ESROCOS: A ROBOTIC OPERATING SYSTEM FOR SPACE AND TERRESTRIAL APPLICATIONS

M. Muñoz Arancón [(1)], G. Montano [(2)], M. Wirkus [(3)], K. Hoeflinger [(4)], D. Silveira [(5)], N. Tsiogkas [(6)], J. Hugues [(7)], H. Bruyninckx [(8)], I. Dragomir [(9)], A. Muhammad [(10)]

[(1)] GMV Aerospace and Defence, Isaac Newton 11 PTM, Tres Cantos, 28760 Madrid, Spain, mmunoz@gmv.com
[(2)] Airbus Defence and Space Ltd, Gunnels Wood Road, SG1 2AS Stevenage, United Kingdom, Giuseppe.Montano@Airbus.com
[(3)] Deutsches Forschungszentrum für Künstliche Intelligenz GmbH - Robotics Innovation Center, Robert-Hooke-Straße 1, 28539 Bremen, Germany, malte.wirkus@dfki.de
[(4)] Deutsches Zentrum für Luft - und Raumfahrt Ev, Linder Höhe, 51147 Köln, Germany, Kilian.Hoeflinger@dlr.de
[(5)] GMVIS Skysoft SA, Av. D.Joao II Lote 1.17.02, Torre Fernao Magalhaes 7°, 1998025 Lisboa, Portugal, Daniel daniel.silveira@gmv.com
[(6)] Intermodalics BVBA, Gaston Geenslaan 9, 3001 Leuven, Belgium, Nikolaos Tsiogkas nikolaos.tsiogkas@intermodalics.eu
[(7)] Institut Superieur de l'Aeronautique et de l'Espace, Avenue Edouard Belin 10, 31055 Toulouse, France, Jerome.HUGUES@isae-supaero.fr
[(8)] Katholieke Universiteit Leuven, Oude Markt 13, 3000 Leuven, Belgium, Herman.Bruyninckx@kuleuven.be
[(9)] Universite Grenoble Alpes, 700 Avenue Centrale, 38401 Saint Martin d'Heres, France, iulia.dragomir@univ-grenoble-alpes.fr
[(10)] VTT Technical Research Centre of Finland Ltd., Tekniikankatu 1, 33720 Tampere, Finland, ali.muhammad@vtt.fi

ABSTRACT

ESROCOS (http://www.h2020-esrocos.eu) is a European Project in the frame of the PERASPERA SRC, (http://www.h2020-peraspera.eu/), targeting the design of a Robot Control Operating Software (RCOS) for space robotics applications. ESROCOS goal is to provide an open-source framework to assist in the development of flight software for space robots, providing adequate features and performance with space-grade Reliability, Availability, Maintainability and Safety (RAMS) properties. This paper presents ESROCOS and summarizes the approach.

1. INTRODUCTION

In the industrial robotics domain, it is common practice for robotics manufacturers to adapt proprietary Real-Time Operating Systems (RTOS) regarding specific functional and business demands. This results in the development of non-standardized, proprietary solutions. The space robotics industry has been following a similar path, with the additional constraints imposed by the operating environment and the particularities of each mission, leading to minimal reuse across developments.

On the other hand, open-source robotics frameworks such as ROS [1], ROCK [2] or GenoM [3] have flourished in the academic world, enabling the growth of reusable functional block libraries and allowing for faster application development.

ESROCOS will provide an open-source robotics framework designed from the beginning to support the development of space robotics. ESROCOS will provide the foundation of a robotics software development eco-system not only relevant for the space industry, but also fulfilling the requirements of other industries like underwater, nuclear or medical robotics.

The paper is structured as follows: § 2 enumerates the specific objectives and challenges, identified for the project; § 3 describes the ESROCOS framework; and § **Error! Reference source not found.** informs on project updates.

2. PROJECT OBJECTIVES AND CHALLENGES

In order to meet the goal of providing to the robotics community an open-source framework that is the base of future space robotics applications, the ESROCOS project has defined several objectives:

1. **Develop a space-oriented RCOS:** the development of ESROCOS follows the ECSS standards [4,5,6] and hardware support (LEON processor) on space RTOS (RTEMS [7]).

