The European Space Agency's involvement and interest is WCET and scheduling analysis

Extended Abstract

Morten Rytter Nielsen, ESA (morten.nielsen@esa.int)
Eric Conquet, ESA (eric.conquet@esa.int)
Jean-Loup Terraillon (jean-loup.terraillon@esa.int)

Abstract

We consider the use of scheduling analysis as not being a standalone exercise but a system-level activity, congruent with the conscious decision for a 'correctness by construction' development model. We describe how ESA have incorporated the ideas of scheduling analysis into our required standard practices; how we have ensured that the enabling technology is available; and where we see the future of WCET technology and scheduling analysis.

The development approach

The traditional development model, which is used in software space projects under the responsibility of ESA, follows the classical waterfall V-model [ESA-PSS-05]. In this development model the User Requirements are ESA's requirements towards industry and the Software Requirements (or Technical Requirements) are industry's refinement of these. These Software Requirements are then followed by Architectural Design, Detailed Design and coding. On the ascending part of the V-model Unit Tests (verifies Detailed Design), Integration Tests (verifies Architectural Design), System Tests (verifies Software Requirements Definition) and Acceptance Tests (verifies User Requirements) are performed. The testing effort is usually 50-60 percent of the total development effort. Each phase of the development model is finalized with reviews and acceptance together with associated payments.

Historic Space Systems

The V-model has proven its value through many years and projects. Traditionally onboard software-systems have been quite simple and with well separated functional blocks. The utilized software technology centered on fixed cyclic schedulers and dedicated proprietary kernels and very often the I/O mechanism was polling or well characterized interrupts. The required method in the ESA standards for controlling the performance behavior was limited to requirements for CPU utilization at the different stages of development (projects typically used 50% at Architectural Design, 60% at Detailed Design and 70% on final acceptance of the software code). Real-time requirements in the form of reactivity/responsiveness and jitter where either non-existent or at best occasional. The CPU utilization was typically acquired by estimation and later by measurement performed on the final code.

Current Space Systems

The new generation of onboard space systems is significantly richer in functionality and complexity, with much more interaction between functional blocks than traditional onboard systems. Among other reasons, this trend originates from:

- Added throughput (dedicated services)
- More intelligent Autonomy and Failure, Detection, Isolation and Recovery (FDIR) functionality
- Intelligent instruments that sporadically interrupt the main computer
- Added capability of the onboard system in general

Many real-time requirements are now part of the requirement baseline to ensure reactivity and enable different units of the satellite to be developed to lesser tolerances.

Several new problems have surfaced in the new generation of onboard systems [ESA STR-260]. Many of these problems occur in the real-time behavior area. Since CPU utilization is not a sufficient way to ensure real-time behavior, the development approach has to be adapted. ESA have thus sponsored and funded a number of initiatives and supported (and still supports) the introduction of scheduling analysis in the ways outlined in the following paragraphs. For us it is clear that the use of scheduling analysis have major repercussions on the implementing technology as well as on the process standards and associated development approach. This altogether raises a clear demand for better tools support, not limited to the extraction of WCET and the scheduling analysis but also extending to the specification of the real-time attributes and properties of the system.

Standards

The new European generation of space standards, the ECSS standards, allows more flexible development approaches to be used (e.g. spiral models and rapid prototyping) [ECSS-E40B-July2000 and ECSS-E40B-Feb2002]. However, they also require that the used computational model of the system be identified. This explicitly includes the component types (e.g. active-periodic, active-sporadic, protected, passive, actors and process), the assumed scheduling type and model (e.g. fixed priority or dynamic priority) and the accompanying analytical model under which the model is executed (e.g. Rate-Monotonic Scheduling or Deadline-Monotonic Scheduling).

This evolution shows that the previously informally used CPU utilization is now being replaced by much more stringent requirements on the chosen architecture and the rationale behind this choice.

Projects may decide to waive requirements in the standards if this implies too much effort. Thus the enabling technology is very important to lower the entrance to applying scheduling analysis.

Enabling technology

Specification and Design level

In order to be able to really harvest the benefits of the scheduling technology early in the development process, ESA saw the need to accommodate the computational model already at design level. The result of this effort is the HOOD derived HRT-HOOD method [HRT-HOOD]. Currently, TNI (France) and Intecs Sistemi (Italy) have commercial tools supporting this specification and design method.

