

Software Systems Division Data Systems Division

Final Presentation Days

**Program
Abstracts
Future Events
Mailing List
Personal Notes**

ESA/ESTEC - Einstein
9-10 June 2016

**Timetable
&
Abstracts**

Thursday 9 June 2016

08:00 Registration

08:30 Program overview

08:40 System Functional Simulations in the Concurrent Design Process

Speaker: Mr. Stephan Kranz (Telespazio Vega GmbH)

09:30 IMA SP Kernel Qualification Preparation

Speaker: Mr. Mark Hann (SciSys)

Formal Methods Expert to IMA-SP Kernel Qualification Preparation

Speaker: Dr. Andrew Butterfield (LERO)

10:40 Coffee Break

11:00 WI-SAT Project

Speaker: Mr. Ovidiu Ratiu (CDS)

11:50 Reconfiguration Module for Multimission core Computer

Speaker: Mr. Angel Grau Llovera (Thales Alenia Space)

12:40 Lunch Break

14:00 LeonSVF Characterisation

Speakers: Mr. F. Cordero & Mr. E. Baumstark (Telespazio Vega GmbH)

LeonSVF Multicore prototype for Leon3

Speaker: Mr. Gregory Quéré (Airbus DS Toulouse)

15.10 Parallel Programming Models for Space Systems

Speaker: Mr. Eduardo Quiñones (BSC)

16:00 Coffee Break

16:20 CCSDS MO Services, CCSDS SOIS and SAVOIR for future spacecraft

Speaker: Mr. Peter Mendham (Bright Ascension)

17:10 File Management Services interface standardisation

Speaker: Mr. Arnaud Bourdoux (Spacebel)

Friday 10 June 2016

08:00 Registration

08:50 Program overview

09:00 CCSDS File Delivery Protocol

Speaker: Mr. Valentin Picos (Enea)

09:50 Advanced CCSDS File Delivery Protocol Hardware
IP Core

Speaker: Mr. Mladen Berekovic (TU Braunschweig)

10:40 Coffee Break

11:00 AOCS SPW Test Bench Preparation

Speaker: Mr. Luc Planche (Airbus DS)

11:50 VSEE-CDF Data Model Transfer and Modelling
Guidelines

Speaker: Mr. Todor Stoitsev (Telespazio Vega GmbH)

12:40 Lunch Break

13:40 RTEMS Qualification Extensions

Speaker: Mr. Helder Silva (Edisoft)

14:30 Porting of MicroPython to LEON Platforms

Speaker: Mr. Damien George (George Robotics Ltd.)

**15:20 Fault Detection, Isolation and Recovery Schemes for
Spaceborne Reconfigurable FPGA-Based Systems**

Speaker: Mr. Felix Siegle (University of Leicester)

System Functional Simulations in the Concurrent Design Process

Speaker: Mr. Stephan Kranz (Telespazio Vega GmbH)

ESA TO: Peter van der Plas (Software System Division)

The study System Functional Simulations in the Concurrent Design Process comprises the design and prototyping of a simulation workbench providing support for rapid definition of System Concept Simulators (SCS), their execution, and the analysis of the simulation results in close link with the Concurrent Engineering (CE) space system databases. The SCS Workbench is meant to support the Concurrent Design Facility (CDF) team, Mission Study teams and Spacecraft Project teams, covering activities in phase O/A, and phase B, while serving as a basis for more elaborate and extensive simulators to be developed in later phases. From interviews with stakeholders a set of specific Use Cases has been identified in which the SCS would bring a high added value to the existing design process from a user perspective. Since the foreseen usage of the SCS in the CDF context provides the most demanding constraints on criteria such as timeliness, flexibility, mixture of fidelity levels and interoperability, this context has been the major focus of the Use Cases.

The main functional components provided by the SCS Workbench are:

- A Simulation Model Development Environment;
- A Simulation Model Importer;
- A Simulation Preparation Environment;
- A Scriptable Simulation Execution Environment;
- A Simulation Recorder/Replayer combination;
- Visualisation tools like alphanumeric displays, plots, surface maps and 3D animations.

