Foreword

The moment I embarked upon writing this foreword to the Rationale for Ada 2005 I couldn't help going back to the previous Rationale document I had seen, that for Ada 95, which was for me the culmination of an extraordinary encounter with the Ada programming language. In actual fact, Ada was not part of the Computer Science curriculum of the course in which I enrolled at the University of Pisa, Italy, in 1981. However, I later learned that some of the local professors had taken a genuine interest in the birth of Ada 83 and had for a while entertained the possibility of procuring some licences of the Ada compiler for use in the department laboratory.

So I graduated without getting even the slightest glimpse of Ada and went to work continuing in complete ignorance of it. My encounter with Ada had to wait a few more years, until I joined the European Space Agency at their Research and Technology centre in the Netherlands in 1991. I then had the good fortune of becoming in short succession first an ESA observer in the Ada 9X process (the process that led to Ada 95) and then also the representative for the Dutch delegation in WG9, the body within ISO/IEC JTC1/SC22 which has responsibility for the maintenance of the Ada standard. From that vantage point I enjoyed an exceptionally stimulating insider view of the inner nature of Ada, its exceptional beauties as well as its limitations and defects.

There is one adjective that best summarizes the impression I drew from that encounter with Ada 95 in-the-making, and which I want to share with you in this Foreword. The adjective is: honest – and it has stayed in my mind ever since. Ada is an honest language, it makes you say what you want and it tells you what happens. Ada prizes that character over magic and marketing. I can easily go back in my memory to the moment when this evidence dawned on me. It was at a discussion in November 1992 in Salem – that place well known for historic witchcraft. The object of the discussion was the need for the "class" keyword in Ada. You know that Ada does not have it. Should it? Speaking as first a user of Ada and later as an educator, I am now convinced that the decision was wise and brought the full OO paradigm into Ada in a clear and honest manner, like all the rest of it in fact. Ada 2005 has made the use of OOP easier and richer, but still following that principle of honesty: you get OO if you want it and no witchery happens behind your back that you don't know about.

Over 15 years after, now that I am looking at the Rationale for Ada 2005, I am really delighted at the wealth of improvements with which the dedicated community of the ARG (the body of technical experts that actually does the language standard maintenance work) has enhanced Ada over the last two decades.

How confused are those who proclaim that Ada is a dead language. Of course it is not! It is amazing to see how many advanced and forward-looking features have been built upon the core of Ada 83 without disrupting its very foundation, which was and continues to be: to value and to seek solidity, rigour and robustness. Among other things, Ada 95 made breakthrough advances in the tasking model, opened the door to object orientation (though perhaps somewhat timidly), and provided a neat model for distributed computing. Despite being only meant to be an *amendment* (whereas Ada 95 was a revision), Ada 2005 has gone further down the road of improvements and advancements. We can only be grateful to the members of the ARG for having put together such a beautiful language. They have included a number of intriguing and attractive ideas and implemented them in a coherent manner as appropriate to the level of perfection necessary for the diligent maintenance of a language standard.

It is not for me to present those new features and to lead you to their use: that is the job of the author of the Rationale, John Barnes, who has accompanied this language from its very beginning and understands all the key features of it. More importantly, John is able to explain them in a clear and accessible manner to real programmers, practitioners and educators alike.

I am especially pleased that Ada-Europe has been instrumental in the production of the Rationale for Ada 2005 from the very beginning. I had the good fortune of being the editor-in-chief of the Ada User Journal (the AUJ), the quarterly publication of Ada-Europe, when we lured John into producing chapters of the would-be Rationale for publication in the AUJ; in that position I was most probably the first to read the draft of those chapters and I have to say I enjoyed them all and so I believe afterwards did the whole readership of the journal.

Some while after the publication of the last chapter in the AUJ, Ada-Europe resumed the project of producing a volume in the LNCS series entirely devoted to the Rationale. Now that the project has been successfully completed, I have the privilege of writing this Foreword in my current capacity of President of Ada-Europe. Who can thus be happier than I at the moment? I am delighted therefore that the very first milestone of my tenure as President is the delivery of this precious gift to the doorstep of the members of Ada-Europe, whether direct, honorary or corporate members or via their own national organization. I wish good reading to you all and I promise you it'll be worth it!

Tullio Vardanega
Ada-Europe, President
Padua
Italy
February 2008

Preface

Welcome to Ada 2005. This Rationale should enable those familiar with Ada 95 to gain a basic understanding of the new features introduced into Ada 2005 and the reasons for the changes from Ada 95.

This document starts with an introduction which provides an overview of the changes. There are then a number of major chapters looking at six key areas (OOP, access types, structure and visibility, tasking, general stuff, predefined library and containers) and finally an epilogue largely concerned with compatibility issues.

Earlier versions of individual chapters were previously published in the Ada User Journal as a number of separate papers in the issues from December 2004 until March 2006.

I need to thank all those who have helped in the preparation of this document. First I must acknowledge the help of Ada-Europe and the Ada Resource Association for financial support for attending various meetings. And then I must thank those who reviewed earlier versions. There are almost too many to name, but I must give special thanks to Randy Brukardt, Pascal Leroy and Tucker Taft of the ARG, to my colleagues on the UK Ada Panel (BSI/IST/5/–/9), and to James Moore of WG9.

I am especially grateful for a brilliant suggestion of Randy Brukardt which must be preserved for the pleasure of future generations. He suggested that this document when completed be called the Ada Language Enhancement Guide. This means that if combined with the final Ada Reference Manual, the whole document can then be referred to as the ARM and ALEG. Thanks Randy.

I must also thank Randy for his efforts in creating a version of this Rationale for the web and especially for creating an extremely valuable index which is incorporated here.

Writing this rationale has been a learning experience for me and I trust that readers will also find the material useful in learning about Ada 2005. An integrated description of Ada 2005 as a whole including some further examples will be found in the latest version of my textbook which is entitled *Programming in Ada 2005* [13].

Finally, for the full details, please consult the Ada 2005 Reference Manual which is also published in the LNCS series [14]. There is also a version known as the Annotated Ada Reference Manual which contains much additional commentary. This will be found online [15].

John Barnes
Caversham
England
January 2008