TECHNICAL REPORT

ISO/IEC TR 15942

First edition
2000-03-01

Information technology — Programming languages — Guide for the use of the Ada programming language in high integrity systems

Technologies de l'information — Langages de programmation — Guide pour l'emploi du langage de programmation Ada dans les systèmes de haute intégrité

Reference number
ISO/IEC TR 15942:2000(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 734 10 79
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Contents

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

In exceptional circumstances, when a 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), it may decide by a simple majority vote of its participating members to publish a Technical Report. A Technical Report is entirely informative in nature and does not have to be reviewed until the data it provides are considered to be no longer valid or useful.

Attention is drawn to the possibility that some of the elements of this Technical Report 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 15942 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*.

Introduction

As a society, we are increasingly reliant upon high integrity systems: for safety systems (such as fly-by-wire aircraft), for security systems (to protect digital information) or for financial systems (e.g., cash dispensers). As the complexity of these systems grows, so do the demands for improved techniques for the production of the software components of the system.

These high integrity systems must be shown to be fully predictable in operation and have all the properties required of them. This can only be achieved by analysing the software, in addition to the use of conventional dynamic testing.

There is, currently, no mainstream high level language where all programs in that language are guaranteed to be predictable and analysable. Therefore for any choice of implementation language it is essential to control the ways that the language is used by the application.

The Ada language [ARM] is designed with specific mechanisms for controlling the use of certain aspects of the language. Furthermore,

1. The semantics of Ada programs are well-defined, even in error situations. Specifically, the effect of a program can be predicted from the language definition with few implementation dependencies or interactions between language features.

2. The strong typing within the language can be used to reduce the scope (and cost) of analysis to verify key properties.

3. The Ada language has been successfully used on many high integrity applications. This demonstrates that validated Ada compilers have the quality required for such applications.

4. Guidance can be provided to facilitate the use of the language and to encourage the development of tools for further verification.

Ada is therefore ideally suited for implementing high integrity software and this document provides guidance in the controls that are required on the use of Ada to ensure that programs are predictable and analysable.

All language design balances functionality against integrity. For instance, the ability to control storage allocation directly will impact the need to ensure the integrity of data. An aspect of the integrity of Ada programs is the possibility of avoiding access types (references) completely, whereas in other languages references are linked to array accessing and/or parameter passing, and therefore cannot be excluded.

There are a number of different analysis techniques in use for high integrity software and this document is not prescriptive about which techniques to use. Furthermore, each analysis technique requires different controls on the use of the language. Ada assists analysis: for instance, the modes of Ada parameters, suitably used, provide information for data flow analysis which other languages cannot always provide. This Technical Report, therefore, catalogues specific verification techniques (see 2.5), and classifies the impact that language features have on the use of these techniques (in the tables in Section 5).

It is the user's responsibility to select the analysis techniques for a particular application; this document can then be used to define the full set of controls necessary for using that set of techniques.

The guidance given here first specifies its scope, by reference to the safety and security standards to which high integrity applications may be written.

Section 2 then analyses the verification techniques that are applied in the development of high integrity systems. By this means, the regulatory rules of the standards for safety and security are abstracted to avoid the need to consider each such standard separately.

Section 3 addresses general issues concerning how computer languages must be constructed if programs written in that language are to be fully predictable. These issues are relevant to any restricted language defined through the application of this guidance.

Section 4 provides identification of a three-way classification system used for Ada language features. This classification is based upon the ease with which verification techniques can be applied to a program containing the feature. This classification is needed since while the majority of the core features in Ada assists verification, the use of certain features makes the resulting code difficult or impossible to analyse with the currently available program analysis tools and techniques.

Section 5 provides the main technical material of this Technical Report by classifying Ada language features. Users of this Technical Report can then determine which features of Ada are appropriate to use from the verification techniques that are to be employed. The assessment has shown that the vast majority of the Ada features lend themselves to effective use in the construction of high integrity systems.

The Technical Report concludes, in Section 6, by providing information to aid the choice of a suitable Ada compiler together with its associated run-time system.

References to relevant standards and guides are provided. A detailed analysis of Ada95 for high integrity systems is available in References [CAT1, CAT2] and [CAT3] .