2. **Integrate advanced modelling technologies:** ESROCOS will include a complete model-based methodology, including robotic-specific modelling semantics, and supporting the design and integration of software components, as well as the verification of the structural and behavioural properties at system level. By relying on formal verification and automatic code generation, this methodology will help to

reduce the number of defects that are introduced in the software.

3. **Focus on the space robotics community:** actors that have a leading role in state-of-the-art robotics missions have participated in the definition and review of the ESROCOS requirements, ensuring that they are aligned with the needs of current and future missions.

4. **Allow for the integration of complex robotics applications:** ESROCOS will support mixed-criticality applications using time and space partitioning, which allows running applications with different levels of quality on the same on-board computer, ensuring no propagation of failures among them.

5. **Avoid vendor lock-in:** ESROCOS will integrate existing open-source tools as well as new developments, and will be distributed as open-source software to the robotics community.

6. **Leverage existing assets:** instead of attempting to develop a new framework from scratch, ESROCOS builds on existing technologies such as the TASTE framework [8] developed and maintained by ESA and partners. It will incorporate tools, libraries and approaches from well-established robotics software ecosystems such as ROCK and ROS.

7. **Cross-pollinate with non-space solutions and applications:** the design of ESROCOS will benefit from the experience gathered from the nuclear robotics domain, with very stringent RAMS requirements.

The development of ESROCOS is not an isolated activity. The PERASPERA SRC has launched six Operational Grants (OG), where each OG targets different aspects of space robotics (software, [...]). The OG's run in parallel and and are commonly coordinated what is essential to ensure mutual success.

The following section presents the ESROCOS framework, as well as the approach selected to overcome these challenges and accomplish the project objectives.

3. DESCRIPTION OF THE FRAMEWORK

3.1 Scope and workflow

ESROCOS is a framework for developing robot control software applications. It includes a set of tools that support different aspects of the development process, from architectural design to deployment and validation. In addition, it provides a set of core functions that are often used in robotics or space applications.

The ESROCOS framework supports the development of software following the ECSS standards. It does not by itself cover all the development phases and verification steps, but it facilitates certain activities and ensures that the software built can be made compatible with the RAMS requirements of critical systems.

The starting point of the workflow is the formal modelling of the robot and the application. The model-based approach facilitates the early verification of the system properties, in particular for RAMS. The modelling activities encompass the following aspects:

- The robot’s kinematics chain, in order to produce a formal model of the robot motion, from which software can be automatically generated.
- The hardware and software architecture of the application, including non-functional properties (real-time behaviour, resource utilisation, etc.).

The models allow for different analyses to verify the non-functional properties of the system and iteratively refine the system architecture. ESROCOS relies on both existing and newly developed tools to support the different modelling aspects.

The model of the application may include functional building blocks, either provided by ESROCOS or specifically generated from the models (e.g. a hybrid dynamics instantaneous motion solver). This model can then be used to automatically generate the software scaffolding for the application, consisting of the skeleton of the application components and the glue code that enables the inter-component communication. The application-specific behaviour is implemented and integrated in this structure, making use of libraries to support the required functionalities.

The application binaries can then be built and deployed in a runtime platform. Distributed applications are possible, with components running in separate nodes or partitions.

ESROCOS can be used to model applications using time and space partitioning (TSP), in order to build mixed-criticality systems in which components with different RAMS levels can safely coexist. These applications can be deployed on a SPARC (LEON) platform using the AIR hypervisor.

The communication between the application components at runtime is enabled by the PolyORB-HI middleware. ESROCOS will provide also bridge components that enable the communication with external middleware for ROS and ROCK. This will allow the robotics engineer to use tools from these ecosystems (data visualizers, simulators, etc.) for testing and debugging the application. A selection of tools will be provided ready to use with ESROCOS, with all the required data types and interfaces. In addition, middleware bridges will allow the user to integrate existing software assets and run them together with newly built software in a distributed environment.

The following sections explain in detail the main elements of the ESROCOS framework.

