USING HRT-UML AND ADA RAVENSCAR PROFILE IN SENTINEL 3

Round table: Formal Methods in Software Engineering

Position paper

Roberto López roblopez@gmv.es, Ana Isabel Rodríguez airodriigelz@gmv.es

GMV

ABSTRACT

This position paper provides the feedback on the use of Hard Real-Time Unified Modelling Language (HRT-UML) and Ravenscar Profile in the design definition of the Instrument Control Module (ICM) OBSW of Ocean & Land Colour Instrument (OLCI), developed in the context of Sentinel-3 satellite. The ICM software, build around an ERC32 microprocessor, is a critical software (ECCS-E-40B critical level B) responsible of interfacing with the Satellite Management Unit, as well as controlling the rest of instrument units. The aim of this paper is discussing the advantages and disadvantages of using this approach in a real on-board critical software development.

The synergy between technologies, HRT-UML, Ravenscar Profile and Ada95, which in fact has been the main reason for selecting this approach, is the possibility of using Static Verification techniques such as Schedulability Analysis and Model Checking. These techniques allow analysis of a system to be performed at early phases of the development, thus avoiding the common problem of finding only during system integration and testing that the design fails to meet its non-functional requirements.

By complying Ravenscar Computational Model (automatically checked at model level) and estimating the timing requirements of each task and protected object (Period, WCET, Deadline and WCET of each protected method), HRT-UML allows performing assignment of fixed priorities to the different tasks and protected objects according to Ravenscar Profile scheduling model, schedulability analysis based on the previous assignment, and CPU load analysis. WCET can be estimated at first stages of the design and refined in subsequent phases, making possible to check the non-functional timing requirements along the whole development life cycle.

On the other side, some difficulties have been faced when using the proposed approach:

- HRT-UML model restrictions to comply Ravenscar Profile seem to be more restrictive than the profile itself. The constraint that forbids a passive object to use a non-passive object is a HRT-UML constraint not derived from the Ravenscar profile.
- The difference between classes and types is not clearly understood. From UML point of view, an Ada type is exactly the same UML class concept.
- The way the types are managed by HRT-UML tool makes difficult the maintenance of the design in large systems. The same maintainability problem can be observed at code level if the provided code generator is used.
- Toolset does not allow other UML diagrams necessary to complete the design, such as use cases, sequence diagrams, etc. The only supported diagrams are HRT-UML object diagrams.