The SCS Workbench is implemented as a Java based Eclipse rich client application using state-of-the-art UI technology provided by the Sirius Eclipse project [1] leveraging the Eclipse Modeling technologies, including the Eclipse Modeling Framework (EMF)

and the Graphical Modeling Framework (GMF). Although not directly exposed to the user, the simulation related parts and artefacts follow the SMP2 standard [2]. The tool chain underlying the SCS Workbench makes reuse of a number of existing tools like ESAs Universal Modelling Framework (UMF) [3], and the SimSat simulation runtime [4].

In an iterative implementation and verification approach, several testing series have been performed based on the specified Use Cases. Within each Use Case, a number of testing steps, linked to the direct operation of the SCS Workbench, have been performed. Multiple test rounds following the implementation sprints allowed direct feedback on design and implementation, and verification of the required capabilities for the SCS.

[1] Sirius Eclipse project <http://www.eclipse.org/sirius/index.html>

[2] SMP2 Standard, portal.vega.de/smp

[3] UMF – A Productive SMP2 Modelling and Development Tool Chain, Ellsiepen, Fritzen, Reggestad, Walsh, SESP 2012

[4] SIMSAT as part of SIMULUS

http://www.esa.int/Our_Activities/Operations/Ground_Systems_Engineering/SIMULUS

IMA SP Kernel Qualification Preparation + Formal Methods Expert to IMA-SP Kernel Qualification Preparation

Speakers: Mr. Mark Hann (SciSys) – Dr. Andrew Butterfield (LERO)

ESA TO: Ms. Maria Hernek (Software Systems Division)

The purpose of this study was to prepare for future IMA Separation Kernel qualification, taking into account both the development process and the product.

The first phase of the preparation was to consolidate a requirements baseline for IMA Separation Kernels (for a single core and multicore processors).

Next, a conformance assessment of pre-selected candidate kernels, against the baselined IMA Separation Kernel requirements (including ECSS E40 and Q80 compliance) was performed. In

addition an estimate of effort required for complete conformance was provided by the kernel suppliers.

When the baseline had been established the overall qualification strategy and a detailed test plan for qualification of IMA Separation Kernels were defined. An identification and analysis of the suitable methods and techniques for verification and validation of the IMA Separation Kernel was performed which included Model Based Techniques, Formal Method techniques and classical techniques.

The final phase of the study was to perform an evaluation of the adequacy and appropriateness of the selected verification and validation methods and techniques. This included a definition of the test environment and experimentation on the methods and techniques given in the test plan.

At the culmination of this study a complete set of documentation and test artefacts has been prepared for the start of the qualification of a selected IMA Separation Kernel for use in flight software.

WI-SAT Project

Speaker: Mr. Ovidiu Ratiu (CDS)

ESA TO: Mr. Nickolaos Panagiotopoulos (Data Systems Division)

Traditionally, wired-data handling systems are used for the data distribution in satellites. These systems are based on extended standard buses such as the Mil-1553b, CAN, SPI, I2C... or point-to-point connections such as RS422, SpaceWire, etc. Furthermore, the majority of these communication standards are based on redundant cables to increase the reliability increasing the complexity and mass of the satellite harness. These elements (harnessing) have a clear impact on the spacecraft dry mass and the spacecraft assembly and testing. Statistics shows that the increment of the dry mass of spacecraft due to harnessing is in the order of a 10%. Furthermore the wired data-handling systems have other associated problems such as:

- Requiring complex assembly (communication paths), integration and testing as spacecraft complexity increase.
- Signal leakage and isolation for avoiding electromagnetic compatibility issues (EMC)
- Restriction in physical dimensions
- High cost of late design changes
- Possible failures of wires and connectors, risk of system malfunctioning due to EMI and risk of total failure due to any short circuit
- In case of failure it is very difficult to isolate the faulty system
- Overhead for tasks associated to wires allocation and assembly, shields, connectors, brackets, fasteners, supported structure, etc.
- In this context, the use of wireless communication on-board satellites for replacing, complementing or extending standard data communication systems can potentially solve the majority of the above problems, reducing the overall dry mass and the effort for assembly and testing tasks and enhancing the design flexibility and late redesign tasks.

