MORA-TSP: Multicore implementation of the On-Board Software Reference Architecture with Time and Space Partitioning capability

Daniel Silveira, GMV, daniel.silveira@gmv.com; Andreas Jung, ESA, andreas.jung@esa.int;
Laura Gouveia, GMV, laura.gouveia@gmv.com;Marco Panunzio, Thales Alenia, marco.panunzio@thalesaleniaspace.com

1 Rationale

ESA (European Space Agency) launched OSRA (Onboard Software Reference Architecture) pushing the use of domain specific languages and a model component based approach suited for the development of onboard-software. The outcome was an editor capable to generate software outputs for onboard (OBSW) applications.

2 Objectives

MORA TSP objective is to demonstrate the feasibility and performance evaluation of an end-to-end process, tools and building blocks from application level specification using the component based approach of OSRA as the starting process down to representative implementation running on a LEON4-N2X board using Time and Space Partitioning (TSP) and multi-core.

3 Methods

This end-to-end process is achieved through a toolchain comprised by OSRA with respective editor, TASTE toolchain and AIR toolchain.

The user is able to achieve the mentioned goals through by following the steps of the OSRA-to-TASTE process:
Step A - The user shall design the OBSW using the OSRA core model, guided by the OSRA graphical editor.
Step B - Enrich OSRA core model with OSRA external models to add SMP RTOS and TSP support.
Step C - The user shall configure and build the LibPUS library to be integrated into the Execution Platform.
Step D - The user shall invoke the OSRA-to-TASTE.
Step E - The user shall invoke the TSP configurator, only applicable for the usage of TSP (with SMP).

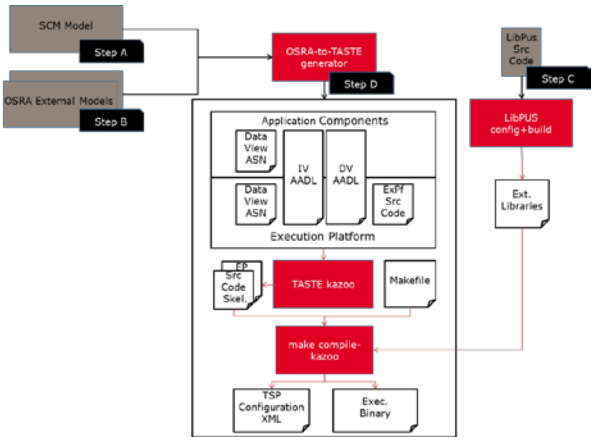

Fig. 1. MORA-TSP Process

4 Results

The end-to-end process was achieved using as use case the EagleEye TSP an ESA's virtual space mission for software testing where the implementation for a multi-core SMP configuration was successful.

In TSP with SMP configuration it was not possible to implement EagleEye because of lack of time to solve software issues. Alternatively it has been deployed a simple TSP with SMP scenario executing IO communication.

5 Conclusions

The upgrade of the OSRA editor and in parallel of TASTE upgraded by its support team made completion of MORA-TSP objectives. This upgrade was able to push both tools to create and architecture, develop and build an OBSW that can fully support the usage of TSP paradigm and at the same time harness to it maximum the processor usage with multi-core.

The demonstration of the EagleEye use case in SMP and a specific TSP example is the proof of the proposed capabilities, the full OBSW is architected in OSRA SCM models and respective external models.