Implementation technology

ESA have supported the Ada Ravenscar definition from a user perspective. Furthermore we have funded the development of the GNAT/ORK kernel and compilation system and the port of Aonix Raven to the space processor ERC32. Also CNS have an Ada Ravenscar system for the ERC32. Ravenscar compilation systems are now used for Beagle2 and GOCE.

WCET extraction

In some projects the extraction of the WCET profile have been done by hand. However, for scheduling analysis to be used widely and systematically in the space domain, we believe that tools supporting this process are needed. Various ways of acquiring the WCET have been tried, including:

- Instrumentation: Logic analyser (Tektronik) and embedded instrument code (Aonix and VxWorks/Tornado) plus user developed instrument code
- Source level analysis with the support of the compiler: a prototype based on the Adaworld compiler have been developed by Aonix
- Static Analysis on image code: Bound-T from SSF (Finland) have been developed for both the DSP 21020 and the ERC32

Scheduling analysis

Tools to help apply different scheduling analysis techniques have been developed and are now available from Spacebell (Belgium). These tools assist in margin analysis and enables persons less fluent in the logic behind the analysis to interpret and evaluate the results.

Test cases

A standardization of core onboard services has taken place in the form of the Packet Utilization Standard [ESA-PSS-05]. OBOSS [OBOSS] is a reference implementation of selected services, which have been used as a guinea pig for scheduling analysis and the Ada Ravenscar profile. Furthermore, the development approach using scheduling analysis and thereby moving the verification of real-time properties from the typical integration testing phase to the specification and design phase have been applied with great success on the European Robotic Arm (ERA) which is a safety critical module to be used on the International Space Station.

Future of Space Systems

The new draft ECSS standards for onboard space engineering require that scheduling analysis must be performed. Several proposals for new onboard systems are base-lining Ada Ravenscar as the implementation technology and the awareness of scheduling analysis is increasing. Together with standards that require a strong development baseline and a consolidation of the tools assisting in the scheduling analysis in all relevant phases of the development process, the entry barrier for the application of this development approach will be continuously lowered.

ESA continues to fund and promote the development of the enabling technology and the support for the development approach referenced in this paper. The near future evolution is the new space processor LEON, which like the current ERC32 has a Sparc instruction set. The transition to LEON, which has cache, raises new challenges that will require 'expert support' in addressing.

The movement and activities described in this paper has been triggered by problems encountered in space projects using the current development approach. These activities focus on a single computation platform with embedded software. As space onboard systems are moving from synchronous to asynchronous behavior, the need to extend the scheduling analysis to system level is surfacing. ESA is participating in organizations supporting the AADL (Avionic Architecture Description Language) standard. The aim of this work is to define a common language for the design and verification of complex avionic systems. We expect from such a standardization effort the emergence of an open framework that can incorporate various design languages and verification tools able to trap performance and behavioral issues in early design phases.

Conclusion

We have in this extended abstract explained the context and the support of the WCET and scheduling analysis in ESA and the problems that we have encountered which let to this. In the full paper we will include experiences of the different areas outlined above and expand on the future as ESA sees it. This will include specific activities started or foreseen to be started in the area of distributed scheduling analysis.

References:

[ESA-PSS-05] ESA PSS-05-0 Issue 2, February 1991: ESA Software Engineering Standards

[ESA-PSS-07-101] ESA PSS-07-101 issue 1, May 1994: Packet Utilisation Standard

[ECSS-E40B-July2000] ECSS-E-40B Draft 1, 28 July 2000: Space Engineering. Software

[ECSS-E40B-Feb2002] ECSS-E40B Draft 1, 15 February 2002: Space Engineering. Software

[HRT-HOOD] Burns, A. and Wellings, A.: HRT-HOOD: A Structured Design Method for Hard Real-Time Ada Systems, Elsevier, 1995.

[OBOSS] OBOSS home page: http://spd-web.terma.com/Projects/OBOSS/Home_Page

[ESA STR-260] Vardanega, V.: Development of On-Board Embedded Real-Time Systems, ESA STR-260 October 1999