Use of wireless communication buses leads to a significant reduction of mass, cost and AIV effort. Wireless communications should be based on wireless micro-sensors combining low power micro controller, transceiver and power supply with one or more attached equipment (e.g. sensor or actuators). For the space industry, the wireless communication buses will benefit the final performance of the on-board data handling system and will offer a number of specific advantages on the development, assembly, integration and testing activities:

- It simplifies development phases, reducing the development costs.
- It allows modular development independently of the final harness necessities.

- It allows cost reductions exploiting the flexibility offered by the wireless systems, e.g. inclusion of late design modifications, more flexible placement of the equipment...
- It simplifies the EGSE facilities, since data monitoring can be also based on wireless systems (it is not needed to physically attach the monitoring equipment to the on-board avionic).

The objective of the project was to investigate the feasibility of a robust spacecraft communication bus system with reduced or no harness and improving the intra-spacecraft interfacing flexibility without impact in power consumption and minimizing the overall mass. In order to achieve this objective, the project investigated the fulfilment of detailed requirements by a breadboard system composed of 4 wireless nodes running a custom implementation of the ISA100 Wireless protocol which is based on a 802.15.4 PHY layer. The project investigated the feasibility of porting the ISA100 Wireless protocol onto a faster PHY layer than 802.15.4 (802.15.4 UWB) and executed the design and implementation of 4 Wireless Nodes and one Wireless Gateway. The end result of the project is an implementation of ISA100 Wireless onto a UWB PHY layer that was tested in laboratory conditions (TRL 4).

Reconfiguration Module for Multimission Core Computer

Speaker: Mr. Angel Grau Llovera (Thales Alenia Space)

ESA TO: Mr. Gianluca Furano (Data Systems Division)

The objectives of these GSTP activity were :

- To design the AURUM FPGA functions, compliant Thales Alenia Space needs for Observation & Science.
- To procure a Bread Board of the TMTCMRM board on which the AURUM FPGA is embedded,
- To verify the newly developed RM functions on the TMTCMRM Bread Board.

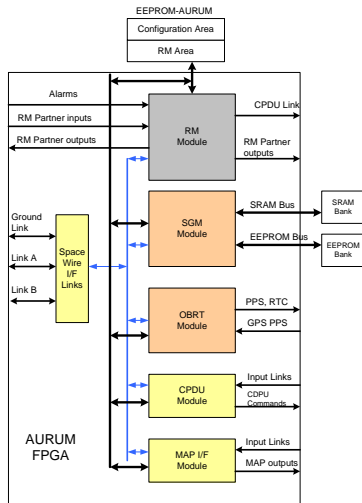
The RM functions as defined on this GSTP allow its re-usability for future CNES & ESA missions to extend its potential applications to Observation & Science applications, for a wide variety of orbits.

The testing was performed by a mixed TAS-F Toulouse and TAS-I Gorgonzola team, using TAS-I Milan test facilities already procured and developed in the frame of the Italian TMTCMRM board design.

The AURUM FPGA, implements the following functions:

- MAP Router to decode and forward the incoming TC segments from an external TC Decoder
- Input MAP I/Fs in order to receive and process “Low Level Commands” and SGM “Memory Load” commands from Ground
- CPDU Arbiter to manage Reconfiguration Commands coming from Local and Partner AURUM as well as HPC from Ground
- Reconfiguration Module for Matched Pair Management, Surveillance and Recovery Functions
- On Board Time Module for timing and synchronization functions
- Safe Guard Memory Module
- Management of Configuration Parameters

- SpaceWire I/F to communicate with Nominal and Redundant Processor Modules



LeonSVF Characterisation Case Study#2

Speaker: Mr. F. Cordero & Mr. E. Baumstark (Telespazio Vega GmbH)

