet and examine actual and potential use cases in space applications. Products and prototype implementations will be presented. The session will be introduced by Professor Hermann Kopetz; leading proponent in fault-tolerant and time-critical computing and the chief architect of TTP, the predecessor of TTEthernet."

High Performance Computing for GNC (24 October AM)

The objective of this track is to explore the requirements and possible solutions of complete avionics able to cope with very demanding autonomous closed loop controlled space applications. The complete avionics solution shall comprise the computer and data handling parts, the real-time software aspects, as well as the guidance, navigation and control algorithms.

This track begins by giving current examples of missions with a high processing demand in GNC. Current demanding avionics applications are represented by Active Debris Removal (ADR) GNC systems and planetary exploration scenarios with stringent GNC requirements. Currently, ADR is by far the most demanding scenario in which we need high image processing power to be able to allow the closed loop control system to grab the target and de-orbit the entire compound target-chaser. In this scenario, the image processing algorithms will need to be allocated into highly powerful computing systems running specialised real-time software components. In both of the selected cases, there is a bottleneck in the processing of sensor information and the understanding of the environment. These use cases require sensors to extract basic GNC information such as relative position and velocity by applying a high-level understanding of the scene or object being observed. In the Active Debris Removal (ADR) case, this high-level understanding is the measurement of the relative attitude of a tumbling piece of debris, without needing any measurement aids such as optical markers or RF beacons. In the Lunar Landing case, this high-level understanding is the measurement of the relative position and velocity of the satellite with respect to the local terrain during descent. In both cases, the object being measured is poorly modelled and undergoing dynamic motion, and yet GNC information is required in a timely manner with minimum measurement errors.

Background:
The continuous increase in available processing performance has enabled a matching increase in the performance and capabilities of spacecraft Guidance, Navigation and Control (GNC) systems. Automated systems such as the ATV have demonstrated significant capabilities that were only recently the sole domain of human-in-the-loop operations. This success continues to drive forward the potential applications of satellite autonomy and performances to meet new challenges, requiring ever greater levels of processing and autonomy.

New missions are being proposed that once again were only recently assumed to require human operators due to the level of complexity involved. High-level understanding of dynamic and complex environments is now needed, with sufficient speed to enable closed-loop control. In some cases, human operations are not possible due to long communication delays, and in other cases the dynamic environment itself prevents continuous communications. With sufficient understanding of the environment, local decision-making can react faster, with greater accuracy and with potentially lower costs.

Objectives of the round table:
The round table shall consider the two use cases with a view to clearly identifying the processing needs. The objective of the round table is to collect inputs, share experiences and discuss topics that encompass these needs. Open discussions shall highlight useful ideas as well as possible avenues for technology development. The round table will try to set light to the definition of the techniques and technologies required, departing from the exiting state of the art in ESA, as well as discuss about the design and optimization of the hardware and software areas in order to

successfully perform the GNC functions on-board.

Software Factory (24 October PM)

Since decades, many attempts have been made to reuse software code, be it specific or generic. The approach has suffered many difficulties, while the concept of domain engineering was progressing around the notion of “variability factors”, i.e. the elements that allow parameterizing software within its domain of reuse. The “variability factors” become key elements of software architecture and make them robust to change within the domain of reuse. The maturity of these techniques has improved, and the pressure from software schedule is such that it is now mandatory to use reference architecture.

In reference architecture, the software concepts become so systematic that tools can generate the code. Getting away from the pure code reuse, instead, these software concepts are reused, The approach is supported by automatic code generation and configuration (parameterized according to the variability factors).

There is a paradigm shift from building generic software, to building generic software factories, from reusing code to reusing solutions.

For space software, the notion of software factory includes in particular:

- definition of component models, including interface that can be parameterised in order to deploy automatically components on an infrastructure, itself configured automatically by the factory.
- model transformation from functional data to their physical implementation in the system data base. For example, the command “change mode” is transformed into the actual operation telecommand. Or the Electronic Data Sheet indicates how to access to particular sensors or actuators.
- parameterisation of specific “metier” viewpoint, such as the Packet Utilisation Standard
- “continuous build” of software iterations through automatic code regeneration (build) combined with automated testing.
- supporting life cycle (e.g. Agile)

The session will include presentations addressing various aspects of software factories, and will possibly address the maintenance of these huge tool sets through an example of open source solutions.