

## Minutes of Electronic ARG Meeting 62A

15 January 2020

**Attendees:** Raphael Amiard (left at 12:30 EST), Steve Baird, Randy Brukardt, Jeff Cousins, Gary Dismukes, Claire Dross, Bob Duff, Brad Moore, Jean-Pierre Rosen (arrived 10:55 EST), Ed Schonberg, Justin Squirek, Tucker Taft, Tullio Vardanega, Richard Wai.

**Observers:** Arnaud Charlet, Yannick Moy, Boris Yakobowski.

### Meeting Summary

The meeting convened on Wednesday, 15 January 2020 at 10:38 hours EST and adjourned at 13:35 hours EST. The meeting was held using Zoom. The meeting covered many of the AIs on the agenda.

### AI Summary

The following AIs were approved with editorial changes:

- AI12-0342-1/03 Various issues with user-defined literals (part 2) (9-0-4)
- AI12-0343-1/02 Return Statement Checks (13-0-1)
- AI12-0345-1/02 Dynamic accessibility of explicitly aliased parameters (10-0-4)
- AI12-0348-1/01 Remove Combiners from Reduction Expressions (11-0-2)
- AI12-0351-1/01 Matching for actuals for formal derived types (12-0-1)
- AI12-0352-1/01 Early derivation and equality of untagged types (11-0-2)
- AI12-0356-1/01 Root\_Storage\_Type\_With\_Subpools should have Praelaborate\_Initialization (12-0-1)

The following AIs were discussed and assigned to an editor:

- AI12-0079-2/01 Default Global aspect for language-defined units
- AI12-0213-1/04 Unify record syntax [Note: this previously approved AI failed a No Action vote, so it remains Approved.]

The following AIs were discussed and voted No Action:

- AI12-0213-2/02 Unify record syntax (11-2-1)

The following AIs were discussed and placed on hold:

- AI12-0239-1/03 Ghost Code (11-0-2)

### Detailed Minutes

#### **Welcome**

Steve welcomes everyone.

#### **Apologies**

Jean-Pierre Rosen said he might be late (and he was). Bob Duff was on a train to New York and was in and out of contact during the meeting (he asked to abstain on all votes because he couldn't consistently hear the discussions).

#### **Previous Meeting Minutes**

No one has any changes to the minutes (all previously sent suggestions have already been applied). Approve minutes: 14-0-0.

#### **Date and Venue of the Next Meeting**

Our next face-to-face meeting will be June 12-14 in Santander, Spain immediately after the Ada-Europe conference.

Randy had proposed a date of March 11<sup>th</sup>, same time, for our next meeting. Steve asks if anyone has a problem with that date. No one does.

### **Presentation on the AdaCore Language Design Circle**

Yannick, Raphael, and Arnaud give a presentation on the LD Circle. The slides from their presentation can be found at [http://www.ada-auth.org/ai-files/grab\\_bag/Ada-and-SPARK-Evolution.pdf](http://www.ada-auth.org/ai-files/grab_bag/Ada-and-SPARK-Evolution.pdf). The presentation closely followed the slides.

Following are a few points of additional elaboration.

Yannick noted early on that the LD circle had asked for modifications to a number of AIs, including to remove AI12-0214-2, to simplify AI12-0208-1 and AI12-0079-1, and that they had no consensus on AI12-0189-1 and AI12-0262-1.

Arnaud notes that there is not a lot of desire from AdaCore customers for parallelism. The interest he has been able to find seems to be to use GPUs and to use APIs that use multicore system, not to write their own code that do those things.

After the presentation, Richard voices “discomfort” - he does not want an appearance that AdaCore is the “owner” of Ada. Justin notes that we are trying to have more coordination between Ada-Comment and LD Circle. Perhaps modernization of Ada-Comment is enough to accomplish this. Raphael notes that the LD Circle doesn't want to replace the role of the ARG. He thinks that it is more about implementation experimentation in a public forum.

### **Ada Design Principles**

Tucker goes over the document. Arnaud suggests that we would include some statement about performance. Tucker writes such a bullet which appears as we talk through the magic of the Internet. Arnaud also suggests splitting the document into two parts. Randy notes to that he was thinking we would “virtually” do that for this meeting, and that he would only post the Overall Principles as an official ARG document.

Approve Overall Ada Design Principles. 13-0-1. Bob abstains.