ESA TO: Mr. Mauro Caleno (Software Systems Division)

Airbus Defence & Space (F) developed the Leon Software Validation Facility (LeonSVF) for on-board computers based on Leon2 (AT697) and Leon3 processors – see also separate abstract “LeonSVF Multicore prototype for Leon3”. Telespazio-Vega GmbH (TPZV) was tasked to characterise independently the LeonSVF for single core Leon3 in a flight-representative application, in order to feedback improvements on the system. The characterisation assessed general aspects of usability, ease of integration within system simulators as well as quantitative performance figures.

TPZV used the MASCOT asteroid lander on-board software as flight-representative application which they developed. The MASCOT is a DLR system developed by TPZV and hosted on the Japanese Hayabusa-II satellite currently travelling in the solar system towards the target asteroid Ryugu (formerly 1999JU3), C type asteroid and planned to deliver the MASCOT lander in 2018/2019. For the LeonSVF characterisation, TPZV replaced the Leon3 software emulator (TSIM) in their pre-existing Software Development and Validation Facility (SDVF) with the Leon Emulation Board (LEB) using the TSIM-adaptation layer from the LeonSVF. TPZV then executed a number of pre-existing operational scenario tests and compared the performance of the LEB-based SDVF with the TSIM-based SDVF.

Findings and enhancements identified during the work were implemented in the LeonSVF product prior to the characterisation or later in the CCN3 which is presented in the session “LeonSVF Multicore prototype for Leon3”. The speed factor of the SDVF while simulating the 40MHz MASCOT Leon3 processor varied between 1.0 and 0.5 at high loads, i.e. all slower than real time, both on the TSIM and LEB depending on the relative CPU and I/O loads. At higher CPU loads the LEB-based SDVF performs better than the software simulation (1.0 versus 0.5), while at higher I/O loads the

overall SDVF performance reverses. In normal load cases, the two platforms had similar speed factor around 1.2.

Overall the Leon Emulation Board for single Leon3 core was found to be a good product with similar performance compared to the software simulators.

LeonSVF Multicore prototype for Leon3

Speaker: Mr. Gregory Quéré (Airbus DS Toulouse)

ESA TO: Mr. Mauro Caleno (Software Systems Division)

The key component of the LeonSVF is the Leon Emulation Board (LEB), a reusable building block for Software Validation Facilities and System Simulators of satellites based on Leon2 and Leon3 Systems on Chip (SoC). The current CCN#3 of the project “OBC Simulator Architectures and Interfaces to System Test Benches” has produced a multicore configuration of the LEB with 2xLeon3 + 2xGRFPU.

The multicore LEB provides a combined level of accuracy and performance that cannot be provided anymore by any kind of software simulations (instruction set or block-oriented simulators) thanks to the execution of the VHDL of the Leon IP Core and the other IP Cores of the target System on Chip.

The LEB is a PCIe board with an FPGA which executes the actual Leon VHDL code thus enabling cycle-accurate execution of the Leon3 dual core at 100MHz minus simulated I/O load overheads. The LEB is designed for Linux workstations and is controlled via an API compatible with TSIM by Cobham-Gaisler. The key mechanism of the LEB is that whenever simulation events trigger from any Leon IP core (I/O or AMBA access, timed events etc), the clock to the whole SoC is suspended to keep time representativity for the on-board software (OSW) while the simulations execute. The synchronisation between the cores is implicit in the VHDL

execution, whereas a software-based CPU emulation would either have a level of inaccuracy or a performance penalty, like also MMU, caches and FPU simulation.

At present 4 configurations of the LEB exist: 1) for the Atmel AT697; 2) for a generic Leon3; 3) for a Leon3 SoC inspired from the SCOC3 SoC by Airbus DS and which embeds the Spacewire IP Cores; 4) dual core SoC with 2xLeon3 + 2xGRFPU and 2 Spacewire IP Cores.

