

## Final Minutes of the 1. ARG Meeting, La Jolla, CA, 1-3 November 1995

**Attendees:** (ARG members) Robert Eachus, Stephen Michell, Vincent Celier, Kiyoshi Ishihata, Gary Dismukes, Erhard Ploedereder, Bryce Bardin, Pascal Leroy, John Barnes, Bob Duff, (Associate) Mike Kamrad, (Guest) Keith Thompson.

### Summary to be presented to WG-9:

1. The Draft Procedures of the ARG were unanimously approved.
2. The ARG proposes to produce a Technical Report comprising the AIs and delivered incrementally to WG-9. If requested by WG-9, the ARG will also produce the necessary documentation for ISO Standard Corrigenda. Based on preliminary information about this process, the ARG does not propose to produce such formal Corrigenda.
3. The next ARG meeting will be in Montreux, June 15-17, 1996.
4. The following AIs were approved „as is“ or with very minor editorial changes. (They will now go through editorial review and, barring a request for letter ballot, will be regarded as approved by ARG):

AI95-00002: Subunits of a preelaborated subprogram  
AI95-00027: Typo: is access all {Ada.}Streams.Root\_Stream\_Type'Class  
AI95-00029: Does Complex\_IO handle extended real literals?  
AI95-00036: What are the rules for named notation in pragmas?  
AI95-00038: Mapping between Interfaces.C.char and Standard.Character  
AI95-00043: Predefined Operators for Generic Formal Array Types  
AI95-00046: Ambiguous "only"; please clarify intent  
AI95-00047: {user-defined} Read and Write attributes  
AI95-00048: Can an RCI unit be a library subprogram? *Letter ballot requested*  
AI95-00053: Case sensitivity of Wide\_Value and Value attributes  
AI95-00056: Create(Mode => Append\_File)  
AI95-00057: Text\_IO.Flush should use IN mode  
AI95-00062: Derived access types share the same pool  
AI95-00064: Elaboration checks for renamings-as-body  
AI95-00069: One queuing policy per partition.  
AI95-00082: The PCS may be defined by the user.  
AI95-00091: Pragma Locking\_Policy Cannot Be "In" a Program Unit  
AI95-00093: Float\_Type'Small

5. Resolutions of the following AIs were approved „in principle“. (They will be partially rewritten, sent out for review and letter-balloted. In some cases, the proposed summary was changed in intent.)

AI95-00001: Ada.Streams.Read and Ada.Streams.Write can raise Status\_Error  
AI95-00026: Stream\_IO.Read and Stream\_IO.Write advance the current index  
AI95-00033: delayed declaration of inherited primitive subprograms  
AI95-00039: Effect of Update(Null\_Ptr,...)

AI95-00049: Reserved\_128, etc.  
AI95-00050: External files of Standard\_Input and Standard\_Output  
AI95-00051: Size clauses for objects specifying large sizes  
AI95-00083: Aggregates of a controlled type  
AI95-00101: Abort\_Task has a parameter of mode 'in'.

6. The following AIs were discussed; approval awaits rewritten versions of the AIs. On most, the intent of the resolution is agreed upon, but voting on the AI was not possible due to significant expected change in the writeup.

AI95-00037: Can wchar\_t be signed?  
AI95-00041: Program unit pragmas in generic units  
AI95-00044: Case of Overriding Missing in 8.3 ?  
AI95-00071: Correction to the Valid function in COBOL Interface  
AI95-00072: Clarification of result length for conversions in COBOL Interface  
AI95-00085: Questions about Append\_File mode  
AI95-00090: Priority inheritance in rendezvous from async select  
AI95-00092: Async Task Control while caller in rendezvous  
AI95-00097: Conversions between access types with different representations.

The above lists include all AIs provisionally classified as being of high priority; all but seven medium priority AIs were also handled.

-----  
---

## **Detailed Minutes:**

Wednesday Meeting

### **1. Discussions about the output of the ARG.**

It was agreed that the AIs will be produced essentially in the current form. All AIs will be presented individually to WG-9 for approval and collected into a Technical Report as the main work product of the ARG. Upon direction by WG-9, the ARG will also produce a draft of an official ISO Corrigendum to the Ada Standard. It was reported that the number of Corrigenda to a standard is severely limited (two ?), before a revised standard needs to be issued. This implies that, at best, official Corrigenda will be produced in 3 year intervals. It appears more opportune to treat the AIs as less „official“ and let the self-interest of the vendors and the ACVC cause adherence to the AI interpretations.