The approved document can be found here: <http://www.ada-auth.org/ai-files/minutes/ARG-design.html>

### **Unfinished Action Items**

There are three unfinished action items (Steve Baird, AI12-0016-1; Bob Duff, AI12-0243-1; Ed Fish: AI12-0215-1). We did not spend any time talking about 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
- AI12-0243-1 (help Bob Duff on request)

Randy Brukardt:

Editorial changes only:

- AI12-0342-1
- AI12-0343-1
- AI12-0345-1
- AI12-0348-1
- AI12-0351-1
- AI12-0352-1

- AI12-0356-1

Bob Duff:

- AI12-0243-1 (with help from Steve Baird)

Edward Fish:

- AI12-0215-1 (with help from Raphael Amiard)

Tucker Taft:

- AI12-0079-2 (create a new alternative with a split between access and non-access functionality)

## ***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 23 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***

### **AI12-0079-2/01 Global-in and global-out annotations**

Tucker took the text from 6.1.2 and tried to simplify the wording as needed, based on the discussions with Claire.

Tucker goes over the differences of this proposal from the previous one:

- We separated the notion of Global for library-level routines vs. nested routines.
- Not specifying Global gives the default, which is Global => null for Pure, Unspecified elsewhere. Probably won't specify it to be Unspecified very often.

Jean-Pierre says that you could have a coding standard that requires always giving a Global, which would require being able to say “Unspecified”.

Richard doesn't like the name “Unspecified”, but we decide to defer that level of question for the future.

Tucker goes on to the generic model, which defaults to assuming that every operation of the generic uses every formal.

“Appropriate and safe use” of access types are automatically allowed for ADTs.

Annex H defines dispatching info, and more details for generic formals.

Richard, wonders why any of this needs to be in the core. He thinks it is primarily for static analysis. Randy thinks that this is useful for code generation/optimization. Arnaud disagrees (and several others do as well). Someone does not think it is possible to verify access types. Claire Dross gives some details. She does not believe that access can be covered in Global. She worries that if we try (as in Tucker's draft), that the result would be incompatible with SPARK.

Randy notes that restricted uses of access definitely can be covered (probably with dynamic checks). Tucker suggests that Ada would have a separate aspect for access types. Raphael doesn't think that people would want to add such information to existing code. Arnaud claims that most people don't want to add Globals.

Tucker suggests splitting the AI to separate the access type issue. Raphael worries that if this feature is not ready, then we should forget about Global completely.

Claire suggests that we could at a later date add object ownership concepts on top of a basic Global aspect that only worries about direct reference to global variables.

Tucker suggests that we could change the goal to simply documenting the global objects used. And then create that AI. We could separately look at access type.

Straw poll: "Should we create a third version that is exclusively focused on global variables that are directly accessed?" 13-0-4 Abstain: Arnaud, Ed, Justin, Bob. (Remember that observers vote on straw polls.)

Tucker gets this back to create this new alternative.

### **AI12-0213-1/04 Unify record syntax** **AI12-0213-2/01 Unify record syntax**

We decide to take a straw poll to determine the preference for each of these options. The options are:

- 0 – Do nothing (use the Ada 2012 syntax)
- 1 - "end record id" (AI12-0213-1)
- 2 - "end id" (AI12-0213-2)

Straw poll:

- 0 – Claire, Richard, Arnaud, Randy, Ed, Jean-Pierre, Steve
  - 1 – Tucker, Brad, Tullio, Jeff, Gary
  - 2 – Justin, Yannick, Raphael (via message to Justin)
- Abstain: Bob, Boris.

No Action on AI12-0213-1: 5-2-6. Opposed: Tucker, Tullio. Abstain: Steve, Bob, Richard, Gary, Brad, Jeff. This vote fails. Since this AI was already Approved and the vote to change the status failed, it remains Approved.

No Action on AI12-0213-2: 11-2-1. Opposed: Justin, Raphael; Abstain: Bob. This vote passes (we can have 2 Opposers – 18% of 14 is 2.52).

[Editor's note: The subject of both of these AIs are misleading, as neither of them propose syntax/semantics that is exactly like some other syntax in the Standard. But I'll let this sleeping dog lie.]

### **AI12-0239-1/03 Ghost Code**

Justin had suggested to postpone further work on this AI, as the SPARK group has concerns about implementability, and believe that more prototyping and refinement is needed.