3.2 Robot and software modelling tools

Kinematic chain models. ESROCOS includes tools to formally model the kinematics and dynamics of ideal, lumped-parameter robots of all possible configurations (serial, mobile, parallel, hybrid, multi-DOF joints, multi-articular actuation), with a structured set of interdependent modelling languages: geometry (e.g., line, point, pose), kinematics (e.g., joint, link, inverse dynamics, Jacobian, singularity), mathematical representations (e.g., frame, quaternion), numerical representations (e.g., homogeneous transformation matrix, n-vector), digital representations (e.g., 16-bit integers, IEEE floats), and physical dimensions (e.g., length, meter, energy, Joule). For all properties and transformations that are physically relevant for robots (e.g., forward kinematics, hybrid dynamics), code-generators will be provided, that take a specification in a formal modelling language as input, and generate code with verifiable properties (e.g., no dynamic allocation).

Distributed real-time system models. ESROCOS relies on the TASTE framework to model robotics applications from a real-time software perspective. TASTE is an open source framework that allows the development of embedded, real-time systems. It relies on technologies such as standardized modelling languages (e.g., ASN.1 [9] and AADL [10]), code generators and real-time systems, and allows for the generation of application skeletons and the production of the system executable. The designers implement their embedded systems using a set of views, abstracting the user from the implementation details of the underlying platform (e.g., operating system, drivers) and guaranteeing the fulfilment of real-time properties.

Model analysis and verification. The TASTE framework supports the analysis of the real-time behaviour and resource utilisation of the software. ESROCOS will complement these capabilities with BIP (Behaviour, Interaction, Priority) [11] formal models, which offer additional possibilities to analyse the software and verify properties at a behavioural level.

The verification and validation of TASTE models is done with the BIP framework via a model translation between the two formalisms. The BIP framework offers several analysis tools: iFinder [12] verifies the satisfaction of safety properties, SMC [13] evaluates a system's performance metrics, and user-guided/automated simulation validates the given requirements. The model translator, the simulator and the SMC are being developed/expanded in the ESROCOS framework to consider more complex systems with hard real-time constraints, such as robot controllers.

3.3 Runtime platforms

The ESROCOS framework supports the development of applications for three target quality levels: laboratory, high-reliability and space. Laboratory applications focus on reduced development times with light quality assurance activities. Space quality applications have demanding RAMS requirements and must follow a strict development process.

ESROCOS uses the TASTE toolset, which supports different hardware and software platforms. For laboratory applications, ESROCOS targets x86/Linux systems and provides a set of tools for logging and data visualization, among others, to facilitate the development and debugging of applications. For space-quality applications, the framework targets SPARC/RTEMS systems and includes formal modelling tools that enable correct-by-construction software development and verification of RAMS properties.

Space robotics applications are complex and they may combine functions with different degrees of criticality and real-time requirements. For instance, the functional layer may rely on hard real-time control loops, while higher-level functions may require a varying amount of time and memory to complete an operation. In order to safely support such diverging requirements, the ESROCOS framework offers the capability to design time- and space-partitioned systems using TASTE and the AIR hypervisor of GMV Portugal.

The TSP concept, also known as Integrated Modular Avionics, offers the possibility of integrating multiple functions into partitions of the same set of physical resources, allowing the aeronautical industry to manage software growth in functionality and in efficiency. Partitioning keeps applications from inadvertently influencing each other by enforcing strict separation, segregating computing resources in space and time.

The ESROCOS framework includes the AIR hypervisor [14], it is an ARINC 653 compliant time and space partitioned operating System that was originally based on RTEMS technology.

3.4 Use Case Demonstration and Validation

The ESROCOS framework will be validated according to the two reference scenarios defined by the PERASPERA SRC as representatives of future space robotics missions: a planetary exploration rover mission and an in-orbit assembly mission. The ESROCOS application will be deployed on a space-representative on-board computer. For the orbital scenario, the application will control a robotic arm and a camera, perform Cartesian and joint space real-time motion control, and acquire and display telemetry. For the planetary scenario, the application will drive a rover platform in Ackermann and point-turn modes, acquiring images and platform telemetry during the traverse.