Parallel Programming Models for Space Systems

Speaker: Mr. Eduardo Quiñones (BSC)

ESA TO: Mr. Marcel Verhoef (Software Systems Division)

Exploiting the performance of current multi-core processors (e.g. NGMP) requires the usage of appropriate parallel programming models. Such a requirement is further exacerbated with the advent of next-generation many-core heterogeneous embedded architectures (e.g. Kalray MPPA), in which parallel programming becomes mandatory to exploit the massive parallel computation capabilities. OpenMP, the de-facto standard for shared memory architectures in the high performance domain, is increasingly being considered in the embedded domain. Originally focused on massively data-parallel, loop-intensive applications, the latest specification of OpenMP (version 4.5) has evolved to consider a very sophisticated tasking model supporting fine-grained and irregular parallelism. Moreover, it also incorporates new features to couple a main host processor to one or more many-core accelerators, where highly parallel code kernels can be offloaded for improved performance/watt.

Our vision is that OpenMP is an excellent choice for current and future real-time embedded systems for a twofold reason: First, it provides the abstraction level required to program parallel applications, while hiding the complexities of multi- and many-core architectures. Second, it facilitates the migration of real-time

embedded systems from multi- core to many-core platforms. Unfortunately, OpenMP adopts a parallel execution model that differs in many aspects from the real-time execution model: The programming interface and the runtime scheduler are completely agnostic to any timing requirement that the target application may have. Moreover, current implementations of OpenMP are not designed for embedded environments in which the execution is constrained by hardware resources, operating systems (OS) or application requirements.

In the "Parallel Programming Models for Space Systems" project, we evaluated the use of OpenMP in the space domain with a twofold objective. First, to demonstrate the benefits of using OpenMP4 in terms of: performance speed-up, programmability and time analysability. Second, to identify the challenges that future implementations of OpenMP must address, by evaluating the implications of using OpenMP in terms of system resources, i.e. memory footprint, operating system services and size of application memory working set.

CCSDS MO Services, CCSDS SOIS and SAVOIR for Future Spacecraft

Speaker: Mr. Peter Mendham (Bright Ascension)

ESA TO: Mr. Andreas Jung (Software Systems Division)

The CCSDS MO Services, CCSDS SOIS, and SAVOIR for Future Spacecraft (MOSS) activity analysed three key technologies:

- CCSDS Mission Operations Services (MO Services),
- CCSDS Spacecraft On-board Interface Services (SOIS) and
- the SAVOIR-FAIRE On-board Software Reference Architecture (OSRA),

with a view to consolidating them into a single harmonised architecture. As such, the following objectives were met:

- the user needs and requirements leading to the three technologies were elicited, evaluated and consolidated to produce a single, coherent list;
- based on the consolidated requirements, a single consolidated architecture was derived and analysed for its impact on the accepted operability concept for spacecraft;
- an executable prototype system was designed and implemented in order to demonstrate the key elements of the consolidated architecture and the ways in which the constituent technologies interact;
- lessons were drawn from the requirements analysis, architectural consolidation and prototyping experience which should be directed back towards the relevant standards so as to improve the applicability and consistency of the constituent technologies.

The MOSS consortium comprises Bright Ascension Ltd as project prime contractor, reporting directly to ESA. RHEA System SA and OHB System GmbH were sub-contractors reporting to Bright Ascension.

File Management Services interface standardisation

Speaker: Mr. Arnaud Bourdoux (Spacebel)

ESA TO: Mr. Christophe Honvault (Software Systems Division)

Over the current years, mass memories for space missions are evolving from simple storage areas to being central to the overall spacecraft avionic architecture. Missions are no longer using mass memories only for the storage of science and housekeeping data, but also rely on this storage capability to hold mission-critical items such as telecommand timelines, software images, on-board control procedures or recovery configurations. In parallel, the concept of mission operations is also slowly shifting from the packet paradigm to the file paradigm. With the significant quantity of data that needs to be transferred between space and ground, and the

availability of reliable file transfer protocols which can ensure global consistency of the transferred files, the usage of files becomes common on the near-future missions.