Tucker notes that he would like to see it prototyped soon. Randy notes that this is a useful step forward from traditional conditional compilation, but agrees that it needs to be prototyped.

Move AI to Hold: 11-0-2. Brad, Bob abstain.

### **AI12-0293-1/03 Add predefined FIFO\_Streams packages**

Bob asked to remove this from the Standard. Randy notes that we thought it was generally useful, and it is already defined.

Since Bob isn't able to discuss this due to his terrible connection, let's defer this.

## **AI12-0342-1/03 Various issues with user-defined literals (part 2)**

Steve explains the current wording. Randy notes that the model is now the same as used for stream attributes.

We talk a moment about the “statically denotes” wording. That prohibits specifying something like a dereference of an access-to-subprogram value as one of these attributes. Randy notes that we make a similar requirement for stream attributes, and as the function needs a result of the type and to be defined before the type is frozen, it is unlikely to provide much capability. There is also likely to be implementation work to support such a case. We agree that “statically denotes” is fine.

!proposal “for purposes {of} static semantics”...

Tucker objects to 4.2.1(6/5) he notes that a null type extension usually is allowed here. Steve thinks that doesn't work for this kind of inheritance. Randy concurs, as he thinks null extension semantics are mostly defined in 3.9.3, while 13.1 points at 3.4 for the definition of inheritance.

Steve then finds that 3.4 does include the null extension definition. So it would be OK to except nonnull extensions from this rule.

If a nonabstract {nonnull extension}[tagged] type inherits any of these three aspects,

Do we use “nonnull extension” anywhere in the Standard? We think that we don't. We think a bit about this, but an answer is not evident. Since this is the last AI of the meeting, everyone wants to move on, so we ask Steve or Tucker to suggest better wording to the editor.

Brad notes that the discussion says “...reemergence [for] with...”

Approve AI with changes: 9-0-4. Jeff, Richard, Bob, Jean-Pierre abstain.

## **AI12-0343-1/02 Return Statement Checks**

Jeff notes a typo in the third paragraph from end of discussion, “to we wouldn't” should be “so that we wouldn't”.

Steve notes that “11.4.1” should be “11.4.2” in the AARM Implementation Note.

Gary notes that the Implementation Note has “conversion [for] of”.

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

## **AI12-0345-1/02 Dynamic accessibility of explicitly aliased parameters**

Randy notes that the group as a whole should be glad that they didn't have to deal with the attempts to word this.

Tucker tries to explain the updated rules.

Gary: 3.10.2(19.2/5), delete “the body of F” (that's deleted wording).

Approve AI with changes: 10-0-4. Abstain: Jeff, Bob, Gary, Jean-Pierre.

## **AI12-0348-1/01 Remove Combiners from Reduction Expressions**

Richard notes that the `reduction_specification` syntax still has a combiner at the end.

Jeff: Second sentence of the summary: “{must} be”.

!problem: necessarily => necessary. “the [the]”.

Bounded Error: “, [then] it is a ...”

Gary: suggests “Sin” be “Sine” as he wonders what the “sin of X” is. Some laugh at the joke.

“using {a} Taylor expansion”.

Richard: “Fact” should be “Factorial” in this expansion.

Gary: “two{-}dimensional” (two occurrences).

Randy grumbles that these last few changes have nothing to do with Combiners.

Approve AI with changes: 11-0-2. Abstain: Gary, Bob.

### **AI12-0351-1/01 Matching for actuals for formal derived types**

Steve describes the problem.

The editor needs to add a !summary.

Approve AI with changes: 12-0-1. Bob abstains.

### **AI12-0352-1/01 Early derivation and equality of untagged types**

In the wording: “...before the {declaration of} primitive equality operator ...”

Gary: 2<sup>nd</sup> paragraph of the discussion “for records {for} which record equality composes”.

3<sup>rd</sup> paragraph of the discussion “...user{-} defined equality...”

Approve AI with changes: 11-0-2. Bob, Richard abstain.

### **AI12-0356-1/01 Root\_Storage\_Type\_With\_Subpools should have Preelaborate\_Initialization**

Randy notes that we always do this for preelaborable units unless there is a reason, and there does not seem to be any reason.

In !question, change “Preelaborate units” to “preelaborated units”.

Approve AI with changes: 12-0-1. Abstain: Bob.