ASTRA 2019

ESROCOS: DEVELOPMENT AND VALIDATION OF A SPACE ROBOTICS FRAMEWORK

Miguel Muñoz Arancón

© GMV, 2019
All rights reserved

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 730080
CONSORTIUM AND PROJECT DATA

- 27 months project, completed in Jan-2019
- 10 partners, 7 countries
- Part of the roadmap of the PERASPERA Space Robotics Strategic Research Cluster
  [www.h2020-peraspera.eu](http://www.h2020-peraspera.eu)
  - 1st phase: ESROCOS and other building blocks
  - Two tracks: on-orbit and planetary mission robotics
  - Aiming at future demonstrator missions

What’s ESROCOS?
- Open-source framework to develop space robotics software
- Defines a model-based development approach and a collaborative development framework
- Provides tools and reusable components
WHY A SPACE ROBOTICS FRAMEWORK

Several envisaged space missions have an important robotics component, e.g.:
- Planetary exploration (e.g., Mars and Moon rovers)
- Orbital servicing and assembly (e.g., robotic arms on ISS)

The space robotics community has identified the need for frameworks that support the developments needed by these missions

Many robotics frameworks exist, however:
- Industrial applications mostly rely on proprietary frameworks and RTOS, specific to the business needs
- Open-source robotics frameworks (e.g., ROS, ROCK...) have not been developed with the level of quality assurance required by space SW

ESROCOS aims to provide an open-source robotics framework
- Conceived to support the development of space robotics
- Enabling the creation of an ecosystem for the development of space robotics applications
- Potentially useful in other industries with similar requirements, such as underwater, nuclear or medical robotics
MAIN CHARACTERISTICS

- ESROCOS is a framework, not an OS (despite its name), providing:
  - Tools for robotics application modelling and development
  - Reusable runtime components

- Models to enable the production of correct SW
  - SW architecture models: TASTE framework, BIP tools
  - Robot kinematics models: kin-gen

- Common functions
  - Robotics: data types, transforms and data streams, data visualization, logging and replay...
  - Space: TM/TC services (PUS), On-Board Control Procedures (OBCP)

- Development features
  - Interaction with existing robotics frameworks (ROS, ROCK) and tools (Gazebo, RVIZ)
  - Collaborative development: combine core, ecosystem and user’s packages using ROCK’s autoprod

- Time and Space Partitioning: hypervisor for safely integrating mixed-criticality SW

- Target environments
  - Lab quality: x86/linux
  - Space quality: sparc/rtems (GR740 board with LEON4 processor, SpaceWire and CAN)
ARCHITECTURE OVERVIEW

Robot Modelling (Kin-Gen)  Real-Time SW Modelling and Analysis (TASTE & BIP)

- ASN.1 data types
- Robotics libraries
- Simulation, logging & visualization
- TM/TC services

Middleware (TASTE – PolyORB-HI/C)  Time & Space Partitioning (AIR hypervisor)

- Laboratory platform (x86-Linux)
- Space-quality platform (LEON4-RTEMS/AIR)
The robotics engineer must often compose heterogeneous HW/SW components with different needs and capabilities (predictability, performance, criticality...) and requires specialized tools (simulation, visualization, logging...)

ESROCOS provides a heterogeneous set of tools and components to address these needs

Model-based approach → separates concerns and enables formal verification
- Robot modelling → kin-gen
- SW modelling → TASTE and BIP

The TASTE toolchain provides the backbone to model and connect all components

Collaborative development
- Application divided in packages which can be independently developed
- Common robotics data types provided for component interfacing
- Build and dependency management → autoprov

Aiming to facilitate the creation of an ecosystem around ESROCOS
TASTE: THE ASSERT SET OF TOOLS FOR ENGINEERING

- Toolchain for heterogeneous embedded systems using a model-centric development approach, developed by ESA

- SW modelled in 4 views:
  - Data view: data types and encoding → ASN.1 and ACN
  - Interface view: SW components and interfaces → AADL
  - Deployment view: HW architecture (nodes, buses, partitions) → AADL
  - Concurrency view: real-time properties (XML)

- Code generation (Ocarina) and middleware (PolyORB-HI) handle runtime and communications

- Component functionality can be implemented in different languages: C, C++, Ada, Simulink, VHDL (for supported FPGAs), SDL (state machines)...

- Additional tools
  - Real-time analysis (e.g., Cheddar)
  - GUI and automation for testing
  - Etc.
BIP: BEHAVIOUR, INTERACTION, PRIORITY

- A framework for rigorous model-based system design

- Modelling language and tools:
  - BIP Compiler & Engines: generate C++ code linked to different engines and simulate behaviour
  - iFinder/iChecker: verify requirement (safety, performance, etc.) and invariants in BIP models
  - Statistical Model Checking (SMC-BIP): run a representative number of simulations and statistically check whether a requirement (safety, performance, etc.) is satisfied
  - FDIR tool: generation of embeddable fault detection SW components

- TASTE2BIP: automated translation from TASTE-SDL
KIN-GEN: KINEMATICS MODELLING AND CODE GENERATION