A discussion on whether the AIs should be incorporated into some document other than a Technical Report, i.e., into the ARM or AARM was partially inconclusive. The majority opinion is to not produce an ACID '95 document. A decision on whether to have AI annotations in an AARM has been postponed; more discussion and experience with the process is needed.

ASIDE: There was consensus to recommend to OC Systems to reduce its printing of the AARM document to the small size of the ARM.

## 2. Procedures for ARG

Erhard reported that he created the procedures from the old ARG procedures. The revised version requires for vote passage only a 2/3 majority of respondents rather than of total membership. This brought up the subject of how membership is established. Formally, members are appointed by the Convenor of WG-9. Practically, it happens by collaboration between the Convenor of WG-9 and the ARG Chair. Erhard recommended that anybody desiring to be a full member let him know and he will make the necessary things happen. This was acceptable to the ARG.

How does the ARG get more participation from compiler vendors? Erhard will issue an invitation to the vendors to appoint someone from their organization as a prospective ARG member or associate.

A short discussion on the AI categorization scheme, prompted by Erhard's voicing of concern over a large number of „binding interpretations“, yielded no consensus for any change.

The Procedures were accepted unanimously by the attending members.

## 3. Technical issues:

### **AI-69            One queuing policy per partition**

The focus of the discussion centered around the term "such" and its application to all implementations supporting queuing policy or multiple queuing policies. The interpretation in the summary is OK and can be justified by 10.1.5(9). AI approved (10-0-0).

### **AI-90/92        Priority inheritance in rendezvous from async select Async Task Control while caller in rendezvous**

The discussion started with AI-90 but quickly incorporated AI-92. It was clear that mandatory transitive inheritance was not intended as the rule for priority inheritance. So the remaining question was whether the rules should allow for simple and transitive inheritance, the respective choice being implementation-defined, or be simple inheritance only.

Bob Duff will update AI-92 to present both cases with the recommendation that simple inheritance be the rule (10-0-1 straw vote).

### **AI-91            Pragma Locking\_Policy Cannot Be "In" a Program Unit**

Unanimously approved (10-0-0).

### **AI-101          Abort\_Task has a parameter of mode 'in'.**

The summary was approved (9-0-1).

The discussion section will be expanded.

A discussion of the difference between a task and a task object was deemed significant but not relevant to the determination of this AI and therefore should be dealt with in its own AI. Bob Eachus will write a new AI on this subject.

### AI-33            **delayed declaration of inherited primitive subprograms**

The argument here is that the case of nested packages needs a wording change to make it correspond to the parent-child instance with private operations and, more importantly, make it conform with Ada83 rules regarding the visibility of operations on private components of composite types.

The parent-child example is

```
package Parent is
  type T1 is ...;
private
  procedure Op(X: T1);
end Parent;
```

```
package Parent.Child is
  type T2 is new T1;
private
  ...
end Parent.Child;
```

```
with Parent.Child;
package body Parent is
  -- Op (X: T2) is not visible here due to the principle that parent can't
  -- see private operations of its children
```

The analogous Ada83 example is

```
package Parent is
  type T1 is private;
private
  type T1 is new Boolean;
end Parent;
```

```
with Parent; use Parent;
package Child is
  type T2 is array (...) of T1;
private
  ...
end Child;
```

```
with Child;
package body Parent is
  -- „And“ operator is not available on the objects of T2 despite the fact
  -- that parent has visibility of the Boolean operations of T1.
```

Approved (8-0-2) the Wording section as a clearer representation of the AI and the appropriate steps to take. Gary Dismukes takes the action to use the Wording section to improve the Summary section and to add examples showing the Ada83 principle as guidance for Ada95 to preserve upward compatibility.

#### **AI-49          Reserved\_128, etc.**

The evidence for this AI needs stronger confirmation before it can be approved as a whole. An action item for Eachus (with possible involvement by Dewar and Brosgol) to check/confirm the current state of the cited standards (10646, 6429, 8859-1) and amend the AI accordingly.

Approved (8-1-1) the summary with the "changes" phrase dropped. The dissenting vote was cast on the basis that in the absence of definitive information about the cited standards the AI should not be voted upon.

