

Minutes of Electronic ARG Meeting 62K

3 June 2021

Attendees: Steve Baird, John Barnes, Randy Brukardt, Arnaud Charlet, Jeff Cousins, Gary Dismukes, Claire Dross, Bob Duff, Brad Moore (off the call between 11:40 and 12:28 for a COVID shot), Jean-Pierre Rosen, Ed Schonberg, Justin Squirek, Tucker Taft, Tullio Vardanega, Richard Wai.

Observers: None.

Meeting Summary

The meeting convened on Thursday, 3 June 2021 at 10:36 hours EST and adjourned at 12:35 hours EDT. The meeting was held using Google Meet. The meeting covered the entire agenda other than optional items.

AI Summary

The following AIs were approved:

- AI12-0425-1/03 Presentation issues from public review (15-0-0)
- AI12-0429-1/01 Examples should have consistent lead-ins (13-0-2)
- AI12-0430-1/02 Examples issues from WG 9 review (15-0-0)
- AI12-0431-1/02 Conforming predicate shouldn't change semantics (12-0-2)
- AI12-0433-1/01 The simple name of a root directory (14-0-0)
- AI12-0436-1/01 Should a parallel block include exception handlers? (8-1-5)

The following AIs were approved with editorial changes:

- AI12-0333-1/04 Predicate checks on out parameters (14-0-0)
- AI12-0426-1/05 Presentation issues from WG 9 review (15-0-0)
- AI12-0427-1/05 Fixups from WG 9 review (14-0-1)
- AI12-0428-1/01 "same declaration list" requirement is too strong (13-0-2)
- AI12-0432-1/02 View conversions of assignments and predicate checks (14-0-0)
- AI12-0434-1/01 Equality operators for container cursors (14-0-0)
- AI12-0435-1/01 Fixups from WG 9 issue #143 (12-0-2)

Detailed Minutes

Welcome

Steve welcomes everyone.

Apologies

No one sent apologies.

Previous Meeting Minutes

There were no comments on the minutes: Approve minutes: 15-0-0.

Date and Venue of the Next Meeting

We do not have a meeting proposed at this time. We need to determine our next steps first.

WG 9 Review

Randy summaries the WG 9 review. There were 179 comments received (after splitting a few comments); some of these covered multiple items. [This was about 3 times the expected amount – Editor.] 42 comments were judged to

require no action. 21 comments were deferred. Most of the deferred comments were improvements (not corrections) for wording unchanged since the original Ada 2012. These were judged out of bounds at this time as any fix can introduce a bug, and making changes where not necessary adds more risk than gain at this last stage. The remainder were unlikely scenarios without obvious answers (for instance, Steve raised a question about the accessibility of the loop parameter of a reduction expression).

All of the remaining 116 comments were addressed. A few were white space changes that are not generally recorded (like bad line breaks or the layout of parameters in a language-defined subprogram). Many were able to be applied as Editorial Review changes on existing AIs that are not yet WG 9 approved (all such changes do not change the intent of the wording and most have no semantic effect). Remaining comments on AARM notes are recorded in AI12-0005-1. And the rest of the comments were turned into AIs for this meeting to discuss and approve. For these, the original comments are recorded in the !appendix of the AI.

ARG Approval of Draft

After all of the AIs were discussed, we took a vote to approve this result.

The additional AIs will be incorporated into the RM, and then the RM will be formally given to WG 9 for its approval.

Vote to approve RM and submit to WG 9: 15-0-0.

We give ourselves a round of applause for completing Ada 202x. We give an additional round of applause for our editor, for his efforts in processing a mountain of comments in a short period of time.

Unfinished Action Items

There are two unfinished action items (Steve Baird, AI12-0016-1; Tucker Taft: OpenMP Technical Report). We did not discuss these.

Current Action Items

The combined unfinished old action items and new action items from the meeting are shown below.

Steve Baird:

- AI12-0016-1

Randy Brukardt:

- Send the RM to WG 9 when ready.

Editorial changes only:

- AI12-0333-1
- AI12-0426-1
- AI12-0427-1
- AI12-0428-1
- AI12-0432-1
- AI12-0434-1
- AI12-0435-1

Tucker Taft:

- AI12-0346-1 – also, construct the Technical Report suggested by this AI.

Detailed Review

The minutes cover detailed review of Ada 2012 AIs (AI12s). The AI12s are presented in numeric order, which is not necessarily the order in which they were discussed. Votes are recorded as “for”-“against”-“abstentions”. For instance, a vote of 6-1-2 would have had six votes for, one vote against, and two abstentions.

If a paragraph number is identified as coming from the working Ada 202x AARM, the number refers to the text in draft 30 of the Ada 202x AARM. Paragraph numbers in other drafts may vary. Other paragraph numbers come from the final consolidated Ada 2012 AARM; again the paragraph numbers in the many drafts may vary.

Detailed Review of Ada 2012 AIs

AI12-0333-1/04 Predicate checks on out parameters

The discussion of AI12-0432-1 noted that this AI should have been classified as a Binding Interpretation – we don't want Ada 2012 implementations doing predicate checks differently from Ada 202x implementations.

Change classification of AI to Binding Interpretation: 14-0-0.

