Session: Status and Future of the Ravenscar Profile

Chair: Brian Dobbing
Rapporteur: Juan Antonio de la Puente

1 Scope and aims

Five position papers were input to this session [1, 2, 3, 4, 5], covering such topics as user experience, design patterns, and static analysis. The chair clarified that the session should also consider any final changes to the Ada issues (AIs) related to the Ravenscar profile that could be submitted to the Ada Rapporteur Group (ARG) as part of the current language amendment process. Accordingly, the topics to be covered in the session were defined as follows:

- Current status of the Ravenscar profile:
  - Ravenscar AI's and rationale.
  - Feedback from users' experiences.
- Usage guidelines:
  - Design patterns for the Ravenscar profile.
  - Usage of the profile with static analysis tools.
- Outstanding issues.

2 Current status

The chair summarised the current status of the Ravenscar profile AIs. AI-249 (the profile definition itself) and AI-305 (pragma Detect_Blocking and additional restriction identifiers) have been approved by WG9[1]. This means that the definition is now stable and will most probably be part of the amendment to the Ada standard that is targeted for the end of 2005 (subject to approval by various ISO committees).

A *user guide* for the Ravenscar profile has been written by members of the HRG[2]. The guide is currently available as a technical report of the University of York [6], and has been offered to ISO for its possible adoption as an ISO Technical Report. The guide includes a rationale for the profile, as well as a number of usage examples, design patterns, and other considerations related to the use of the Ravenscar profile in high-integrity systems.

[1] WG9 is the working group on the Ada language of SC22 (programming languages) of the International Standards Organization (ISO).

[2] HRG is the *High-integrity systems Rapporteur Group*, reporting to the ARG *(Ada Rapporteur Group)* on all topics related to high-integrity systems.

3 User experiences

Morten Nielsen commented on the ESA[3] Ravenscar evaluation programme which has been carried out by six European companies, including Deimos Space (Spain), SciSys (UK), Terma (DK), Alcatel (France), and TCP Sistemas e Ingenieria (Spain). The companies evaluated both the Aonix ObjectAda/Raven and the GNAT/ORK compilation systems using programs from previous space software projects.

The results of the evaluation were very positive in general. All the companies concluded that the Ravenscar profile is a good match for typical space applications, and the usefulness of the Ravenscar profile to enable very efficient and small runtime components was acknowledged.

Most of the evaluators had found problems with programming timeouts, which in all cases were reported as a required feature. This problem has been addressed in the Ravenscar profile user's guide [6] via the inclusion of patterns for programming timeouts. Another issue that was reported was the lack for support for exception handling in former versions of the ObjectAda/Raven compiler. Since the definition of the profile is silent about whether exception handling is supported or not, there was general agreement that the Ravenscar profile definition is not affected. A more general version of this comment was that there is a porting issue between Ravenscar profile implementations due to variations in the restrictions that apply to the sequential constructs of the language (on which the profile is silent).

Some other limitations of the compilers have been corrected, or are expected to be corrected in the next few months, as a result of new actions from ESA and the compiler and kernel developers.

The delegates viewed very positively the use of the Ravenscar profile in the Beagle 2 Mars Lander spacecraft and the European Robot Arm (ERA) project, and expressed

[3] European Space Agency.

a write-up of these success stories from ESA to be made publicly available.

Ricardo Maia presented the results of the STADY project, which has used GNAT/ORK as a case study for the software analysis technology developed by Critical Software [2]. The results of the analysis showed a number of faults in the runtime kernel, but the workshop agreed that most of the problems only could happen when the kernel functions are called directly from application code instead of via the use of Ada tasking constructs. Therefore the faults are only an issue if the kernel is used (directly) with non-Ada programs, or if Ada application code directly accesses kernel functions (which is erroneous). Morten Nielsen and Juan Antonio de la Puente commented that the next professional version of GNAT/ORK is expected to put limitations on the kernel visibility from Ada, which will solve these problems for Ada programs.

Marc Richard-Foy reported very positive users' feedback gathered by Aonix on ObjectAda/Raven. The profile appears to be used in aeronautics, space, and ground transportation systems with different levels of criticality. Some users have reported the need for programming timeouts and overrun detection. The workshop agreed that the best way to address overrun detection was via the proposed execution-time clocks package (see AI-307) coupled with a design pattern as proposed in [3]. It was agreed that such a pattern should not be added to the User Guide until after the execution-time clocks package has been approved for addition to the Ada standard. The ability to build a Ravenscar program as a process within an ARINC 653 style of software architecture, and support for multi-language (C and Ada) development are also important issues for Ravenscar users, but outside the scope of the profile definition.

Arnaud Charlet gave a brief account of ACT's activities relating to the Ravenscar profile, including run-time support for the profile on different architectures. The feedback from (potential) users is positive.

4 Design patterns

Tullio Vardanega introduced the discussion on Design Patterns. The specific patterns were initially motivated by a discussion with Niklas Holsti, from Space Systems Finland (SSF), about a possible lack of expressive power when using a scheduling model that was required for schedulability analysis by certain commercially-available tools used within ESA projects [7]. It was made clear in the discussion that this model required a more restrictive subset of Ada tasking than that defined by the Ravenscar profile, with two additional restrictions: every task must have a single suspension point, and every protected entry must have a single task caller (statically, rather than dynamically as in the profile).