#### **AI-83          Aggregates of a controlled type**

It was observed that the problem equally arises from object components initialized by aggregates. An accordingly revised summary (to apply to all object creations, such as allocations, component initialization, etc.) was approved (7-0-3). Action item for Bob Duff to rewrite the AI in this more general form.

#### **AI-82          The PCS may be defined by the user.**

For discussion purposes, Erhard took the position that this AI was (ab)using the language definition to mandate a particular solution to a licensing issue and that this was stepping beyond the boundaries of language definition. The discussion was spirited but finally the AI was approved unanimously (10-0-0) on Thursday morning.

Thursday Meeting

#### **AI-41          Program unit pragmas in generic units**

This is a tough issue to resolve in terms of general language rules. The argument was made that pragma semantics are already somewhat outside the general language rules and so we should not be afraid to codify unique capabilities for each pragma, if deemed necessary. Erhard takes the action item to amend the AI to discuss the case for each of the affected library unit pragmas rather than relying on general language rules. Voting awaits the changes from Erhard.

#### **AI-37          Can wchar\_t be signed?**

The initial consensus favored the recommendation from Pascal Leroy instead of the summary. Erhard recommend that the AI be rewritten to reflect this consensus, changed to a Binding Interpretation and then subjected to a letter ballot. But when Kiyoshi Ishihata asked for clarification on the treatment of "NUL", it became apparent that the AI needed to be extended (since 'VAL for an enumeration type with negative encodings would not yield the desired result of a „zero“ representation.). The consensus was

- both values will be initialized with an implementation defined value
- note that both values should be discrete
- the initializing value should be represented by zero.

Vote will be delayed until the AI is rewritten by Bob Duff.

**AI-38 Mapping between Interfaces.C.char and Standard.Character**

Approved (9-0-1)

**AI-39 Effect of Update(Null\_Ptr,...)**

Summary approved (9-0-1); some editorial changes to the discussion. It was observed that the current justification inappropriately relies on the raising of Dereference\_Error by Strlen, since Strlen is not called if Check is false.

**AI-71 Correction to the Valid function in COBOL Interface**

At a minimum, the AI discussion needs to include an explanation of the COBOL semantics to which a match is desired. Referred to the Eachus/Dewar/Brosgol for appropriate amendments.

**AI-72 Clarification of result length for conversions in COBOL Interface**

The rationale needs beefing up to respond to an issue raised by Bob Eachus about the length of the format.

Referred to the Roberts and Ben Brosgol for resolution.

**AI-97 Conversions between access types with different representations**

The opposing views:

- Erhard and Bryce: the permission is too broad. An implementation supporting the interfacing with C should be expected to go to some effort to deal with different conventions and only appeal to such a permission for the extreme cases. Also, conversions of such pointers should not be implicitly defined; the user should be required to specify an explicit conversion.
- Bob Duff: the permission is needed because implementation need the flexibility to deal with weird C implementations.

A compromise was struck to limit this AI to only conversions and be silent on other operations (until interpretations are explicitly requested). Vote awaits a rewrite by Bob Duff.

**AI-57 Text\_IO.Flush should use IN mode**

Approved (10-0-0).

**AI-85 Questions about Append\_File mode**

In a late note, Robert Dewar commented on the Append mode. Without Robert to expand on the discussion, no new problem was apparent to the group. The AI will be written to only answer the two points by Keith Thompson. Stephen Michell will reword. By consensus, the AI (if the resolution stays as recommended by Keith) should be categorized as confirmation.

**AI-56 Create(Mode => Append\_File)**

Action Item for Bob Duff: add an answer to the question about the existence of files.

Approved (8-0-2)

Action Item for Robert Dewar: to submit an expanded comment regarding the backscanning issue mentioned in his reply comment.

**AI-27          Typo: is access all {Ada.}Streams.Root\_Stream\_Type'Class**

Bryce wanted paragraph references added in the response because this represents a new rule in Ada95.

Approved with expected references (10-0-0)

**AI-01          Ada.Streams.Read and Ada.Streams.Write can raise Status\_Error**

Modification to the summary was substantial. Reverse the position in the first sentence to raise Status\_Error and then add Text\_IO and Wide\_Text\_IO streams to Read and Write operations. Approved with those changes (7-0-3) and referred to Norm for changes.

**AI-26          Stream\_IO.Read and Stream\_IO.Write advance the current index**

Modification to the AI to reflect the Append mode semantics. Approved (10-0-0) and sent back to Norm for changes.