AI12-0425-1/03 Presentation issues from public review

These were last minute things found just before publishing Draft 29 (the one reviewed by WG 9). These should not change any semantics.

Approve AI: 15-0-0.

AI12-0426-1/05 Presentation issues from WG 9 review

Typo: “refernece” in the !wording section.

Approve AI with changes: 15-0-0.

AI12-0427-1/05 Fixups from WG 9 review

Confusing in !problem (10). “Tucker notes... language bug {because}[in that] code...”

(14) Tucker suggests adding a `chunk_specification` to parallel blocks. Wording is proposed at the end of the !appendix.

After discussion, we agree to replace (14) with Tucker's proposal.

(17) “statically denotes” → “statically names”. Replace the round quotes as they don't display properly on some systems.

In !problem (17): “we now [has]{have}...”

Brad: typo in problem (1): “type{s} do not have names...”. “The RM [we] doesn't ...”

Jeff: problem 15: should verb be plural? We split down the middle on whether it should be changed. Perhaps adding commas would be better:

AI12-0355-2{,} as well as the previous issue{,} add{s} aspect_specifications to certain statements and expressions. 13.1.1(11/5) says:

Approve AI with changes: 14-0-1. Arnaud abstains.

AI12-0428-1/01 “same declaration list” requirement is too strong

Tucker wonders if other aspects have this problem.

Tucker finds Default_Iterator has the same problem. The editor will add corresponding wording to that.

Approve AI with changes: 13-0-2 Ed, John abstain.

AI12-0429-1/01 Examples should have consistent lead-ins

This AI was missing from the agenda and indexes, [The header of this AI was missing all status lines which prevented the tools from including it in any index or report.- Editor.] Steve asks if there is any objection to voting on this AI; none was voiced.

Tucker explains the AI; he is making the presentation of the examples consistent throughout the RM.

Approve AI: 13-0-2. Claire, Arnaud abstain.

AI12-0430-1/02 Examples issues from WG 9 review

Approve AI: 15-0-0.

AI12-0431-1/02 Conforming predicate shouldn't change semantics

This is only in the case where the predicate is statically True. We don't try to solve the problem for a predicate that is confirming for a subtype.

Approve AI: 12-0-2. Randy, Claire abstain.

AI12-0432-1/02 View conversions of assignments and predicate checks

For 3.2.4(31/5) use Randy's simplifications.

Gary wonders why we treat target names differently in these rules. We want the semantics of @ to match as closely as possible the semantics of writing the name explicitly. If you write T(X) in an expression, certainly the predicates of T will be checked. So we want to do that for @ as well if it represents T(X).

Randy notes that this is an Amendment as it depends on AI12-0333-1 which was an Amendment. It seems like this should apply to Ada 2012 as well. Agreed, change these to Binding Interpretations.

Approve AI with changes: 14-0-0.

AI12-0433-1/01 The simple name of a root directory

Approve AI: 14-0-0.

AI12-0434-1/01 Equality operators for container cursors

Choose (3): change “predefined” to “primitive”.

Approve AI with changes: 14-0-0.

AI12-0435-1/01 Fixups from WG 9 issue #143

(1) A null extension would become a record aggregate.

(2) This redundant with AI12-0427-1 (1). Tucker suggests dropping the one from 427 (the part about add after 4.10(4/5)) as this wording includes item (3) here. The other part of (1) remains. Randy notes that the rule was early so that it goes with the aspect; this is fairly late. That seems OK.

(3) This just added “nonabstract” to the previous rule.

(4) Tucker proposes replacing this with:

Modify 13.13.2(38/4):

The stream-oriented attributes may be specified for any type via an `attribute_definition_clause`. Redundant[Alternatively, each of the specific stream-oriented attributes may be specified using an `aspect_specification` on any `type_declaration`, with the aspect name being the corresponding attribute name.] Each of the class-wide stream-oriented attributes may be specified using an `aspect_specification` for a tagged type T using the name of the stream-oriented attribute followed by 'Class; such class-wide aspects do not apply to other descendants of T. {If not directly specified, a default implementation of a stream-oriented attribute is implicitly composed for a nonlimited type, and for certain limited types, as defined above.}

Approve AI with change: 12-0-2. Jeff, Gary abstain.

AI12-0436-1/01 Should a parallel block include exception handlers?

Richard agrees that we should allow the handlers here. He thinks that they are likely to be needed here, and it is great that Ada doesn't require an extra nesting level to include a handler.

Bob views this as a control flow construct, and we don't allow handlers in **if** or **case** statements.

Arnaud asks how Richard would indent this; he would use a blank line before **and**, with **exception** and **and** at the same level.

Straw poll: Should have handlers?

- Steve, Randy, Bob, Claire, Gary, J-P, Jeff: No handlers
- John, Justin, Richard, Tucker, Tuillio: Handlers.
- Ed: Abstain.
- Arnaud: opposes all versions of parallel blocks.

Tucker suggests removing them now and considering adding them later if it seems important to usage.

Approve AI: 8-1-5. Richard opposed. Tucker, Ed, Arnaud, John, Tullio abstain.

[Note: Jeff e-mailed a typo in this AI: “convinience” should be “convenience” in the discussion.]