A comprehensive index is provided to ease the use of the Technical Report.

Levels of criticality

Many of the Standards to which high integrity software is written use multiple levels to classify the criticality of the software components which make up the system. While the number and nature of the levels vary, the general approach is always the same: the higher the criticality of the system, the more verification techniques need to be used for its assurance. Table 1 relates the various levels of classification used in some well known International Standards.

Table 1: Levels of criticality in some Standards

Standard	Number of levels	Lowest Level	Highest Level
[DO-178B]	4	D	A
[IEC-61508]	4	Safety Integrity Level 1	Safety Integrity Level 4
[ITSEC]	7	E0	E6

This Technical Report emphasizes the higher levels of criticality, for which the more demanding verification techniques are employed and for which Ada provides major benefits.

This Technical Report, however, does not directly use any such levels but focuses on the correlation between the features of the language and the verification techniques to be employed at the higher levels of criticality. The material in [ISO/IEC 15026], [DS 00-56], [ARP 4754] and [ARP 4761] may be useful in determining the criticality of a system if this is not covered by application-specific standards.

Readership

This Technical Report has been written for:

1. Those responsible for coding standards applicable to high integrity Ada software.
2. Those developing high integrity systems in Ada.
3. Vendors marketing Ada compilers, source code generators, and verification tools for use in the development of high integrity systems.
4. Regulators who need to approve high integrity systems containing software written in Ada.
5. Those concerned with high integrity systems who wish to consider the advantages of using the Ada language.

This Technical Report is not a tutorial on the use of Ada or on the development of high integrity software. Developers using this report are assumed to have a working knowledge of the language and an understanding of good Ada style, as in [AQS].

History

When proposals were made that a subset of Ada should be specified for high integrity applications, it was realized that the provision of the Safety and Security Annex in the Ada standard did not satisfy all the requirements of the developers of high integrity systems. In consequence, a group was formed under WG9 to consider what action was needed. This group, called the HRG, proposed and drafted this Technical Report over a three year period.

Conventions

In line with the Ada standard, the main text is in a Roman font. Ada identifiers are set in a sans-serif font, and the Ada keywords in a bold sans-serif font.

Postscript

The guidance provided here reflects the understanding of the issues based mainly on using the previous Ada standard in developing high integrity applications. Over the next few years, the current Ada standard will be used for further high integrity applications which will no doubt need to be reflected in a revision of this guidance. Specifically, further detail can be produced based upon the experience gained.

Instructions for comment submission

Informal comments on this Technical Report may be sent by e-mail to hrg@cise.npl.co.uk. If appropriate, the project editor will document the issue for corrective action.

Comments should use the following format:

```
!topic: Title which is a summary in one sentence
!reference TR 15942- ss.ss.ss
!from Author, Name, yy-mm-dd
!keywords keywords related to topic
!discussion
text of discussion
```

where *ss.ss.ss* is the section number, *yy-mm-dd* is the date. If the comment requests a change, a rationale for this and the substance of the actual change proposed would facilitate the processing of the comment.

Information technology - Programming languages - Guide for the use of the Ada programming language in high integrity systems

1 Scope

This Technical Report provides guidance on the use of Ada when producing high integrity systems. In producing such applications it is usually the case that adherence to guidelines or standards has to be demonstrated to independent bodies. These guidelines or standards vary according to the application area, industrial sector or nature of the risk involved.

For safety applications, the international generic standard is [IEC 61508] of which part 3 is concerned with software.

For security systems, the multi-national generic assessment guide is [ISO CD 15408].

For sector-specific guidance and standards there are:

Airborne civil avionics: [DO-178B]

Nuclear power plants: [IEC 880]

Medical systems: [IEC 601-4]

Pharmaceutical: [GAMP]

For national/regional guidance and standards there are the following:

UK Defence: [DS 00-55]

European rail: [EN 50128]

European security: [ITSEC]

US nuclear: [NRC]

UK automotive: [MISRA]

US medical: [FDA]

US space: [NASA]

The above standards and guides are referred to as Standards in this Technical Report. The above list is not exhaustive but indicative of the type of Standard to which this Technical Report provides guidance.

The specific Standards above are not addressed individually but this Technical Report is synthesized from an analysis of their requirements and recommendations.