**AI-29          Does Complex\_IO handle extended real literals?**

Approved (9-0-1).

**AI-93          Float\_Type'Small**

Approved (10-0-0).

**AI-62          Derived access types share the same pool**

Approved (9-0-1).

**AI-64          Elaboration checks for renamings-as-body**

Approved with very minor editorial changes (8-0-2).

**AI-48          Can an RCI unit be a library subprogram?**

The most compelling reason to reverse the position stated in the summary is uniformity. The most compelling reason to maintain the position is the cost of potential wording changes, since the ARM was written with the summary position in mind. Approved (5-0-5). Letter ballot was requested.

The remainder of Thursday was spent on an initial discussion of the issues that arise from 'Size' representation items.

Friday Meeting

## **AI-51            Size clauses for objects specifying large sizes**

There was a lengthy discussion on how large the size of objects can be made by the implementation and how the implementation treats the extra bits that exceed the range of the object values. In particular, various issues regarding aliased objects were discussed. After considerable discussion the group approved three modifications to the summary:

- An implementation need not support a Size clause for an object of discrete subtype that specifies a size that is greater than the largest size of the class of the object's type (9-0-0)
- Size clauses on aliased objects need not be supported (8-0-1)
- An implementation need not support an Alignment clause for an aliased object that specifies an alignment that is less restrictive than the alignment of the object's subtype (9-0-0)

## **AI-02            Subunits of a preelaborated subprogram**

Approved with an editorial comment that nested and subunit packages should act the same. (9-0-0)

## **AI-36            What are the rules for named notation in pragmas?**

This AI confirms an anomaly that prohibits pragmas to support parameter reordering in named notation. Approved (4-0-5) with an editorial change that makes clear that optional parameters in the syntax count for positional association (and can therefore only be omitted, if named association is used).

## **AI-43            Predefined Operators for Generic Formal Array Types**

This AI is needed to correct a problem resulting from the legacy of Ada83 in 12.5(8). Approved with editorial modification to point out the legacy cause of this problem (8-0-1).

## **AI-44            Case of Overriding Missing in 8.3 ?**

This AI needs to be rewritten and Bob Duff was given the action item to do so. This includes the replacement of "hide" terminology with "overrideable" terminology.

## **AI-46            Ambiguous "only"; please clarify intent**

Approved with minor editorial changes and moved to Confirmation category (8-0-1).

## **AI-47            {user-defined} Read and Write attributes**

There is a need to add some examples in the discussion and delete the first (irrelevant) occurrence of the term "limited" in the summary. Approved with these changes (7-0-2).

## **AI-50            External files of Standard\_Input and Standard\_Output**

The intent is clear that Standard\_Input, Standard\_Output and Standard\_Error are distinct file objects, but not necessarily distinct external files, as noted in A.10.3(25). The summary

needs to promote this note to be a language rule. Further citations of A.10(5) and A.14(2-3) should be added. The summary will be updated accordingly, add Standard\_Error and say that the association to external files is not defined by the language.

Approved (9-0-0).

### **AI-53 Case sensitivity of Wide\_Value and Value attributes**

Approved (7-0-2).

This concluded the work on AIs. The broader discussion of the Size issue was then continued. Discussion on the alternatives from Tucker Taft and Robert Dewar for T'Size for subtypes showed that there are problems with both, especially with surprising „remote“ effects when subtype declarations are modified. Next step is to get positions for each alternative formalized through discussion and AI write-ups.

### **Final Administrative Details**

Stephen Michell is starting a membership drive for the DRG. Interested parties should contact him.

The group approved the next meeting to be on **15-17 June 96 in Montreux** after the Ada-Europe Conference and WG-9 meeting.

The group thanked Gary Dismukes and SAIC for the excellent meeting facilities.

### **Summary of Action Items:**

Erhard Ploedereder: Invite vendors, AI-41, initiate letter ballots, set up Montreux meeting

Bob Eachus: see AI-101, AI-49, AI-71, AI-72

Gary Dismukes: AI-33

Robert Dewar: AI-49, AI-71, AI-72, see AI-56, write-up of Size repr. item issue

Tucker Taft: write-up of Size repr. item issue

Ben Brosgol: AI-49, AI-71, AI-72

Norm Cohen: AI-01, AI-26

Bob Duff: AI-\*\*

ALL: review revised AIs as they appear, cast letter ballots