In this context, the SAVOIR community has set up a working group dedicated to the establishment of unified functional and interface requirements of the mass memory function: the SAVOIR-MAS AIS group. The FMSIS (for File Management Services Interface Standardization) activity has been conducted under the supervision of this working group. In a first phase, a set of user and system requirements, produced by the SAVOIR-MAS AIS group, were analyzed alongside a survey of mass memories implemented in missions currently in flight or under development. These user interface requirements were updated, and generic use cases, based on the mass memory survey, were defined. Definition of the use cases was supported by an existing AAML toolset improved in the frame of this study.

In a second phase, a generic technical specification for mass memories was derived from the user requirements. A communication protocol, implementing these requirements, was then defined. This new set of requirements, encompassing both 'file-aware' and 'simple' mass memories, was then assessed for impact against the SAVOIR Avionics and On-board Software Reference Architectures. In parallel, a demonstrator was designed and implemented according to the specified mass memory requirements, and new PUS-C services relevant to the mass-memories (in particular, services 6 – Memory Management; 15 – On-board Storage; and 23 – File Management) were deployed on top of a mass memory simulation fulfilling the proposed requirements, in order to verify the adequacy of the technical and communication specifications.

After an iteration based on experience gained deploying the mission use cases on this simulator, the specification documents output by the FMSIS study are submitted to the SAVOIS-MAS AIS group for future standardization.

CCSDS File Delivery Protocol

Speaker: Mr. Valentin Picos (Enea)

ESA TO: Mr. Luca Fossati (Data Systems Division)

The major objective was the implementation of a file transfer protocol which is integrated into an onboard software framework and which sets the basis for a further use or an adaption for future missions. The implementation will contain Class 1 & 2 message transfer and Store and Forward Overlay procedures.

The 2nd objective was to avoid a standalone CFDP protocol implementation but to integrate the new development into an existing onboard software framework. The integration of the CFDP protocol into the development framework will facilitate the development of applications making use of the CFDP protocol as basic services such as event notification, message handling, equipment handling and logging are already available. This was achieved using KARS framework provided by Airbus.

The CFDP implementation was design to support multiple architectures (x86, SPARC) and OSes (RTEMS, PIKEOS and Linux). The programming language used for CFDP implementation is C. The CFDP implementation software consists of three major components: CFDP Core (implements the CFDP class 1 and class 2 and SFO), CFDP User (is using the CFDP User library to exercise the CFDP Core features), UT (this component is transferring the in/out CFDP PDUs of CFDP Core – Ethernet (UDP) and Spacewire are supported). Each component is available as standalone executable application.

We can mention also other challenges that we encounter like: writing unit tests to achieve high code coverage numbers and develop a flexible functional test system to be able to handle complex test setup (x86 pc running Linux and Leon4-N2X board running RTEMS).

For the demo we will transfer a file from Leon4-N2X board running RTEMS to a Regular PC x86 running Ubuntu using Class 2 procedures.

Advanced CCSDS File Delivery Protocol Hardware IP Core

Speaker: Mr. Mladen Berekovic (TU Braunschweig)

ESA TO: Mr. Luca Fossati

The CCSDS File Delivery Protocol (CFDP) standard was released to extend the CCSDS protocol stacks for space communication on higher layers, enabling reliable file transfer and remote file system management over interplanetary distances. It features delay tolerant reliable and un-reliable file delivery functions, management services and advanced configuration that can reduce operations cost and risk in future space missions. In this ESA study we developed a VHDL IP Core implementing the full CFDP protocol features. The IP Core implements uplink and downlink functionalities, includes SEE mitigation measures and can be used in conjunction with space-grade Leon2FT and Leon3FT processors.

It is using a high-speed AHB interface configuration that was evaluated in a Design Space Exploration in course of the project, resulting in an optimized version of the IP which supports streaming interfaces with low latency and a maximum throughput of 400MB/s.