Tullio proposed two nontrivial design patterns which addressed these problems [1]. After some discussion the participants agreed that, while the patterns are of interest, they do not address problems that are specific to the Ravenscar profile itself but to a more restricted profile. There was vote on a proposal by Brian Dobbing to put a note in the User Guide [6] to refer to the paper that contains these patterns for assistance when this type of scheduling model is adopted. The proposal was approved with 15 votes in favour and 4 abstentions.

Juan Antonio de la Puente presented UPM's work on using execution-time clocks with the profile [3]. He proposed several patterns for overrun detection and handling using the interface to execution-time clocks which has been proposed in AI-307[4]. The participants agreed that the patterns are indeed useful, but they should not be included in the User Guide until execution-time clocks are included in the amendment to the Ada language standard.

Juan Antonio also identified the need to make some modifications to the interface presented in the execution-time clocks AI, and the workshop agreed on what these modifications should be. The AI must be rewritten to adapt it to this new model.

5 Usage of static analysis tools with Ravenscar programs

Brian Dobbing introduced the discussion with a summary of the Praxis Critical Systems static analysis language RavenSPARK. He pointed out the need to use further restrictions to those in the Ravenscar profile in order to perform the kind of static analysis techniques that are achievable in RavenSPARK. It was agreed that the User Guide should not explicitly list these further restrictions, but Brian took an action to check that the Guide does discuss all of the topics relating to these restrictions in the section on use of static analysis tools.

Alan Burns proposed new SPARK annotations for specifying temporal properties, such as deadlines, which could be integrated with RavenSPARK. These would enable the use of model checking and response time analysis tools to be applied to Ravenscar programs. It was again agreed that this level of detail was out of scope for inclusion in the User Guide, but that Alan should check the Guide to ensure that the discussion on this subject is complete and up-to-date

6 Outstanding issues

There was a discussion on whether non-preemptive scheduling should be considered as an alternative for the

[4] Available at `http://www.ada-auth.org/cgi-bin/cvsweb.cgi/AIs/AI-00307.TXT`.

Ravenscar profile. The issue of non-preemptive scheduling was discussed in the previous IRTAW and it was agreed at that time that preemptive scheduling (as specified by the FIFO_Within_Priorities policy) is an integral part of the profile. The workshop confirmed the previous decision that the profile definition is now closed, and therefore no changes are to be made. Non-preemptive scheduling could be used in the future to define a new profile, using a different identifier from "Ravenscar" in the pragma Profile if this is found to be required.

7 Conclusions

The session adjourned with the following conclusions:

- The lack of direct support for programming timeouts in the profile has been addressed by an existing design pattern in the User Guide.
- The lack of direct support for programming overrun detection in the profile will be addressed if AI-307 is accepted for the amendment to the Ada standard. At this point, the Guide will require updating to add design patterns for overrun detection using this new feature, plus to reflect other changes in the language definition.
- A note should be added to the User Guide that includes a cross-reference to the design patterns in [1] when the scheduling model requires use of a single suspension point per task.
- The interface of execution-time clocks in AI-307 must be revised to reflect the changes agreed at the workshop.
- The section in the Guide that relates to the use of static analysis tools needs to be checked to ensure that it reflects completely and accurately the latest developments such as RavenSPARK and the model checking and response time analysis techniques.
- Confirmation that the FIFO_Within_Priorities and Ceiling_Locking policies are an integral part of the Ravenscar profile definition.

References

[1] Tullio Vardanega, "Ravenscar Design Patterns? Reflections on use of the Ravenscar profile," in *Proceedings of the 12th International Ada Real-Time Workshop (IRTAW12)*, September 2003.

[2] Ricardo Barbosa, Ricardo Maia, and Luís Miguel Pinho, "Verifying, Validating and Monitoring the Open Ravenscar Real Time Kernel," in *Proceedings of the 12th International Ada Real-Time Workshop (IRTAW12)*, September 2003.

[3] Juan A. de la Puente and Juan Zamorano, "Execution-time clocks and Ravenscar kernels," in *Proceedings of the 12th International Ada Real-Time Workshop (IRTAW12)*, September 2003.

[4] Peter Amey and Brian Dobbing, "Static analysis of Ravenscar programs," in *Proceedings of the 12th International Ada Real-Time Workshop (IRTAW12)*, September 2003.

[5] Douglas J. Howe and Stephen Michell, "An approach to formal verification of real time concurrent Ada programs," in *Proceedings of the 12th International Ada Real-Time Workshop (IRTAW12)*, September 2003.

[6] Alan Burns, Brian Dobbing, and Tullio Vardanega, "Guide for the use of the Ada Ravenscar profile in high integrity systems," Tech. Rep. YCS-2003-348, University of York, 2003.

[7] Niklas Holsti and Thomas Langbacka, "Towards static verification of real-time performance: The case of the GOCE platform application software," in *Reliable Software Technologies — Ada-Europe 2003*, Jean-Pierre Rosen and Alfred Strohmeier, Eds. 2003, number 2655 in LNCS, Springer-Verlag.