Minutes of Electronic ARG Meeting 62F

9 September 2020

Attendees: Raphael Amiard , Steve Baird, Randy Brukardt, Arnaud Charlet, Jeff Cousins, Gary Dismukes, Claire Dross, Bob Duff, Brad Moore, Ed Schonberg, Justin Squirek, Tucker Taft, Tullio Vardanega (until 12:44), Richard Wai.

Observers: None.

Meeting Summary

The meeting convened on Wednesday, 9 September 2020 at 10:32 hours EDT and adjourned at 12:53 hours EDT. The meeting was held using Google Meet. The meeting covered the entire agenda and finished early.

AI Summary

The following AIs were approved:

AI12-0355-2/01 Aspect specifications for parallel constructs (9-0-5)
AI12-0392-1/01 Conditional expressions containing raise expressions (14-0-0)
AI12-0395-1/01 Allow aspect_specifications on formal parameters (12-0-2)

The following AIs were approved with editorial changes:

AI12-0378-1/06 View conversions and out parameters of access types revisited (14-0-0)
AI12-0388-1/02 Still more presentation issues (14-0-0)
AI12-0389-1/01 Ignoring unrecognized aspects (12-1-1)
AI12-0390-1/02 Conversions of anonymous access function results (14-0-0)
AI12-0391-1/01 List containers need Append_One (8-0-6)
AI12-0393-1/01 No invalid static constants (14-0-0)
AI12-0394-1/01 Named Numbers and User-Defined Numeric Literals (13-0-1)
AI12-0396-1/01 Fixups for various aspects of aspects (11-0-3)
AI12-0397-1/01 Default_Initial_Condition applied to derived type (13-0-1)

The following AI was discussed and assigned to an editor:

AI12-0384-1/02 Fixes for Put_Image and Text_Buffers

The following AIs were discussed and placed on hold:

AI12-0215-1/01 Implicit instantiations (13-0-0)
AI12-0215-2/02 Implicit instantiations (13-0-0)
AI12-0387-1/01 Private Global aspect (14-0-0)

The following AIs were discussed and voted No Action:

AI12-0355-1/01 Generalized aspect specifications (14-0-0)
AI12-0374-1/05 Fixes for nonblocking (12-0-2)

Detailed Minutes

Welcome

Steve welcomes everyone.

Apologies

Jean-Pierre Rosen has a conflict that we discussed last time. John Barnes says that he hasn’t figured out how to call in yet. Tucker sent him a phone number to use, but did not join the meeting.

Previous Meeting Minutes

There were no comments on the minutes: Approve minutes: 14-0-0.

Date and Venue of the Next Meeting

Following our approximately 6 week schedule, Randy had suggested Wednesday, October 21st for our next meeting. No one has any objections, so that date is chosen.

Reference Manual Review

Steve notes that Randy has sent out RM review assignments for everyone. He jokingly suggests that everyone other than Tucker must be disappointed to not get 3.10.2, aka The Heart of Darkness.

Unfinished Action Items

There are three unfinished action items (Steve Baird, AI12-0016-1; Ed Fish: AI12-0215-1; Tucker Taft: OpenMP Technical Report). We did not spend any time talking about these. [Note that we removed the Ed Fish action later in the meeting.]

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:

- Propose to extend aspect_specification syntax to most other declarations (see discussion of AI12-0395-1).

Editorial changes only:

- AI12-0378-1
- AI12-0388-1
- AI12-0389-1
- AI12-0390-1
- AI12-0391-1
- AI12-0393-1
- AI12-0394-1
- AI12-0396-1
- AI12-0397-1

Tucker Taft:

- AI12-0346-1 – also, construct the Technical Report suggested by this AI.
- AI12-0384-1
- AI12-0396-1 – propose additional wording to handle inheritance of sets of subprograms.
- AI12-0397-1 – propose additional wording to handle primitives of untagged types.
- New AI to handle storage pools for formal access types (from private conversation).
- New AI to define a Preelaborable_Initialization aspect (from ARG mailing list discussion).

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 26 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

AI12-0215-1/01 Implicit instantiations

No progress has been made on this and it is getting too late to do for Ada 202x

Hold: 13-0-0.

AI12-0215-2/02 Implicit instantiations

As with the previous, no progress has been made on this and it is getting too late to do for Ada 202x.