AOCS SpaceWire Test Bench Preparation

Speaker: Mr. Luc Planche (Airbus DS)

ESA TO: Ms. Elena Tremolizzo (Control Systems Division)

SpaceWire is a well-established communication network standard for Space systems, defined by a set of ECSS standards. Current implementations mostly concentrate on payload systems as SpaceWire is one of the few medium rate solutions available (from 2 to few hundred Mbps, to be compared to the ~1 Mbps needed by a spacecraft avionics).

The “AOCS SPW Test Bench Preparation” study aims at confronting the SpaceWire technology with the Attitude and Orbit Control Systems needs and architectures, in view of complementing a preliminary assessment performed by the SAVOIR1-SAIF2 working group. In that respect it intends to support the AOCS engineers in understanding the implications of implementing SpaceWire at sensors, actuators and system level both in term of impacts and opportunities. Some functions within the AOCS may indeed benefit from implementing SpaceWire interfaces on board for communication performance reasons like e.g., for connecting a camera unit that will be used for relative navigation, or for integrating the instrument in the AOCS closed loop (like on Gaia for example).

The study combines an assessment of how SpaceWire technology may respond to the AOCS needs w.r.t. the on-board communication layer, with an experimentation of a SpaceWire-based AOCS, featuring representative AOCS functions and architecture as well as a representative SpaceWire-based communication layer. It addressed in particular a few basic questions:

- Does SpaceWire fit the requirements of an AOCS?
- Which opportunities at AOCS level are opened by the use of SpaceWire technology?
- Which SpaceWire development is required to support the AOCS roadmap?

The study has confronted the AOCS needs with the SpaceWire technology and roadmap, and selected a reference architecture for investigating more in depth and confirming some of the key points resulting from the analysis. The outcome is that SpaceWire technology could be used as communication backbone without requiring a modification, at functional level, of the AOCS controller or the sensors/actuators. However, the decision to use SpaceWire as main AOCS communication backbone would be a more complex one. Opportunities opened by this case have been investigated that could contribute to such decision.

VSEE-CDF Data Model Transfer and Modelling Guidelines

Speaker: Mr. Todor Stoitsev (Telespazio Vega GmbH)

ESA TO: Mr. Christian Philippe (Software Systems Division)

The ESA Concurrent Design Facility (CDF) is using the Open Concurrent Design Tool (OCDT) for conceptual space system definition in the early mission phases O and A. OCDT provides an open source environment for concurrent, collaborative and distributed engineering, which is based on the ECSS-E-TM-E-10-25A meta-model. Model-Based Systems Engineering (MBSE) for space systems in later mission phases (B, C, D, E) is addressed through the Virtual Spacecraft Engineering Environment (VSEE). The VSEE framework is based on the ECSS-E-TM-10-23A meta-model. One of the main tools of the VSEE is the Space Systems Design Editor (SSDE) which is used for creating and editing space system models. In the future it is envisaged to merge both ECSS meta-models into a single standard that can be used across all space system life-cycle phases (i.e. Phases O to F). This activity addresses the envisaged integration on meta-model as well as on tooling level. As part of the activity, the compatibility between the two meta-models has been analyzed and a set of required model transformations from 10-25 to 10-23 and vice versa have been defined. Further, an integration between the SSDE and OCDT tooling has been implemented and validated with reference data sets to demonstrate the integration approach. The implemented integration enables model exchange between the OCDT and VSEE environments, fostering reuse of system engineering models elaborated in different mission phases. Finally, a set of system technical budget reports have been developed and associated modelling guidelines for producing the budgets have been defined.

RTEMS Qualification Extensions

Speaker: Mr. Helder Silva (Edisoft)

ESA TO: Mr. Christophe Honvault (Software Systems Division)

The RTEMS Qualification Extensions objectives were to identify the users' needs for communication protocols interfaces and services, to analyse the recent RTEMS versions for management of communications interfaces and monitoring of kernel events. It was implemented in RTEMS by EDISOFT the support drivers for Spacewire and MIL-STD-1553 interfaces for a typical LEON3 processor (GR712RC) and implemented the support for kernel and application monitoring services. Both implementations were provided with the correspondent qualification package for the RTEMS version, including the testsuite covering 100% statement and decision coverage, as requested by the Galileo Software Standards for the Development Assurance Level B. The implementation is provided pre-qualified for the GR712RC board using a single processor implementation of RTEMS.