- Model of robot kinematics using an innovative approach based on model composability
  - Complete toolchain for Forward and Inverse kinematics
  - Robot model: kinematic trees and arbitrary frame
  - Semantic checks on the model
  - Definition of custom solvers as queries to the model

- Code generation for custom solvers, ensuring correctness

- URDF import for integration with existing tools
TIME AND SPACE PARTITIONING: AIR HYPERVERVISOR

- TSP allows running applications of different criticality on the same HW
- Space robot control SW combines:
  - Critical real-time SW (control loops)
  - Non-deterministic algorithms (image processing, planning...)
- TSP enables running both together with the assurances required by space SW
- AIR is an ARINC 653 hypervisor using paravirtualisation
  - Runs on LEON processors
  - Supports APEX and RTEMS as guest RTOS
  - Implements ARINC 653 APEX (application executive)
  - Partitions communicate through ports and channels
  - Provides services for I/O and health monitoring
  - Includes configuration and emulation tools
- Updated to RTEMS5 and partially integrated in TASTE
CASE STUDIES

- ESROCOS framework validated by developing applications in 3 reference scenarios:
  - Planetary exploration: Bridget rover (ExoMars engineering model) at Airbus Mars Yard (Stevenage, UK)
  - On-orbit satellite servicing: GMV’s platform-art© facility (Madrid, ES)
  - Nuclear robotics: ITER robotics DTP2 test facility at VTT (Tampere, FI)

- Testing with simple tele-operation applications
  - Test focus was usage of the framework and components
  - Next activities (PERASPERA 2nd call) integrate other Building Blocks (autonomy, sensor fusion...)

![Image of test facility](image-url)
SUPPORT TOOLS AND COMPONENTS

- Robotics data types in ASN.1 (derived from ROCK and ROS)

- Libraries
  - Transformer (geometric transformations)
  - Stream Aligner (data stream synchronization)
  - Data Logger
  - Visualization (vizkit3d)
  - PUS services library with OBCP engine (MicroPython)

- Interoperability with ROS and ROCK
  - Import data type definitions
  - Generate runtime bridges between middleware environments

- External tool integration
  - Eigen
  - OpenCV
  - RVIZ
  - Gazebo
ON-ORBIT SERVICING SCENARIO

Satellite mock-up inspection with UR5 manipulator
ON-ORBIT SERVICING

- Teleoperation of UR-5 arm with camera to simulate inspection of a captured satellite
- 4 test cases:
  - Simulated arm, lab platform (x86/linux)
  - Actual arm and camera, lab platform (x86/linux)
  - Actual arm, space-representative platform (leon4/rtems: GR740, SpaceWire)
  - EtherCAT driver → not performed
- Components tested: TASTE, BIP, kin-gen, visualization, PUS, Gazebo, ROCK&ROS integration
PLANETARY EXPLORATION SCENARIO

A ROBOT CONTROL OPERATING SYSTEM FOR SPACE
Planetary Exploration Validation test: A tele-operation application with image data processing

Bridget Mars rover (video by ADS and DFKI)
PLANEETARY EXPLORATION

* Teleoperation of BRIDGET rover

* 4 test cases
  - Rover teleoperation and control
  - Rover trajectory data logging and replay
  - CAN bus device management
  - Integration of data processing functions

* Lab (x86/linux) and space-representative (leon4/rtems – GR740) platforms

* Components tested: TASTE, BIP, transformer, stream aligner, logger, visualization, CAN, OpenCV
NUCLEAR ROBOTICS

- Teleoperation of CMM manipulator
- 2 test cases
  - Trajectory planning and control using interpolation in Cartesian space
  - Trajectory planning and control using interpolation in joints space
- Lab platform (x86/linux)
- Components tested: TASTE, kin-gen, visualization, Gazebo integration
RESULTS AND CONSTRAINTS

- Satisfactory result: ESROCOS is functional and useful to support robotics SW development using a sound modelling, implementation and verification approach

- Not all technical objectives met, some constraints and limitations:
  - Installation not fully managed by autoproj, some dependencies must be installed manually
  - TASTE toolchain has some limitations for large models or large data messages
  - TASTE driver configuration is difficult
  - Transformation of TASTE to BIP models is only partially automated
  - AIR hypervisor integrated in TASTE incomplete, currently mixing AIR and other nodes is not possible
  - SOEM EtherCAT library could not be ported to the GR740
  - Runtime components provided are not verified to the level required for flight SW

- Despite limitations, the validation was positive and ESROCOS is usable in its current state

- Maturity will increase in 2nd phase of PERASPERA
MOR E INFORMATION

- Project completed in January 2019
- ESROCOS website: general info, publications and project documentation https://www.h2020-esrocos.eu/
- ESROCOS @ GitHub: https://github.com/ESROCOS
- Contact: esrocosweb@gmv.com
- PERASPERA and H2020 Space Robotics SRC: https://www.h2020-peraspera.eu/
- H2020 MOSAR in charge of the maintenance of ESROCOS within the SRC
- The main tools and technologies continue developing outside ESROCOS