Hold: 13-0-0.

AI12-0355-1/01 Generalized aspect specifications

Having approved an alternative, this one should be dead.

No Action: 14-0-0.

AI12-0355-2/01 Aspect specifications for parallel constructs

This just adds aspect_specifications everywhere that **parallel** can be given.

Approve AI: 9-0-5. Arnaud, Claire, Justin, Ed, Raphael abstain.

AI12-0374-1/05 Fixes for nonblocking

We approved an alternative, so this needs to be killed.

No Action: 12-0-2. Arnaud, Raphael abstain.

AI12-0378-1/06 View conversions and out parameters of access types revisited

Steve notes that this permission never applies to derived subprograms (as those conversions are necessarily between related types).

Arnaud says this looks like a good compromise. He thanks everyone for continuing to look for a simplification of the earlier proposals.

Brad: !question. "...[we]{AdaCore} implemented this in GNAT ... .in [our]{their} customer..."

Approve AI with changes: 14-0-0.

AI12-0384-1/02 Fixups for Put_Image and Text_Buffers

Raphael and Tucker proposed a modification today: the root package would only contain routines that are needed to write a Put_Image routine. The unbounded package would be used to implement Image.

Tucker thinks that we need to develop a full alternative for next time. Randy notes that it is too soon to discuss an alternative that some of us may have only seen a few minutes before the meeting.

Keep alive: 12-0-1. Bob abstains.

AI12-0387-1/01 Private Global aspect

Tucker does not think there is enough support for this now. So he suggesting putting it on hold.

Brad notes a typo in the second sentence of the !proposal: "The syntax would be identical {to}[as] the Global aspect"

Hold AI: 14-0-0.

AI12-0388-1/02 Still more presentation issues

Brad notes a typo in wording for 6.1.1(29/5): "{is}[was] described".

Approve AI with changes: 14-0-0.

AI12-0389-1/01 Ignoring unrecognized aspects

As with pragmas, there is the possibility of implementation-defined syntax. Implementers don't have to deal with that; rejection is always allowed.

Arnaud notes that this helps for compatibility; ignoring new aspects is helpful with working with older implementations.

Richard says that from a user perspective it is strange that pragmas and aspect specs are different; they control similar kinds of things (and in some cases the same kind of things). He is in favor of making them the same.

The square brackets for 13.12.1(6.3/3) should have "Redundant" in them, as that's the purpose of the brackets.

Gary says !discussion is empty. "(See !problem.)".

Approve AI with changes: 12-1-1. Randy opposes, he notes that he explained why in e-mail, found in the !appendix. Bob abstains.

AI12-0390-1/02 Conversions of anonymous access function results

This is a simplification of the 3.10.2(10.3-4) rules.

Bob asks what GNAT implements; Justin has already implemented these rules. (He has been implementing this.)

Brad wonders if the last sentence of the !question could be simplified. Steve explains it; it doesn't have much to do with this question. Moreover, the question contained within it is not answered by this AI. Thus, the parenthesized part should be dropped, it just confuses.

Approve AI with changes: 14-0-0.

AI12-0391-1/01 List containers need Append_One

Gary can't undertstand the first sentence of the !problem.

> Ada.Containers.Doubly_Linked_Lists has [in] the Aggregate aspect[,] {with} Add_Unnamed using the existing Append operation.

Randy will reword this, a reordering would be better.

Randy notes that the name Append_One seems like a hack, it doesn't match the style of the other names in the package. He doesn't have a better idea, though. Brad had suggested some other names (in e-mail) that didn't get any traction.

Richard notes that it is a hack and it doesn't help to hide the equivalence to Append.

Ed suggests that we could instead change the Aggregate aspect to allow additional defaulted parameters. No one seems to respond to this.

It was suggested to use overloading for this instead (that is, remove the default from the existing routine and add a new overloading that has only two parameters). That is marginally incompatible, and it doesn't help with the aggregate overloading problem as Append_One does.

Approve AI with changes: 8-0-6. Brad, Bob, Randy, Justin, Tullio, Gary abstain.

AI12-0392-1/01 Conditional expressions containing raise expressions

Swaps the polarity of the list of object types, describing the list of things that are variables. That simplifies this list and reduces maintenance. No change is intended (although some things were still missing from the list of constants, such as parenthesized expressions).