The RTEMS by EDISOFT was updated to include the test campaign for the new and modified requirements, all tests were executed successfully, 100% of source code statement and decision coverage was achieved and RTEMS Tailored new performance information (RTEMS directives, critical sections, interrupt timing, CPU occupancy and memory usage) was collected and reported in the RTEMS Improvement Software Budget Report.

Porting of MicroPython to LEON platforms

Speakers: Mr. Damien George (George Robotics Ltd.)

ESA TO: Mr. David Sanchez de la Llana (Software Systems Division)

MicroPython is “a lean and fast implementation of the [Python3](#) programming language that is optimized to run on a microcontroller (ARM)”. It was developed from scratch by Damien George (using crow-funding) and it is available under the MIT Open Source license.

MicroPython has been ported to LEON on top of RTEMS 4.8. The following aspects have been taken into account for the porting: optimization of resources, deterministic and bounded use of resources, interface with the Operating System, interface with C code, concurrency and multitasking. Supported Python version is 3.4 and the number of language features is very high (most of the language, classes, exceptions, etc.).

The standard MicroPython test bench plus specific test cases for LEON have been executed successfully. The port has been tested first into a LEON-2 simulator (the ESOC EMU simulator) and after in a real LEON target (AT-697E).

Intended uses and License policy will be commented.

Fault Detection, Isolation and Recovery Schemes for Spaceborne Reconfigurable FPGA-Based Systems

Speaker: Mr. Felix Siegle (University of Leicester)

ESA TO: Mr. Dirk Thurnes (Data Systems Division)

This research contributes to a better understanding of how reconfigurable Field Programmable Gate Array (FPGA) devices can safely be used as part of satellite payload data processing systems that are exposed to the harsh radiation environment in space. Despite a growing number of publications about low-level mitigation techniques, only few studies are concerned with high-level Fault Detection, Isolation and Recovery (FDIR) methods, which are applied to FPGAs in a similar way as they are applied to other systems on board spacecraft. This PhD thesis contains several original contributions to knowledge in this field. First, a novel Distributed Failure Detection method is proposed, which applies FDIR techniques to multi-FPGA systems by shifting failure detection mechanisms to a higher intercommunication network level. By doing so, the proposed approach scales better than other

approaches with larger and complex systems since data processing hardware blocks, to which FDIR is applied, can easily be distributed over the intercommunication network. Secondly, an innovative Availability Analysis method is proposed that allows a comparison of these FDIR techniques in terms of their reliability performance. Furthermore, it can be used to predict the reliability of a specific hardware block in a particular radiation environment. Finally, the proposed methods were implemented as part of a proof of concept system: On the one hand, this system enabled a fair comparison of different FDIR configurations in terms of power, area and performance overhead. On the other hand, the proposed methods were all successfully validated by conducting an accelerated proton irradiation test campaign, in which parts of this system were exposed to the proton beam while the proof of concept application was actively running.

Future Events

10th ESA Workshop on Avionics, Data, Control and Software Systems (ADCSS 2016)

18-20 October 2016

ESA/ESTEC – Noordwijk (The Netherlands)

Website: adcss.esa.int

7th International SpaceWire Conference – ISC2016

25-27 October 2016

Yokohama, Japan

Website: 2016.spacewire-conference.org

Software Systems Division & Data Systems Division Final Presentation Days

6-7 December 2016

ESA/ESTEC – Noordwijk (The Netherlands)

Workshop on Simulation and EGSE for Space Programmes – SESP2017

28-30 March 2017

ESA/ESTEC – Noordwijk (The Netherlands)

Mailing List

If you would like to stay up to date of Future Events, please register via: lists.estec.esa.int/lists/

