Abstract. This paper is a report from the ISO/Ada Realtime Rapporteur Group on its current activities. It presents the status of the project on focuses on the way the editing team has handled the compatibility with Ada9X.

1 Introduction

1.1 What is the RRG
In 1991, a New Project (JTC1 N1266) was assigned by ISO/IEC JTC1 to SC22 and then attributed to WG9(Ada) with the objective to defined standardized Ada real time extensions; the Realtime Rapporteur Group (RRG) was created inside WG9. Like the other Rapporteur Groups of WG9, the RRG focuses on a special area, namely HRT. Indeed, WG9 has established several rapporteur groups. Some are tasked to produce secondary standards: NRG (related to numeric computation), SRG (related to SQL Ada binding) or Technical Reports CRG (related to characters). Others are in charge of language maintenance (ARG), implementation uniformity (URG), Ada9X revision (XRG).

In this context, the objective of the RRG is to provide Ada packages for Hard Real-Time applications (e.g., safety Critical) for the Ada language.

1.2 What is the status of the RRG work
The RRG has decided to publish its results in the Technical Reports (Type 2) series of publications (according to sub-clause G.4.2.2 of ISO/IEC directives part 1) as a «prospective standard of provisionnal application» in the field of hard real time applications developed using the Ada programming language, because there is an urgent need for guidance on how standards in this field should be used to meet identified need.
The RRG has considered that it was timely to publish its results as a Technical Report of Type 2 because of the urgency to meet the requirements in the field of hard real time applications programmed in Ada and because of the revision project of Ada with which this work is related. The Technical Report of Type 2 status attributed to this document will allow for more flexible evolution within ISO and in particular, it can be revised in a short amount of time (3 years) after its publication in order to meet new requirements that will arise after the publication of Ada9X. The proposed technical solutions can also take advantage of the new functionalities introduced in Ada9X.

2 RRG is full compatible with Ada9X

The revision project of Ada (Ada9X) is now in its final step and the new standard is forseen by the end of 94. Given this context, the RRG has focused its efforts towards the compatibility with Ada9X.

Full compatibility with Ada9X is defined as the possibility to use simultaneously the Ada9X features and the RRG features in the same application in a very confident way. Moreover, an Ada83 application that uses the RRG features do not need to change its code (the part that uses RRG) in order to transition to Ada9X.

Diferent steps have been adopted to achieve such a goal:

Identify clearly the relation classes

Four categories of relations between Ada9X features and features of the RRG are identified:

- **obsolescent features**: library units that are totally subsumed by Ada9X. Such library units are defined in the RRG for transition purpose. They are defined as in Ada9X in order to avoid unecessary duplications.

- **library units that are partially subsumed**: These units contains as a subset the related functionality provided by Ada9X. In addition, extensions are defined. The subset is defined as in Ada9X.

- **library units that may be enhanced by Ada9X features**: For example, the package EXTRA_Timers defines timer handlers as procedures (generic formal parameter). In the Ada9X context, a protected procedure can be used instead of normal a procedure.

- **library units that can take full advantage of some Ada9X features**, if the bodies of these units are implemented using Ada9X features. For example, the protected type facility can be used to implement different library units; it is also the case for child units.

Provide a special section to explain such relations

In order to address and to explain the relationship with Ada9X, the RRG document provides an annex that details the relationships, the matrix of relations, the proposed transition policy and interface packages using Ada9X features.