An Ada95 Solution for Certification of Embedded Safety Critical Applications

Jacob Frost

DDC International
Gl. Lundtoftevej 1B
DK-2800 Lyngby, Denmark
Tel: +45 45871144
Fax: +45 45872217
Email:jf@ddci.dk

Abstract: The Ada95 programming language is widely used for implementation of embedded safety-critical airborne systems and equipment. The RTCA/DO-178B is a mature standard for certifying such systems and equipment through testing. This paper focuses on the solution DDC-I will provide for RTCA/DO-178B level A certification of Ada95 applications through its new SCORE compiler/debugger product line and associated test tools products. Furthermore, the paper outlines how the DDC-I approach relates to the work in the ESPRIT OMI/SAFE project which aims at establishing a complete software factory for the development and certification of embedded safety-critical applications.

1 Introduction

The use of software in safety-critical systems continues to increase rapidly. In addition to being used in an increasing number of real-time embedded systems, software also plays an increasingly critical role in these systems. Some of the most prominent examples of this trend can be found in the avionics and automotive industry. In both airplanes and cars, software controls an increasing number of safety-critical functions such as flight control and breaking. In addition, the traditional mechanical backup systems are starting to disappear as the industry introduces new systems that rely entirely on software.

Such use of embedded systems place strong demands on the correctness and reliability of the associated software. Formal methods have been suggested to deal with the increased demands to software correctness in embedded systems. However, industry still relies on strict discipline and exhaustive testing to provide the required level of confidence.

The high demands on modern embedded software caught the attention of designers of programming languages and organisations such as the RTCA back in the early eighties. The Ada programming language, which is widely used in safety-critical embedded applications, encourages and enforces a strict and clean style of programming. In its latest incarnation, Ada95 [4], it provides a dedicated safety and security annex (Annex H) supporting the development of safety-critical software. Similarly,
the RTCA has published a mature set of guidelines called the RTCA/DO-178B [3] dealing with the requirements to certification of safety-critical software such as that found in airborne systems.

This paper focuses on the solution DDC-I plans to provide for RTCA/DO-178B certification of Ada95 applications through its new SCORE compiler/debugger product line and associated test tools products. In particular, the paper describes the support of Annex H and two dedicated test tools supporting level A (highest level) of certification/testing according to the RTCA/DO-178 guidelines. Furthermore, the paper briefly outlines how the DDC-I approach relates to the work in the ESPRIT OMI/SAFE project which aims at establishing a complete software factory for the development and certification of embedded real-time safety-critical applications.

The rest of this paper consists of two sections and a conclusion. The first section describes how DDC-I plans to provide support for Annex H and DO-178B certification of embedded real-time safety-critical applications. This includes a general overview of the approach and more details about the specific tools developed. The second section provides a brief description of the OMI/SAFE project and its goal to develop a complete software factory.

2 The DDC-I Certification Tools

The goal of the DDC-I software development tools/features described in this section, is to provide software developers with a basic set of no-nonsense tools and features supporting development and certification of safety-critical embedded real-time applications. Each of the tools/features described are designed to complement each other, addressing different issues of software certification. The remaining of this section describes the implementation of Annex H, the CoverCode structural coverage test tool and finally the requirements coverage test tool.

2.1 Ada95 Annex H

The Ada95 core language has a number of features that makes it particular suitable for the development of safe and secure software. The strong typing system, pointer initialisation, extensive run-time checks, a robust exception mechanism, abstract data types, packages are all examples of such features. Annex H adds a number of new features. The main issues are the following: understanding program execution, reviewing object code and restricting the use of certain language features. The SCORE Ada95 compiler from DDC-I will provide a full implementation of Annex H.

Understanding Program Execution. Annex H supports the understanding of program execution through pragma Normalize_Scalars and by documenting implementation decisions.

The use of pragma Normalize_Scalars forces all otherwise uninitialised scalars to be assigned a predictable value initially. If possible an invalid value will assigned to the scalar. As examples of the strategy, consider the declarations in figure 1. The