In addition, in order to demonstrate the benefits of the ESROCOS framework for other critical robotics applications beyond space systems, ESROCOS will be validated in a nuclear robotics scenario. For the validation in nuclear scenario the DTP2 platform of ITER [15] is used.

4. FOLLOW UP

Updates on the activity can be found at the project's website (http://www.h2020-esrocos.eu). The ESROCOS consortium plans to make the framework available through GitHub (https://github.com/ESROCOS), once the software reaches a sufficient level of maturity for widespread usage.

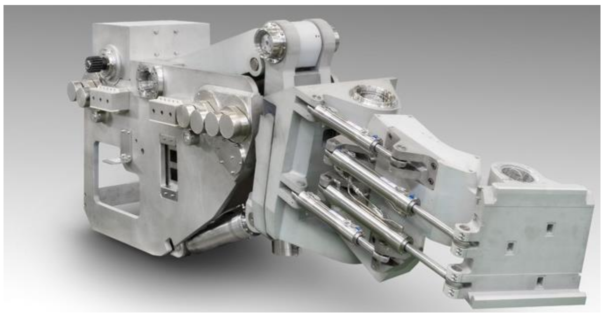

Figure 1. Five-degrees-of-freedom CMM robot at the DTP2 facility at VTT Tampere (bottom)

5. REFERENCES

1. The Robot Operating System. Online at http://www.ros.org
2. Robot Construction Kit. Online at http://rock-robotics.org
3. Ceballos (2011), A., et al. GenoM as a robotics framework for planetary rover surface operations. ASTRA, 2011, p. 12-14.
4. European Cooperation for Space Standardization (ECSS), Space Engineering: Software. ECSS-E-ST-40C, 6-Mar-2009
5. European Cooperation for Space Standardization (ECSS), Space Product Assurance: Software Product Assurance. ECSS-Q-ST-80C, 6-Mar-2009
6. European Cooperation for Space Standardization (ECSS), Space Engineering: Telemetry and Telecommand Packet Utilization. ECSS-E-ST-70-41C, 15-Apr-2016
7. RTEMS. Online at: https://www.rtems.org/
8. Perrotin, M., Conquet, E., Dissaux, P., Tsiodras, T., Hugues, J. , The TASTE Toolset: turning human designed heterogeneous systems into computer built homogeneous software. ERTS 2010, Toulouse (2010)
9. International Telecommunications Union, Recommendation X.680-X.693 (08/2015): Information Technology - Abstract Syntax Notation One (ASN.1) & ASN.1 encoding rules.
10. SAE International, Architecture Analysis & Design Language (AADL), AS5506C (18/01/2017).
11. A. Basu, M. Bozga, J. Sifakis (2006), Modeling heterogeneous real-time components in BIP, in SEFM '06: Proceedings of the 4th IEEE Conference on Software Engineering & Formal Methods. Washington, DC, USA: IEEE Computer Society, 2006, pp. 3–12.
12. Ben-Rayana S., Bozga M., Bensalem S., Combaz J. (2016), RTD-Finder: A Tool for Compositional Verification of Real-Time Component-Based Systems. In: Chechik M., Raskin JF. (eds) Tools and Algorithms for the Construction and Analysis of Systems. TACAS 2016. Lecture Notes in Computer Science, vol 9636. Springer, Berlin, Heidelberg
13. Nouri, A., Bensalem, S., Bozga, M. et al. Statistical model checking QoS properties of systems with SBIP. International Journal of Software Tools for Technology Transfer (2015) 17: 171.
14. C. Silva. Integrated Modular Avionics for Space applications: I/O Module. Master's thesis, IST, October 2012.
15. ITER – the way to energy. Online at: http://www.iter.org

ACKNOWLEDGEMENTS

We would like to thank the European Commission / Research Executive Agency and the members of the PERASPERA Programme Support Activity (ESA as coordinator, ASI, CDTI, CNES, DLR and UKSA) for their support and guidance in the ESROCOS activity. The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 730080.