The type conversion change is to prevent value conversions of tagged types in any case; we want all type conversions of tagged types to be view conversions.

Approve AI: 14-0-0.

AI12-0393-1/01 No invalid static constants

Tucker explains the change.

Gary: Last paragraph of discussion: "run{ }time" and "...so code using {it} would never be[ ] executed, and thus how it could be used is not relevant."

Arnaud notes that the test is probably B490001, rather than C490001. Randy will need to check this and change the number in the AI as needed.

Approve AI with changes: 14-0-0.

AI12-0394-1/01 Named Numbers and User-Defined Numeric Literals

A user noted that one cannot use named numbers with Big_Reals and Big_Integers. PI was noted in particular.

Tucker notes that named real numbers are really a rational value, so we need to support that.

Steve: 4.2.1(8/5): "Stirng" should be "String".

Steve wonders if we need to change the rule for static expression of a named number. 4.9(5) needs "of a numeric type" like 4.9(3/5). [After the meeting, it was determined that this was insufficient – Editor.]

Jeff typo: 2nd sentence of discussion: "situaiton"

Brad asks if "lowest terms" is defined. Tucker says that it is a proper mathematical term. The compiler has to use the lowest terms; there's no requirement on the function to do so.

4.2.1(8/5) ... The actual parameters of this notional call are each [being] a {String}[Stirng] with the textual representation of a decimal integer literal ...

Gary last sentence !proposal "... and {one}[ane] representing..."

Randy wonders of the importance of doing this given the complexity; the vast majority of constants should have types in Ada. Tucker says that it is weird that it would be different than numeric types.

Approve AI with changes: 13-0-1. Randy abstains.

AI12-0395-1/01 Allow aspect_specifications on formal parameters

Randy had suggested adding aspect_specifications to extended_return_statement, discriminant_specification, enumeration_literals. (Essentially all declarations other than those associated with larger constructs, such as iterators.)

Tucker worries that we could get into weeds discussing the details. So he would prefer making that a separate AI. Randy will take an action item to create such an AI.

Approve AI: 12-0-2. Jeff, Bob abstains.

AI12-0396-1/01 Fixups for various aspects of aspects

Tucker goes through all of the changes.

Remove the 7.3.4 wording change; Stable_Properties is not an assertion aspect.

Steve asks how 13.1(15.1/3) works if it denotes a set of subprograms (especially if some are primitives and some are not). Constant_Indexing works on a set of subprograms, for instance.

Tucker thinks we need to add an extra bullet for that. He will propose some wording.

Ed asks about the freezing. Tucker says that the equivalence will generally freeze everything. For "ignore" for assertion aspects, freezing still will happen.

Arnaud asks about implementation impact for these changes. Tucker doesn't think that there would be any. The intent is that these changes will make explicit the "obvious" interpretation that is already implemented.

Brad has editorials. Question (5): Period inside the quotes should be outside the quotes. (Two of these) Jeff thinks that this was changed in English English rather recently. [Editor's note: A recent discussion of this on the ARG mailing list seems to indicate that the "English" style of keeping punctuation out of quotes is preferred for our work. Modifying the contents of quoted strings this way can change their meaning and add confusion, particularly when Ada code is involved.]

Discussion, 1st para: "{a} true third kind".

Gary: 11.4.2(10.2/3): Remove the comma before the }

Approve AI with changes: 11-0-3 Gary, Jeff, Claire abstain. [Jeff and Claire say it is too much to take in.]

AI12-0397-1/01 Default_Initial_Condition applied to derived type

DIC is defined to apply to all descendants, but it was not explained how that works.

Tucker notes that DIC is similar to a postcondition that applies to all descendants. Essentially is similar to Post'Class. So he made the wording as close to that as he could.

Note that only the primitives of the type can be used directly (one can use an explicit conversion to call a nonprimitive subprogram).

Gary would like some discussion of this in the AI. Tucker will send Randy some additional discussion about that to add to the AI.

Steve worries about untagged types using a primitive operation that is defined later, such that it is not inherited. Tucker notes that the same issue applies to a generic unit; it uses the original operation (not overridden). Steve wonders if we need wording to cover that case. The generic text is "...even if is never declared for the actual type"; we need something similar here. Tucker will propose something.

Jeff: typo in problem. "all of these questions {should} be answered".

Approve AI with changes: 13-0-1. Jeff abstains.