TECHNICAL
REPORT

ISO/IEC TR 24718

First edition
2005-02-15

Information technology — Programming languages — Guide for the use of the Ada Ravenscar Profile in high integrity systems

Technologies de l'information — Langages de programmation — Guide pour l'usage de «Ada Ravenscar Profile» dans les systèmes de haute intégrité

Reference number
ISO/IEC TR 24718:2005(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

In exceptional circumstances, the joint technical committee may propose the publication of a Technical Report of one of the following types:

— type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;

— type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;

— type 3, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC TR 24718, which is a Technical Report of type 3, was prepared by the University of York for the British Standards Institution (BSI) as guidelines published in 2003, and was adopted (without modifications) by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*.

Introduction

The use of Ada has proven to be of great value within high integrity and real-time applications, albeit via language subsets of deterministic constructs, to ensure full analysability of the code. Such subsets have been defined for Ada 83, but these have excluded tasking on the grounds of its non-determinism and inefficiency. Advances in the area of schedulability analysis currently allow hard deadlines to be checked, even in the presence of a run-time system that enforces preemptive task scheduling based on multiple priorities. This valuable research work has been mapped onto a number of new Ada constructs and rules that have been incorporated into the Real-Time Annex of the Ada language standard. This has opened the way for these tasking constructs to be used in high integrity subsets whilst retaining the core elements of predictability and reliability.

The Ravenscar Profile is a subset of the tasking model, restricted to meet the real-time community requirements for determinism, schedulability analysis and memory-boundedness, as well as being suitable for mapping to a small and efficient run-time system that supports task synchronization and communication, and which could be certifiable to the highest integrity levels. The concurrency model promoted by the Ravenscar Profile is consistent with the use of tools that allow the static properties of programs to be verified. Potential verification techniques include information flow analysis, schedulability analysis, execution-order analysis and model checking. These techniques allow analysis of a system to be performed throughout its development life cycle, thus avoiding the common problem of finding only during system integration and testing that the design fails to meet its non-functional requirements.

The Ravenscar Profile has been designed such that the restricted form of tasking that it defines can be used even for software that needs to be verified to the very highest integrity levels. The aim of this guide is to give a complete description of the motivations behind the Profile, to show how conformant programs can be analysed and to give examples of usage.

Information technology — Programming languages — Guide for the use of the Ada Ravenscar Profile in high integrity systems

1 Scope

This Technical Report provides a description of the motivations behind the Ravenscar Profile, to show how Ada programs using the profile can be analysed, and gives examples of usage.

Guide for the use of the Ada Ravenscar Profile in high integrity systems

Alan Burns, Brian Dobbing and Tullio Vardanega

University of York Technical Report YCS-2003-348
January 2003

Contents

1 Introduction

There is increasing recognition that the software components of critical real-time applications must be provably predictable. This is particularly so for a hard real-time system, in which the failure of a component of the system to meet its timing deadline can result in an unacceptable failure of the whole system. The choice of a suitable design and development method, in conjunction with supporting tools that enable the real-time performance of a system to be analysed and simulated, can lead to a high level of confidence that the final system meets its real-time constraints.

Traditional methods used for the design and development of complex applications, which concentrate primarily on functionality, are increasingly inadequate for hard real-time systems. This is because non-functional requirements such as dependability (e.g. safety and reliability), timeliness, memory usage and dynamic change management are left until too late in the development cycle.

The traditional approach to formal verification and certification of critical real-time systems has been to dispense entirely with separate processes, each with their own independent thread of control, and to use a *cyclic executive* that calls a series of procedures in a fully deterministic manner. Such a system becomes easy to analyse, but is difficult to design for systems of more than moderate complexity, inflexible to change, and not well suited to applications where sporadic activity may occur and where error recovery is important. Moreover, it can lead to poor software engineering if small procedures have to be artificially constructed to fit the cyclic schedule.

The use of Ada has proven to be of great value within high integrity and real-time applications, albeit via language subsets of deterministic constructs, to ensure full analysability of the code. Such subsets have been defined for Ada 83, but these have excluded tasking on the grounds of its non-determinism and inefficiency. Advances in the area of schedulability analysis currently allow hard deadlines to be checked, even in the presence of a run-time system that enforces preemptive task scheduling based on multiple priorities. This valuable research work has been mapped onto a number of new Ada constructs and rules that have been incorporated into the Real-Time Annex of the Ada language standard [RM D]. This has opened the way for these tasking constructs to be used in high integrity subsets whilst retaining the core elements of predictability and reliability.

The Ravenscar Profile is a subset of the tasking model, restricted to meet the real-time community requirements for determinism, schedulability analysis and memory-boundedness, as well as being suitable for mapping to a small and efficient run-time system that supports task synchronization and communication, and which could be certifiable to the highest integrity levels. The concurrency model promoted by the Ravenscar Profile is consistent with the use of tools that allow the static properties of programs to be verified. Potential verification techniques include information flow analysis, schedulability analysis, execution-order analysis and model checking. These techniques allow analysis of a system to be performed throughout its development life cycle, thus avoiding the common problem of finding only during system integration and testing that the design fails to meet its non-functional requirements.

It is important to note that the Ravenscar Profile is silent on the non-tasking (i.e. sequential) aspects of the language. For example it does not dictate how exceptions should, or should not, be used. For any particular application, it is likely that constraints on the sequential part of the language will be required. These may be due to other forms of *static analysis* to be applied to the code, or to enable *worst-case execution time* information to be derived for the sequential code. The reader is referred to the ISO Technical Report, Guide for the Use of Ada Programming

Language in High Integrity Systems [GA] for a detailed discussion on all aspects of static analysis of sequential Ada.

The Ravenscar Profile has been designed such that the restricted form of tasking that it defines can be used even for software that needs to be verified to the very highest integrity levels. The Profile has already been included in the ISO technical report [GA] referenced above. The aim of this guide is to give a complete description of the motivations behind the Profile, to show how conformant programs can be analysed and to give examples of usage.

Structure of the Guide

The report is organized as follows. The motivation for the development of the Ravenscar Profile is given in the next chapter. Chapter 3 includes the definition of the profile as agreed by *WG9*; the definition is included here for convenience, but this report is not the definitive statement of the profile. In Chapter 4, the rationale for each aspect of the profile is described. Examples of usage are then provided in Chapter 5. The need for verification is an important design goal for Ravenscar and Chapter 1 reviews the verification approach appropriate to Ravenscar programs. Finally in Chapter 7 an extended example is given. Definitions and references are included at the end of the report.

Readership

This report is aimed at a broad audience, including application programmers, implementers of run-time systems, those responsible for defining company/project guidelines, and academics. Familiarity with the Ada language is assumed.

Conventions

This report uses the *italics* face to flag the first occurrence of terms that have a defining entry in Chapter 8. For all Ada-related terms the report follows the language reference manual [RM] style: it uses the Arial font where there is a reference to defined syntax entities (e.g. delay_relative_statement). For all other names (e.g. Ada.Calendar) it uses normal text font, as do language keywords in the text except that they are in **bold** face.