Subject: Problems with SELECT TYPE and ASSOCIATE constructs
From: Van Snyder

1 Edits

Edits refer to 02-007r2. Page and line numbers are displayed in the margin. Absent other instructions, a page and line number or line number range implies all of the indicated text is to be replaced by associated text, while a page and line number followed by + (-) indicates that associated text is to be inserted after (before) the indicated line. Remarks are noted in the margin, or appear between [and] in the text.

[The non-constraint restriction at [163:24-25] should be part of C808. This is the only case where a prohibition against appearing in a variable definition context is not in a constraint. Editor: Insert ", is a *variable* that is not permitted to appear in a variable-definition context (16.5.7)," before "or".] 163:2

[Now covered by C808. Editor: Delete.] 163:24-25

C808$\frac{1}{2}$ An *associate-name* shall not be the same as another *associate-name* in the same *associate-stmt*. 163:3+

[Or should this be in 16.3?]

[The material of subclause 8.1.4.3 is about association, and properties of *associate-name* that arise as a result of its association with *selector*. As such, it belongs in 16.4.1.5. Editor: Delete [163:17] and move [163:18-23] to [401:8+], where the rest of the properties are discussed.] 163:17-23

[Editor: Insert "and 8.1.4.3" 398:21

2 Not sure what to do

I couldn't find anything, either in normative prose or syntax rules, that says an associate name is a variable. An *associate-name* isn't in the hierarchy of syntax rules starting from *variable* or *variable-name*. It seems necessary to put it under *part-name* (which is defined by the default rules), *parent-string* and *object-name*. I hope there's something simpler.

Hmmm, maybe it isn't always a variable – suppose the *selector*'s a procedure pointer? This is not directly addressed, but it appears to be indirectly prohibited by [401:11-12]. We need to say something concrete. If we conclude the *selector* can't be a procedure pointer, we could replace "an entity" by "a variable" at [163:11], but the syntax problem would remain.

Although the syntax allows nesting SELECT TYPE or ASSOCIATE constructs with each other or FORALL within them, I couldn't find anything about the semantics of doing this. [398:9-25] is the logical place for it.

(1) Do we want to allow them to be nested? Probably, since the syntax allows it.

(2) Do we want to allow the *associate-name* to be the same as an *index-name* of an enclosed FORALL? Probably, since nothing prohibits it now, and the *index-name* is in a new scope.

(3) Do we want to allow the *associate-name* to be the same as one for an enclosed SELECT TYPE or ASSOCIATE construct? Probably, since nothing prohibits it, but there's no discussion of which name means what. Host association doesn't do it.